Розроблено методологію онтологічної підтримки процесів конструктивно-продукційного моделювання (КПМ) структурно складних інформаційних технологій (ССІТ). Моделі онтологій предметних областей формуються і представляються на основі конструктивної структури, що містить первинні класи екземплярів онтології, активні сполучення операторів дій і виконавців, класи порівняльних властивостей, підпорядкованості та еволюційного розвитку. Універсальність моделі онтологічної конструктивної структури, можливості її розвитку та налаштування на предметні області дозволяє підвищити якість автоматизованих процесів створення ССІТ

Ключові слова: конструктивно-продукційне моделювання, онтологія, концептуалізація, уніфікація моделей, конструктивний об’єкт

Разработана методология онтологической поддержки процессов конструктивно-продукционного моделирования (КПМ) структурно сложных информационных технологий (ССИТ). Модели онтологий предметных областей формируются и представляются на основе конструктивной структуры, содержащей первичные классы экземпляров онтологии, активные связующие операторов действий и исполнителей, классы сравнительных свойств, подчиненности и эволюционного развития. Универсальность модели онтологической конструктивной структуры, возможности ее развития и настройки на предметные области позволяет повысить качество автоматизированных процессов создания ССИТ

Ключевые слова: конструктивно-продукционное моделирование, онтология, концептуализация, унификация моделей, конструктивный объект

UDC 004.82/.942
DOI: 10.15587/1729-4061.2017.119497

DEVELOPMENT OF ONTOLOGICAL SUPPORT OF CONSTRUCTIVE-SYNTHESIZING MODELING OF INFORMATION SYSTEMS

V. Skalozub
Doctor of Technical Sciences, Professor*
E-mail: skalozub.vl.v@gmail.com
V. Ilman
PhD, Associate Professor*
E-mail: valery_ilman@ukr.net
V. Shynkarenko
Doctor of Technical Sciences,
Professor, Head of Department*
E-mail: shinkarenko_vi@ua.fm
*Department of Computer Information Technologies
Dnipropetrovsk National University of Railway
Transport named after academician V. Lazaryan
Lazaryanastr., 2, Dnipro, Ukraine, 49010

1. Introduction

The development of diverse applications of information technology (IT) is characterized by the increasing use of ontologies and ontological systems (OnSs) [1, 2]. As a rule, ontological technologies are used in complex for formalization, weakly structured fields of activity or business for processing, classifying and interpreting growing volumes of data [3, 4]. For example, in the field of the World Wide Web, the use of OnSs has become a generally accepted and effective tool for solving numerous problems [5, 6]. System-wide standards have been developed for the ontologies and many applied areas of IT [6]. Languages to describe ontologies and software for creating, integrating, and automatic administration are being developed [4, 6–8]. Ontologies are explicitly used as data sources for many computer applications such as text analysis, knowledge extraction, information retrieval, system design, etc. The advantage of ontology is to provide efficient processing of complex and diverse information, as well as the ability to reuse it in a new environment [9, 10].

Ontology means a system of concepts that is represented by a set of entities connected by different relations that characterize a certain area of knowledge and are used for a formal specification. An important component of ontology is a set of axioms that provide a representation of additional knowledge that does not cover the hierarchy of concepts.

At present, the application of subject-matter ontologies of specific areas of knowledge to support complex and knowledge-intensive IT is important for many purposes, including the evaluation of conceptual data models [11], the use of inaccurate knowledge represented by examples, with the help of fuzzy algebra [12], and the construction of domain ontologies based on the model of fuzzy output [13].

In the course of research on the development of applied ontological systems, there are also problems of developing ontological support for the processes of constructive-synthesizing modeling (CSM) for structurally complex information technologies [15–19].

The application of CSM allows solving a number of tasks of information technologies in the field of software development and power supply, for example:

– automation of rational power distribution processes for train traction recovery in DC systems [15];

– procedures for adapting data structures in RAM [16];

– improving data storage structures in plagiarism detection problems [17];

– adaptation of compression algorithms to archived data [18];

– improving the decision-making processes by ranking alternatives by the hierarchy analysis method [19].

The wide range of these tasks demonstrates the universality and perspectives of applying CSM to the solution of several classes of IT tasks of different subject areas.

The above mentioned articles reveal the universality, high generality, and typical procedures of this method of modeling information systems.

It should be noted that the solution of problems in [15–19] is based on the same type of constructive procedures, the use of which has helped obtain the structure of models and standardly implement diverse objects of informatization. In general, the design processes are widespread; they arise at different stages of IT development and have a fairly high generality. For example, the formation of software system architectures, the conceptual modeling of the OnS domain, the development of database structures [20] and others are essential design tasks. To a large extent, the processes of this kind of construction contain intuitive, unstructured elements. The use of automation systems to implement some common models of design processes produce "an imprint" on the capabilities and quality of playback of various IT areas.

Multiplicity of forms and prevalence, target similarity, complexity and uncertainties of design tasks in the IT field make research for intellectual and technological support of design processes relevant and important. At the same time, at this stage of research on constructive modeling, it should be pointed out that there is insufficient methodological basis and lack of information and technological support for the specialized processes of CSM.

Therefore, it is essential to carry out research and development aimed at improving the efficiency and reliability of information modeling, as well as reducing the time of creating IT based on methods of ontological support of CSM.

2. Literature review and problem statement

In practice, any technological processes, including information, are provided with methodological, organizational, information and technical means. Information support is unthinkable without the use of ontologies, formal or informal. Ontological support in the context considered here is the basis and a connecting link of the means listed above. In scientific sources on ontological systems in most works, the predominant directions of research are the creation, improvement and use of applied ontologies in various IT areas. Thus, in [21], a heuristic clusterization algorithm is proposed for coordinated solutions based on the dominance of simple decision systems on ontological graphs. The development of the analysis of ontological graphs is supplemented in [22] using paradigmatic and syntagmatic relations. In [11], the CASE toolkit is used to create a conceptual data model based on ontologies. An ontology editor for modeling domain ontologies with a set of rules for the semantic estimation of the model has been developed. It is noteworthy that all ontologies in these works have a simple hierarchical structure.

In [23], a methodology is developed to create classification systems based on ontology. The use of ontology for working with complex objects and properties is solved by creating an intrusion detection system [2] that helps identify complex attacks occurring in the network. Modeling conceptual data is one of the most important stages in the development of information systems and technologies. Study [11] suggests a method and an ontological system for evaluating conceptual data models integrating the rules of the PROLOG system.

Important issues concern the application of ontologies of specific areas of knowledge to support complex and knowledge-intensive IT. Work [12] solves the problem of using inaccurate knowledge obtained from past experiences (cases). Based on the Case Based Reasoning (CBR) ontology, such knowledge is used to interpret or solve new problems with fuzzy algebra. Study [13] proposes a method of automated construction of domain ontology based on the Stanford fuzzy inference model in information learning Web systems. A specialized hierarchical-network object-oriented model of data and knowledge has been developed. These properties are due to the specificity of the domain and the heterogeneity of content that significantly distinguish the system from many OnSs.

Formal means for describing ontologies are an important advantage of the computer approach. Also, there is a known mathematical approach in which definitions of the concept of ontology in mathematical terms are given [1, 9, 10]. In [1], it is noted that "the ontological approach to the representation and integration of scientific knowledge allows creating effective tools for constructing systems ... of transdisciplinary interaction and ontological engineering." This provision extends to structurally complex interacting information technologies.

In [9, 10], ontology is defined as "a set of logical axioms designed to take into account the supposed meaning of the dictionary. Given the language L with ontological commitment K, the ontology for L is a set of axioms designed in such a way that the set of its models maximally approximates the set of assumed models L by K" [9]. It is emphasized that "ontology depends on language while conceptualization does not depend on language" [9].

When forming the ontology of CSM for the concept of "subject domain ontology" (SDO), we adhere to a number of fundamental methodological principles [6]. At the content level, the set of agreements of the SDO cannot be refuted by empirical observations. The properties of the subject domains (SDs) (ontology, conceptualization, knowledge, and reality) should be modeled by a single mathematical construction. The model of the SDO should contain formal elements and their meaningful interpretation (in special terms of this subject area). Ontology and its model should be observable, even for complex subject areas with a large number of concepts. According to [8], there is no single right way to model the domain, which depends on the intended application and the expected extensions.

In many cases, applied OnSs play several roles in knowledge representation tasks. They express taxonomic and other properties in conceptualization, and they also can implement the functions of specialized knowledge bases. In the latter case, ontological support for IT extends to the formation of models of concepts (individual concepts, links, and means of creating instances) and the use of specialized inference procedures. Many application ontologies for mathematically and logically complex IT applications have such characteristics (fuzzy control, multicriteria analysis, etc.). The noted peculiarities of applied IT ontologies give grounds for the further development of a system of ontology models of CSM.

The analysis of research materials of applied ontologies has shown the priority of developing means of supporting conceptualization processes in OnSs. For example, UML [24] is used as a tool. It is also important to pursue “the development of methodology and tools for the automated design of a formal (computer) ontology” [25].

As Academician A. V. Palagin notes, “the ontological approach presents the user with a holistic system view of the subject area or cluster of subject areas” [1]. Ontological support is designed to maintain specific technologies and technological processes in applied systems.

The line between ontology and ontological provision is rather “subtle”, but at the same time, the object of many studies is provision as a means of improving technology. Let us indicate a number of the revealed advantages of ontological support of processes:

– planning, which provides a “richer presentation of plan-related data and semantics” and allows the construction of “flexible, non-destructive, scalable and coordinated changes in plans” [26];

– supporting the evolution of service-oriented architectures, which have become a useful basis for the development of compatible large-scale systems [27];

– developing a template in equipping the aviation industry “helps reduce the complexity of design, improve the quality of the original design, and reduce pre-training” [28];

– working design, predetermining the possibility of automatic detection of design errors in the final CADD model [29].

In all the studies [2–13, 21–29] explicitly [12, 24] or implicitly, ontology models are based on the well-known paradigm of object-oriented programming – the whole world is represented by a set of objects with their properties and connections.

The developed methodology of CSM is built on the paradigm of structures – the world is seen as a set of designs and constructive processes. At the same time, ontological constructive models and procedures are put in correspondence with the selected fragments of reality the elements of which can be objects and the like. It seems promising to harmonize approaches and modeling methods with their ontological support.

In [25], it is stated that “there is still no generally accepted methodology, and the question of the appearance of a constructive theory for the development of formal ontologies remains open”.

The tasks of conceptualizing the properties of constructive structures, as well as the procedures for forming IT elements when modeling, arise when creating the ontological support of CSM.

The problem of creating intelligent means of automated support of CSM methodology is due to the potential complexity and uncertainty of the application areas of CSM methodology, which put forward high requirements for ontological CSM (OCSM). In ontology procedures, the main attention should be paid to the universality of the concept model, both in terms of structural properties and attributive content. A common goal in this case is to provide the formation of concepts for structurally complex modeling areas. Therefore, in OCSM, structures other than taxonomic of the concept system may be formed, given the uncertainties of the hierarchy of concepts and distinctive features.

The questions of ontological support of CSM with the indicated properties do not yet have a general concept of implementation and have not been formalized. There is a need to develop the foundations of ontological support. Due to the specific nature of the constructive approach to modeling, namely, as the representation of all components of IT in the form of structures and constructive processes, existing approaches to the formation of ontological support for CSM should be substantially revised.

3. The aim and objectives of the study

The aim of the work is to develop a methodology and tools for ontological support of the processes of constructive-synthesizing modeling (CSM) for structurally complex information technologies.

To achieve this aim, the following new tasks in the field of creating an applied ontology of CSM were formulated and solved:

– to identify and study the properties of the structures of notions, concepts and basic system-forming relations under CSM, to formulate requirements for the properties of the methodology, as well as to specialize models and methods of applied OCSM;

– to develop a unified model of an ontological constructive structure, which should be universal, developed and customizable for modeling subject areas;

– to devise a model and procedures that implement within CSM the possibility of introducing constructed relations the structure of which is not known in advance, including recursive relations;

– to develop means of OCSM that provide opportunities to formulate and maintain models of conceptual systems of subject areas other than taxonomic. At the same time, uncertainty factors should be taken into account when choosing the structure of conceptual models, both for systems of concepts of subject areas and their distinctive features.

4. Materials and methods of studying ontological support of constructive-synthesizing modeling

We will proceed from the fact that the ontological system can be specified by primary sets of abstract conceptual categories that have the property of universality and are systematically determined by the toolkit for constructing objects. Let us consider the basics of possible constructive modeling of ontology elements on sets of instances of an arbitrary subject domain.

We assume that a variety of potentially existing sets of objects of the SD system form a conceptual class of objects $Ob\overline{\mathfrak{K}}$, and let $Rf\overline{\mathfrak{K}}$ be the class of relations and mappings between objects and components of this class. The system taxonomy $\overline{S}=\langle Ob\overline{\mathfrak{K}}, Rf\overline{\mathfrak{K}}\rangle$ of an ontological universe $\overline{\mathfrak{K}}$ defines an ontology subsystem $S=\langle Ob\mathfrak{K}\ Rf\mathfrak{K}\rangle$ $(S\subset\overline{S})$ of the $\mathfrak{K}$ ontology if

– the class $Ob\mathfrak{K}$ is part of the class $Ob\overline{\mathfrak{K}}$, $Ob\mathfrak{K}\subset Ob\overline{\mathfrak{K}}$;

– $Rf\mathfrak{K}\subset Rf\overline{\mathfrak{K}}$;

– the characteristics of the objects of the class $Ob\overline{\mathfrak{K}}$ are also preserved in the objects of the class $Ob\mathfrak{K}$;

– the properties of relations and mappings of the class $Rf\overline{\mathfrak{K}}$ are transferred to the relations and mappings of the class $Rf\mathfrak{K}$ in whole or in part.

Ontological samples have generic and specific definitions [1], which form the properties, indicators, attributes and other components of concepts. Thus, if an object a of a set of specific definitions is determined by the generic definition x, then the concept of the object is represented by the dependence $a(x)$.

Conceptualization of $\mathfrak{K}$ ontology as a process of forming certain concepts is constructed by its classes, $Ob\mathfrak{K}$ and $Rf\mathfrak{K}$. Mapping is constructed using the class of operators $\mathfrak{D}$ and mappings of the generating subclasses of the class $Rf\mathfrak{K}$, and the design of objects is performed on the generating objects of the class $Ob\mathfrak{K}$ by means of relations and mappings of subclasses of the class $Rf\mathfrak{K}$ and their hybrids.

Definition 1. The operator $\phi \in \mathfrak{D}$ constructs the mapping ψ on the class $Rf\mathfrak{K}$ if:

– for components $\mu_i, \mu_j \in Rf\mathfrak{K}$, the mapping $\psi = (\mu_i, \mu_j)\phi$ is admissible (the operator ϕ is admissible in the mappings μ_{ji}, μ_j);

– the properties of the components μ_i, μ_j are completely or partially carried over to the mapping ψ.

Thus, if we assume that the composition operator is $\phi = \bullet$, $\phi \in \mathfrak{D}$, then it constructs the n-place list mapping $\psi = (\mu_1, \mu_2, \cdots, \mu_n)\phi = \mu_1 \bullet \mu_2 \bullet \cdots \bullet \mu_n$, $\mu_i \in Rf\mathfrak{K}$, $1 \le i \le n$. The maps ψ can also be formed on other admissible operators of the class $\mathfrak{D}$.

Definition 2. Let us call an ontological object $B \in Ob\mathfrak{K}$ recursively constructed (generated) from an object $A \in Ob\mathfrak{K}$ by the mapping $\psi = (\mu_1, \mu_2, \cdots, \mu_n)\phi$, $\mu_i \in Rf\mathfrak{K}$ if the components form an admissible constructible sequence $((A_i)((\mu_i)\phi) = B_i;\ A_{i+1} = B_i, A_i \subset A, B_i \subset B)$, and this generation is represented as ${}^{\psi}_{A}B$ or if it is necessary to point to the generation, as $A \prec B$.

The process of constructing ontology of objects depends on the evolution of the mappings and the level of iteration of their derivations. In particular, for a singular (identity) mapping, we have $A_i \prec A_i$; if the process of construction "cycles", then the generation is empty (the object B_i is not generated); generation can be finite or infinite.

Statement 1. Infinite generation (without cycling) of the object ${}^{\psi}_{A}B$ is countable.

For successive immediate generations ${}^{\psi}_{A_i}B_i$ of the constructive process can be numbered or counted using the length of the generations (the number of direct generators) of the constructions.

It is possible to assign a certain component $\mu_i \in Rf\mathfrak{K}$ to an inductive mapping $\varphi_i = (\varphi_{i1}, \varphi_{i2}, \cdots, \varphi_{im})$ so that

$$(A_i)\mu_i = (A_{i1}, A_{i2}, \cdots, A_{in})\mu_i = \{(A_{j_1}, A_{j_2}, \cdots, A_{j_k})\varphi_{ij},$$

$$A_{ik} \subset A_i \,\&\, (A_{iq}, A_{ir})\varphi_j = (A_{iq})\varphi_j (A_{ir})\varphi_j\}, \quad A_{ik} \in Ob\mathfrak{K}.$$

The generated object B of the ontology is determined by the form $Str(B)$ and the content $Sod(B)$. The form is structured by the formal structure of the object construction. The substantive side of the constructed object is related to the base and its properties, which are manifested through comparison with other objects. When constructing, generic indicators of semantic concepts and the values of objects on which signs of ontological criteria are formed can be taken into account. The conceptual properties of the components of objects are generated (formed) when they are constructed.

Thus, for an admissible relation $\nu \in Rf\mathfrak{K}$, $(A,B)\nu = Sv$ $A, B, Sv \in Ob\mathfrak{K}$ $\nu \triangleq$ "as an object property with respect to the object," the sequence of relations ν and $\multimap$ $(A,B)\nu \multimap gs$ determines the properties Sv of the object B with respect to the object A with the exponent $gs \in Ob\mathfrak{K}$, where the symbol $\multimap$ is the derivative relation of the property $S=(A,B)v$ to its exponent gs. In turn, gs can be assigned by the operator (:=) some value u and $(A, Sv, gs, u) \subset Sod(B)$.

If the relation $\rho \in Rf\mathfrak{K}$ is a relation "to be subordinate," the property Sv becomes a subordinate concept of the object B under the influence of the relation $(Sv, B)\rho = {}^{\rho}_{Sv}B$ [14], i. e. the constructive object ${}^{\rho}_{Sv}B \in Ob\mathfrak{K}$ is specified by the composition of relations $\nu \bullet \rho$, and the form of the object ${}^{\rho}_{Sv}B$ is determined by the sequence of relations and the operator $(\nu, \multimap, :=, \rho)$.

Let us note that the construction of an object ${}^{\rho}_{Sv,gs,u}B$ is the basis for the formation of a knowledge base (KB) network of the ontology $\mathfrak{K}$.

Let the $\mathfrak{O} \subset Ob\mathfrak{K}$ class of constructed conceptual objects be such that $(\forall B \in \mathfrak{O},\ \exists A \in \mathfrak{O},\ A \prec B)$.

Definition 3. A set of generated objects $\{B_j;\ {}^{\psi}_{A}B_j,\ A \in \mathfrak{O}\}$, together with A, is defined by a class of directed (non-cycled) structures generation ${}^{\psi}_{A}\mathfrak{O}$.

We have the obvious lemma:

Lemma 1. ${}^{\psi}_{A}\mathfrak{O} \subset \mathfrak{O}$.

Definition 4. The class $\mathfrak{C}_{\psi}$ is a simple (elementary) mapping of $\psi \in Rf\mathfrak{K}$ if $\forall B \in \mathfrak{C}_{\psi}, (\nexists A \in \mathfrak{C}_{\psi}, {}^{\psi}_{A}B)$.

Statement 2. $\mathfrak{C}_{\psi} \subset \mathfrak{O}$.

Definition 5. A related generation with the initial simple object A is directed to form a fully connected class of objects ${}^{\psi}_{A}\tilde{\mathfrak{O}}$.

Theorem 1. In the class of objects $\mathfrak{O}$ there is at least one fully connected subclass.

From the fact that the class $\mathfrak{O}$ is constructed, it is always possible to specify various objects of $A_i \prec B_j$ related to generation and to construct the directed class ${}^{\psi}_{A_i}\mathfrak{O}$ that is countable by statement 1 and is a subclass of the class $\mathfrak{O}$ by lemma 1. In this case, since by statement 2 the class O contains also simple objects by the generating mapping ψ, then:

1) at $A_i \in \mathfrak{C}_{\psi}$, $B_j \in {}^{\psi}_{A_i}\tilde{\mathfrak{O}}$, which proves the theorem;

2) if the generation B_j is incompletely connected, then $(\exists A_m \in \mathfrak{C}_{\psi}), (A_i \in {}^{\psi}_{A_m}\tilde{\mathfrak{O}})$ and the generation of the object B_j becomes fully connected in the class $\mathfrak{O}$.

Consequence 1. The structure of the fully bound generation of the ontological object B defines the path of generating this object ${}^{\psi}_{B}P$, where ${}^{\psi}_{B}P \subset \mathfrak{O}$. The generation paths in the class O define the topology of this class.

Theorem 1 proves the local existence of a set of simple generators, which are objects of the ontology class $\mathfrak{O}$ for a certain mapping. For different mappings constructed on the components of the class $Rf\mathfrak{K}$, other simple sets of objects are distinguished in the class $\mathfrak{O}$. The whole set of simple sets of objects distinguished by different mappings $\psi \in Rf\mathfrak{K}$ forms the family $\mathfrak{C} \subset \mathfrak{O}$.

The proposed conceptualization of the ontology of the SD system is incomplete; replenishment and development will be carried out in the process of constructive-synthesizing modeling. An exogenous constructive structure is considered as an ontological model [14]. Let us note that the model of a constructive structure is systemic, admitting a structural decomposition and a substantive specification. The constructive structure has a defined or set input and a constructed output. These system properties will be further taken into account in the level and order constructive decompositions of the abstract ontology.

The ontology development process proceeds from being simple to complex, so let us start constructing the components of the ontology in the ontological constructive structure (OCS) into simple subclasses of the conceptual classes of objects $\mathfrak{C} \in Ob\mathfrak{K}$ with relations and mappings of the class $\mathfrak{Y} \subset Rf\mathfrak{K}$.

5. Development of an ontological constructive structure and results of its research

5. 1. Ontological constructive structure

A convenient tool for modeling the ontology of an IO system is the constructive structure of an OCS set by the ordered tetrad

$$\hat{C} = \langle \hat{\mathrm{M}}, \hat{\Sigma}, \hat{\Lambda}, \hat{\mathrm{Z}} \rangle, \quad (1)$$

containing carrier classes (Carrier construction) $\hat{\mathrm{M}}$, signatures $\hat{\Sigma}$, calculations $\hat{\Lambda}$ and performers $\hat{\mathrm{Z}}$.

The classes of the carrier and the performers allow us to consider structure (1) in two aspects: *determining* and *generating*. The determining aspect is the point of view on the *Carr* structure as the main defining part, which forms with the help of external performers $\hat{Z}_V \subset \hat{Z}$ components of the carrier $\hat{\mathrm{M}}$, signature Σ, calculations Λ and internal performers, i. e.

$$Carr \xrightarrow{\hat{Z}_V} C = \langle \mathrm{M}, \Sigma, \Lambda, \mathrm{Z} \rangle, \quad (2)$$

where $\mathrm{Z} = \hat{Z}_V \bigcup Z_v$.

The generative aspect of constructive structure (1) manifests itself in the structure *C*, an arbitrary subject domain, through the construction of conceptual objects of the carrier, signatures, calculations, and internal performers. From this point of view, structure (1) in a definite image of *C* is a generating structure.

Suppose that the defining aspect of constructive structure (1) of the IO in the image is related to the ontology system $\mathfrak{K}$ and defines the constituent parts of the ontological constructive structure as a set of components: the carrier, signatures, calculations, and performers.

Carrier classes. The carrier *M* of the structure *C* includes the input base of simple objects (instances) of M_0, which is intended to form conceptual objects on it. The carrier *M* also includes abstract notions of additional symbols *D*, elements of classes, such as operators, relations, mappings, and constructive forms; characteristic notions of their indices and values, as well as classes of free structures of the arbitrary level. The base carrier M_0 is fundamental to the evolutionary formation of generic and specific constructive classes: objects $\mathfrak{O}^k$, attributes $\mathfrak{L}^k$ and derived ontological objects $\mathfrak{O}^k_*$ of the *k*-th levels of development.

Signature classes. The signatory class Σ of the ontological structure defines actions for the formation of objects and new mappings and relations. The class is composed of the system-generating basic Σ^0, the generating Σ^+ and the constructed Σ^* signatures. The generating signature consists of a class of simple operators $\mathfrak{D}^0 \subset \mathfrak{D}$, a class of simple relations and mappings $\mathfrak{Y}^0 \subset \mathfrak{Y}$, including binding subclasses $\mathfrak{A}^0$ and $\mathfrak{R}^0$, and a mapping class of generating selection $\mathfrak{V}^0$ of ontological objects. The generating signature includes classes of the mappings $\mathfrak{Q}$ and $\mathfrak{M}$, which are intended for the formation of constructs – actions of the Σ^* class.

Classes of performers. Classes of performers *Z* of the constructive structure include subclasses of external Z_V and internal Z_v performers and their hybridization Z_g. As a rule, experts play the role of external performers. Internal performers are designed to implement class $\mathfrak{D}$ action operators, interpretations of signature Σ components, and action algorithms constructed in structure (2).

A class of calculations. The considered components of the OCS are postulated, determined and converted by calculations from the class Λ. The calculation class defines a permissible "operator" and consists of postulates (axioms), definitions, instructions, rules, properties of structure components and other tools necessary for organizing the construction of objects and mappings. It reflects the properties of the systemic structuring of the OCS. The calculations are the core of the OCS because the rules of calculating Λ construct classes of the structure, i. e. $\Lambda: \mathrm{M} \times \Sigma \to \mathfrak{O}^k$, $\Lambda: \Sigma \times \Sigma \to \Sigma^*$, $\Lambda: Z_V \times \mathrm{Z}_v \to Z_g$ and $\Lambda: \Lambda_i \times \Lambda_j \to \Lambda_g$; moreover, the permissive characteristic of the calculation specifies different IOs.

Since Λ calculations are the most important component of the constructive model of the OCS, let us consider its components in the individual specified classes of structure (2).

5. 2. A basic carrier of ontology instances

The construction of objects in structure (2) can be performed on any instances, defining and defining concepts. However, the property of the system and the taxonomy of the ontological structure can be violated. In order to avoid this, the basic carrier M_0 is introduced into the structure, giving the initial positions of the generation of objects.

Basic prerequisites. Let us consider the system of requirements that determine the basic carrier:

– the basic carrier M_0 is determined by external performers Z_V on copies of simple specific and generic classes $\mathfrak{C}_0, \mathfrak{X}_0 \subset \mathfrak{O}$ so that $M_0 = \mathfrak{C}_0 \cup \mathfrak{X}_0 \cup \mathfrak{J}_0 \cup \mathfrak{U}$, $\mathfrak{C}_0 \subset \mathfrak{C}$, $\mathfrak{X}_0 \subset \mathfrak{X}$, where $\mathfrak{C}_0$ and $\mathfrak{X}_0$ are, respectively, the specific and generic base classes of concepts: $\mathfrak{J}_0 \subset \mathfrak{J}$ is the class of indicators, $\mathfrak{U}$ is the class of values of the indicators, characteristics, criteria, etc.; the $\mathfrak{U}$ class is heterogeneous, including clear and fuzzy numerical, symbolic, graphical and other data collections;

– the capacities of the classes $\#\mathfrak{C}_0$ and $\#\mathfrak{X}_0$ are finite;

– c_i and x_i are instances of the classes $\mathfrak{C}_0$ and $\mathfrak{X}_0$, i. e. $c_i \in \mathfrak{C}_0$, $x_i \in \mathfrak{X}_0$; the instance $c_i \in \mathfrak{C}_0$ may have internal generic definitions $x_i \in \mathfrak{X}_0$, i. e. $c_i(x_i)$;

– the record $c \equiv g$ indicates that the instance $c \in \mathfrak{C}_0$ has an exponent $g \in \mathfrak{J}_0$, and $g := u_i$ is the assignment of the value $u_i \in \mathfrak{U}$, and $\forall x_i \in \mathfrak{X}_0$; $x_i := u_i$ is permissible, where the operators are $\equiv, := \in \mathfrak{D}^0$;

– the constructions $(c_k(t), c_i)_j$, $(c_k, c_i(t))_j$, $(c_k(t), c_i(t))_j$, $t \in T$ are dynamic;

– the instances $x_0 \in \mathfrak{X}_0$ and $c_0 \in \mathfrak{C}_0$, for which, respectively, the generic and specific concepts and elements $g_0 \in \mathfrak{J}_0$, $u_0 \in \mathfrak{U}$ without a meaningful value ($c_0 := u_0$, $x_0 := u_0$ and $g_0 := u_0$) are called empty; in the classes $\mathfrak{C}_0$, $\mathfrak{X}_0$, $\mathfrak{J}_0$ and $\mathfrak{U}$ there exist the corresponding unique empty elements c_0, x_0, g_0, and $u_0 \in D$.

Subclasses of basic classes. Let us introduce preliminary requirements for subclasses of the base class:

– the class $\mathfrak{C}_1$ is a subclass of the class $\mathfrak{C}_0$, i. e. $\mathfrak{C}_1 \subset \mathfrak{C}_0$ if $\forall c_i \in \mathfrak{C}_1$, $c_i \in \mathfrak{C}_0$;

– the subclass $\mathfrak{C}^0 \subset \mathfrak{C}_0$ is empty if $c_0 \in \mathfrak{C}^0$ and $\#\mathfrak{C}^0 = 1$;

– the classes $\mathfrak{C}_0$ and $\mathfrak{X}_0$ and their subclasses are free if the location of elements in the classes is arbitrary (character indices are used to distinguish between elements or classes); if the elements in the classes $\mathfrak{C}_0$ or $\mathfrak{X}_0$ are repeated, they denote multiclasses;

– if $\mathfrak{C}_1, \mathfrak{C}_2 \subset \mathfrak{C}_0$, then $\mathfrak{C}_1 \cup \mathfrak{C}_2 \subset \mathfrak{C}_0$, and if $(c_0 \in \mathfrak{C}_1) \& (c_0 \in \mathfrak{C}_2)$, then $\mathfrak{C}_1 \cap \mathfrak{C}_2 \subset \mathfrak{C}_0$.

It is noteworthy that the results of the preliminary information section substantiate the postulation of the basic carriers $\mathfrak{C}_0$ and $\mathfrak{X}_0$ of the OCS. The calculation defines the simple carrier class $\mathfrak{C}_0 \cup \mathfrak{X}_0 \subset M_0$ as elementary and free, which are necessary conditions for performing the properties of the systematicity and structuredness of the generated classes of objects in the constructive structure. The results of this paragraph supplement the basic statements of the conceptualization of the SD system ontology.

5. 3. Simple relations of comparison

Generation of ontological objects of IO is performed by the signature component of the OCS containing the basic component Σ^0.

Definition 6. The class of relations and mappings $\mathfrak{Y}^0 \subset \Sigma^0$, $\mathfrak{Y}^0 \subset \mathfrak{Y} \subset Rf\mathfrak{K}$ on the base carrier $\mathfrak{C}_0 \cup \mathfrak{X}_0 \subset M_0$ sets the ontological taxonomy $\langle \mathfrak{C}_0 \cup \mathfrak{X}_0, \mathfrak{Y}^0 \rangle$ of the objects of the zero level $\mathfrak{K}^0$.

In turn, the class $\mathfrak{Y}^0 \subset \Sigma^0$ contains a relationship class $\mathfrak{R}^0$ intended for the formation of properties, indicators and binding of ontology concepts. The construction of individual properties of instances and indicators in OCS structure (2) is performed by the relations of the subclass $\mathfrak{R}^0_\nu \subset \mathfrak{R}^0$ comparison, as well as by the derivational relations of class correspondence $\mathfrak{R}^0_c \subset \mathfrak{R}^0$ according to the rules of calculating Λ.

Generic and specific comparisons. The classes of specific $\mathfrak{C}_0$ and generic $\mathfrak{X}_0$ instances set the initial state of constructing with the relation $\nu \in \mathfrak{R}^0_\nu$. The calculations with the relation ν are the following:

– the generic and specific elements $c_k, c_i \in \mathfrak{C}_0$ are common as to v when the relation is comprehensively applicable to them and their internal generic definitions;

– the simple comparative specific and generic relations $\nu_j, \nu_q \in \mathfrak{R}^0_\nu$ are performed on the directionally compared elements of the pairs (c_k, c_j) and (x_k, x_j), and the relation $\nu \in \mathfrak{R}^0_\nu$ affects the carrier $\mathfrak{C}_0 \cup \mathfrak{X}_0 \subset M_0$ so that $\forall c_k, c_i \in \mathfrak{C}_0$, $\exists v_j \in \mathfrak{R}^0_\nu$, $(c_k, c_i)\nu_j$ and $\forall x_k, x_i \in \mathfrak{X}_0$, $\exists v_q \in \mathfrak{R}^0_\nu$, $(x_k, x_i)\nu_q$;

– the relation $\nu_q \in \mathfrak{R}^0_\nu$ on the definitions $x_k, x_i \in \mathfrak{X}_0$ sets the generic construction $(x_k, x_j)\nu_q$, which at $x_i \neq x_k$ or $x_i \neq x_0$ determines the generic comparison $(x_k, x_i)_q$ (the property of the generic element x_i in terms of ν_q to the element x_k);

– if $c_i \neq c_k$ or $c_i \neq c_0$, then the comparative relation $(c_k, c_j)\nu_j$ determines the construction-property $s_j = ((c_k, c_i)^1_j, (x_k, x_i)^2_j)$ (s_j is the property of the element c_i in terms of ν_j to the element c_k); $\forall \nu_j, \nu_q \in \mathfrak{R}^0_\nu$ and $c_i \neq c_0$, $x_i \neq x_0$; the properties $(c_i, c_i)_j = no$ and $(x_i, x_i)_j = no$ are not defined and not considered further;

– the comparative constructions $(c_k, c_i)_j$ and $(c_g, c_r)_h$ are of the same type if $\nu_j = \nu_h$ and $c_i = c_r$;

– the $\forall v_j \in \mathfrak{R}^0_\nu$ construction

$$(c_k, c_i)_j = \begin{cases} c_i, \; if \; c_k = c_0, \\ c_0, \; if \; c_i = c_k = c_0, \end{cases}$$

is degenerated;

– $(c_0, c_i)^l\nu$ is a left-degenerated construction, and ${}^l v \in \mathfrak{R}^0_\nu$;

– similarly determined are the homogeneity of the generic properties of the elements x_k and x_i, with the degeneracy and the left degeneracy of the comparative properties.

Properties of comparative relations. For the relations $\nu_j, \nu_q \in \mathfrak{R}^0_\nu$ in the classes $\mathfrak{C}_0$ and $\mathfrak{X}_0$, true are properties that are given for the class $\mathfrak{C}_0$ (for the generic class, they are the same):

– freedom if the comparison $(c_k, c_i)_j$ allows the comparative relation $v'_j : (c_i, c_k)_{j'}$,

– the relation $v'_j \in \mathfrak{R}^0_\nu$ preserves the properties of the induction of the relation ν_j,

– the relation $\nu_j \in \mathfrak{R}^0_\nu$ in the generic and specific elements $c_k, c_i \in \mathfrak{C}_0$ determines the construction $(c_k(x_k), c_j(x_i))$ ν_j, with inducing comparison relations of φ^1_j and φ^2_j so that $((c_k, c_i)\varphi^1_j, (x_k, x_i)\varphi^2_j)$.

Theorem 2. Various instances of $c_k, c_i \in \mathfrak{C}_0$ coincide at least in one relation $v_j \in \mathfrak{R}^0_\nu$. The proof follows from the previous statement: $\mathfrak{C}_0 \cap \mathfrak{X}_0 \neq \varnothing$.

Theorem 3. The operator $\phi = \bullet$ is defined by $(\nu_i, \nu_j)\phi$ when and only when $\nu_i = {}^l\nu$.

Calculation of derivative relations. The relation $\xi_q \in \mathfrak{R}^0_c$ is true for both specific and generic comparative properties by the rules:

– $(\forall (c_i, c_k)_j, \exists \xi_q \in \mathfrak{R}^0_c, g \in \mathfrak{J}_0 \subset \mathfrak{J})$, $(((c_k, c_j)_j, g)\xi_q = ((c_k, c_j)_j, g)_q = (c_i, c_k)_j \multimap_q g)$, where g is the property indicator $(c_i, c_k)_j$ and $g := u$, $u \in \mathfrak{U}$; $c_k = (c_0, c_k)_j \multimap g$, where $g \neq g_0$ is an element of the class $\mathfrak{C}_0$; moreover, the property $(c_i, c_k)_j$ is unambiguous by the determined indicator, but it is multivalued by the values of the indicator;

– comparison is possible by the relation $v \in \mathfrak{R}^0_\nu$ $(g, c_k)v \multimap g$ so that the instance c_k has a property with the exponent g if the components of the relation are comparable;

– between the exponent of the property and its values there is a single semantic background that allows a comparative relation in the values of the elements $c_i := u_i$ and $c_k := u_k$;

– the relation $\xi \in \mathfrak{R}^0_c$ is satisfied only in the left composition $v \bullet \xi$;

– the composition of relations $\xi_i \bullet \xi_j \mapsto (v_i \bullet \xi_i) \bullet (v_j \bullet \xi_j)$ is permissible, and it is assumed that $c_0 = (c_0, c_0)_j \multimap g_0$.

The purpose of the relations ν and ξ determines the following.

Statement 3. For the classes of the derivative relations $\mathfrak{R}^0_c$ and comparing $\mathfrak{R}^0_\nu$, $\mathfrak{R}^0_c \cap \mathfrak{R}^0_\nu = \varnothing$.

The derivation relation has this property.

Theorem 4. The derivative relation $\xi \in \mathfrak{R}^0_c$ is homomorphism.

Indeed, the property $(c_i, c_k)_j$ is homomorphism; the expression $((c_i, c_k)_j)\xi = ((c_i)\xi, = ((c_k)\xi)_j$ does not change the property indicator g that follows from the formulary chain dependence $= ((c_i, c_k)_j)\xi = (((c_0, c_i)_r, (c_0, c_k)_r)_j)\xi = (((c_0, c_i)_r)\xi, ((c_0, c_k)_r)\xi)_j = (g, g)_j = g$.

The considered calculation allows using ontological instances to form semantic properties and their indicators. The comparative properties of the ontology are constructible; the kind of the property depends on the applied relation, and the simple property is defined on a pair of instances of the ontology. A derivational homomorphism of a simple property generates a property exponent, and the value of the exponent is multivalued.

5. 4. The generating of objects and classes of ontology by comparative relations

In the previous paragraph, we considered the calculation of property construction; now we will form simple concepts of knowledge, attributes of ontological objects by comparative relations of the zero level.

The zero order (rank) of relations or mappings $\beta_j \in Rf\mathfrak{K}$ is determined by their ranks not exceeding two. The objects constructed on the basis carrier M_0 of the OCS structure (2) by means of mappings and zero-order operators from the classes $\mathfrak{Y}^0$ and $\mathfrak{D}^0$ are called objects of the zero level and the zero order of the ontology construction.

Construction of zero-level objects. The formation of objects of the zero level and order by comparative relations is carried out according to the following rules:

– the zero-level ontology object b_{kij}, formed on the instances a_k and a_i by the comparative relations of ν_j, is given by the construction $b_{kij}=(a_k, a_i)_j$, $\forall a_k, a_i \in \mathfrak{C}_0 \mid \mathfrak{X}_0$; the objects b_{kij} and b_{qrm} are of the same type if the corresponding constructions of the comparative properties are the same; particular cases of the objects b_{kij} are the degenerated objects $b_{0ij}=(a_0, a_i)_j=a_k$ and the empty object $b_{00i}=b_0=a_0$;

– objects built on the class $\mathfrak{C}_0 \mid \mathfrak{X}_0$ with the help of the relations $v_j \in \mathfrak{R}_\nu^0$ form a constructive family of concepts $\mathfrak{F}_{\mathfrak{R}_v^0}^0$ of the zero level so that $\mathfrak{C}_0, \mathfrak{X}_0 \subset \mathfrak{F}_{\mathfrak{R}_v^0}^0$;

– if $\mathfrak{O}_0^0 \subset \mathfrak{O}$ is a free class of objects of the zero level and order, then $\mathfrak{F}_{\mathfrak{R}_v^0}^0 \vec{\subset} \mathfrak{O}_0^0$, where the symbol $\vec{\subset} \in \mathfrak{D}^0$ determines the operator of including the family into the class;

– $S_1^0 \subset S_2^0$, where $S_1^0 = \left\langle \mathfrak{C}_1, \mathfrak{R}_{\nu,1}^0 \right\rangle$, $S_2^0 = \left\langle \mathfrak{C}_2, \mathfrak{R}_{\nu,2}^0 \right\rangle$, $\mathfrak{C}_1, \mathfrak{C}_2 \subset \mathfrak{C}_0$ and $\mathfrak{R}_{\nu,1}^0, \mathfrak{R}_{\nu,2}^0 \subset \mathfrak{R}_\nu^0$ if:

– $\mathfrak{C}_1 \subset \mathfrak{C}_2$ and $\mathfrak{R}_{\nu,1}^0 \subset \mathfrak{R}_{\nu,2}^0$;

– the properties of relations in the class $\mathfrak{R}_{\nu,2}^0$ are preserved for relations of the class $\mathfrak{R}_{\nu,1}^0$.

Theorem 5. The taxonomy $S^0 = \left\langle \mathfrak{C}_0 \cup \mathfrak{X}_0, \mathfrak{R}_\nu^0 \right\rangle$ of the ontology $\mathfrak{K}_v^0$ does not allow the composition of relations of the class $\mathfrak{R}_\nu^0$.

The proof follows from the defining rules of calculating the relations $\nu \in \mathfrak{R}_\nu^0$.

Consequence 2. $\mathfrak{K}_v^0 \vec{\subset} \mathfrak{K}^0$.

Family and class of indicators. Derivative relations in families of objects of the zero level can be constructed as separate indicators and into their classes:

– $(\forall b_j \in \mathfrak{F}_{\mathfrak{R}_v^0}^0, \exists\, \xi_q \in \mathfrak{R}_c^0), (b_j \multimap_q g_j,\ g_j := u_k,\ g_j \in \mathfrak{J},\ u_k \in \mathfrak{U})$;

– $\forall \xi_q \in \mathfrak{R}_c^0,\ b_0 \multimap_q g_0,\ g_0 \in \mathfrak{J},\ g_0 := u_0$;

– a set of various indicators obtained by the relations $\xi_j \in \mathfrak{R}_c^0$ on all objects of the class $\mathfrak{F}_{\mathfrak{R}_v^0}^0$ to determine the family of indicators $\mathfrak{Z}_{\mathfrak{R}_c^0}^0 \subset \mathfrak{J}$;

– the final subclass of the set of indicators $\{g_i\} \subset \mathfrak{Z}_{\mathfrak{R}_c^0}^0$ specifies the property $p_i \subset \{g_i\}$; the various characteristics form a class of characteristics of the zero level $\mathfrak{L}_0^0 = \{p_j\}$;

– the classes $\mathfrak{R}_v^0$ and $\mathfrak{R}_c^0$ allow constructing the taxonomy system $\left\langle \mathfrak{C}_0 \cup \mathfrak{X}_0, \mathfrak{R}_\nu^0 \cup \mathfrak{R}_c^0 \right\rangle$ of the ontology $\mathfrak{K}_{v,\xi}^0$.

Methods for specifying properties.

Let us consider two ways to specify the characteristics of the class $\mathfrak{L}_0^0$.

1. The property is set by the expert (external executor).

2. The property is constructed algorithmically:

a) the ratio $v_j \in \mathfrak{R}_\nu^0$ is chosen, and a family of sets of indicators is formed: $(\forall a_i,\ a_k \in \mathfrak{C}_0 \cup \mathfrak{X}_0, (a_k, a_i)_j \multimap g_{ki}),\ p_j = \{g_{ki}\}$;

b) the pair of element $a_i, a_k \in \mathfrak{C}_0 \cup \mathfrak{X}_0$ is registered to construct the set of indicators $(\forall v_j \in \mathfrak{R}_{\nu,k}^0, \mathfrak{R}_{\nu,k}^0 \subset \mathfrak{R}_\nu^0,\ \#\mathfrak{R}_{\nu,k}^0 \le m,$ $m \in 1..\,n;\ (a_k, a_i)_j \multimap g_j),\ p_i = \{g_j\}$;

c) $\mathfrak{L}_0^0 = \{p_j\} \cup \{p_i\}$.

The above calculation allows constructing simple objects-properties, property indicators and their zero-level classes with the help of comparative relations and derivational correspondences on the basis carrier. The calculation also allows for the formation of a simple taxonomy of the ontology on a simple class of elements and relations.

5. 5. Actions and performers

The class of operators and operators-actions $\mathfrak{D}^0$ introduced in constructive structure (2) assumes that any class action is realizable by at least one internal performer of the class $\mathbf{Z}_v$. The composition of the class of performers is diverse: computers, automatic action models, etc., and some actions can be performed by several performers and their bundles. Let us consider some computational requirements on the class $\mathfrak{D}^0$:

– the class $\mathfrak{D}^0$ is simple, i.e. its components satisfy definition 2;

– $\#\mathfrak{D}^0$ and $\#\mathbf{Z}_v^u$ are finite; any operator-action $\circ \in \mathfrak{D}^0$, single $\circ^1$ or double $\circ^2$ and $Dom\ \mathfrak{D}^0 \subset \mathfrak{O}_0^0$; the relations and mappings of the class $Rf\,\mathfrak{K}$ are representable by algorithms for the execution of operators implemented by at least one performer of the class $\mathbf{Z}_v^u$;

– $(\forall \circ^1 \in \mathfrak{D}^0,\ \exists z \in \mathbf{Z}_v),\ ((b)\overset{z}{\circ} \vec{\in} \mathfrak{O}_0^0,\ b \in \mathfrak{O}_0^0)$ sets the action $\circ^1$ over the object b by the performer z, where $\vec{\in}$ is the operator of including the object in the class;

– $(\forall \circ^2 \in \mathfrak{D}^0,\ \exists z \in \mathbf{Z}_v),\ ((b_1, b_2)\overset{z}{\circ} \vec{\in} \mathfrak{O}_0^0,\ b_1, b_2 \in \mathfrak{O}_0^0);\ (z_j, z_k \in$ $\in \mathbf{Z}_v, (z_j, z_k)\overset{z}{\phi} \vec{\in} Z_v,\ z \in Z_v, \phi \in \mathfrak{D}^0)$;

– the implementation of any action $\circ \in \mathfrak{D}^0$ is unique to a certain performer, but it is possible by other performers;

– any action is implemented by an acceptable performer in a finite time;

– it is possible to have the composition of actions $\overset{z_j}{\circ}_j \bullet \overset{z_k}{\circ}_k$, where $\circ_j, \circ_k \in \mathfrak{D}^0$ so that $((\cdot)\overset{z}{\circ}_j, (\cdot)\overset{z}{\circ}_k)\overset{z}{\bullet}$;

– not for any operator $\circ \in \mathfrak{D}^0$ there is a reverse operator $\circ^{-1} \in \mathfrak{D}^0$;

– the action $\circ_0 \in \mathfrak{D}^0$ is identical if $b\overset{z}{\circ}_0 = b,\ b \in \mathfrak{O}_0^0$ and $z \in \mathbf{Z}_O^v$.

The actions of the operators are performed according to algorithmic instructions by the performers in a certain way. The calculations provide the general permissive capabilities of operators and performers necessary for the construction of algorithms, criteria, etc.

5. 6. Simple subordination relations

The component of the subordination relations $\mathfrak{R}_V^0 \subset \mathfrak{R}^0$ carries out the binding of objects of different nature into a single conceptual construction of subordination – the essence, thereby setting the main determining component and the determined subordinate component of the structure. A special case of the subordinate concept of knowledge is the property-concept attribute. Let us consider the calculation of subordination relations:

– if $\mathfrak{R}_V^0, \mathfrak{R}_v^0 \subset \mathfrak{R}^0,\ \mathfrak{R}_V^0 \cap \mathfrak{R}_v^0 = \varnothing,\ Dom\ \mathfrak{R}_V^0 = \mathfrak{C}_0 \cup \mathfrak{X}_0 \cup \mathfrak{J}_0$, then the subordination relations $\rho_j \in \mathfrak{R}_V^0$ are true for the arranged pairs (a_i, a_k);

– if $\forall a_i, a_k \in \mathfrak{C}_0 \cup \mathfrak{X}_0 | a_i \in \mathfrak{J}_0$ and $\forall \rho_j \in \mathfrak{R}_V^0$, the true rule is the following: $(a_i, a_k)\rho_j = ({}_{a_i}a_k)_j = {}_{a_i}^{\rho_j}a_k$;

– the representing of subordinations $({}_{a_i}a_k)_j$ and ${}_{a_i}^{\rho_j}a_k$ of the pair of elements (a_i, a_k) by the relation ρ_j is called the constructive concept of subordination (an ontological entity) that connects the main element a_k with the subordinate element a_i;

– the relation $\rho_j \in \mathfrak{R}_V^0$ can take into account the generic and specific definitions of the constructive objects, and in this case it acts by the following rule $(c_i(x_i), c_k(x_k))\rho_j=((c_i, c_k)\varphi_{j,1}),$ $(x_i, x_k)\varphi_{j,2})=,\ (({}_{c_i}c_k)_{j,1}, ({}_{x_i}x_k)_{j,2})$, where $\varphi_{j,1}$ and $\varphi_{j,2}$, inducing the subordinate relation ρ_j; the constructions $({}_{a_k}a_m)_j$ and $({}_{a_r}a_q)_i$ allow comparison by the ration ν by the main and subordinate elements;

– the family of objects $\mathfrak{F}_{\mathfrak{R}_V^0}^0 = \{b_j; (a_i, a_k)\rho_j = b_j,\ \forall \rho_j \in \mathfrak{R}_V^0,$ $a_i, a_k \in \mathfrak{C}_0 \cup \mathfrak{X}_0\ \&\ a_k \ne a_0\}$ determines the class of the subordination constructions so that $\mathfrak{C}_0 \cup \mathfrak{X}_0 \subset \mathfrak{F}_{\mathfrak{R}_V^0}^0,\ \mathfrak{F}_{\mathfrak{R}_V^0}^0 \setminus (\mathfrak{C}_0 \cup \mathfrak{X}_0) \vec{\subset} \mathfrak{O}_0^0,$ $\mathfrak{F}_{\mathfrak{R}^0}^0 = \mathfrak{F}_{\mathfrak{R}_v^0}^0 \cup \mathfrak{F}_{\mathfrak{R}_V^0}^0,\ \ \mathfrak{F}_{\mathfrak{R}^0}^0 \subset \mathfrak{O}_0^0$; it is possible to form the taxonomies of the ontologies $\mathfrak{K}_V^0$ and $\mathfrak{K}_{v,V}^0$ by the classes $\mathfrak{C}_0 \cup \mathfrak{X}_0 \cup \mathfrak{J}_0, \mathfrak{R}_V^0$ and $\mathfrak{C}_0 \cup \mathfrak{X}_0, \mathfrak{R}_v^0 \cup \mathfrak{R}_V^0$, respectively.

Properties of subordination relations. Subordination relations have the following properties:

– there is a left-side composition of the relations $\rho_i, \rho_j \in \mathfrak{R}_V^0$, $\rho_i \bullet \rho_j$;

– the associative property of a composition of subordination relations is not reasonable;

– the subordination relation $\rho_j \in \mathfrak{R}_V^0$, $(a_i, a_k)\rho_j$ allows an inverse relation $\rho'_j \in \mathfrak{R}_V^0$, $(a_k, a_i)\rho'_j$ if $a_i \neq a_0$ and $a_k \neq a_0$;

– any relation $\rho_j \in \mathfrak{R}_V^0$ constructs the identical concept $({}_{a_k}a_k)_j$ and the left degenerated entity $(a_0, a_m)^l\rho_j = a_m$.

Theorem 6. The relation $\rho_j \in \mathfrak{R}_V^0$ in the instances $a_i, a_k \in \mathfrak{C}_0 \cup \mathfrak{X}_0$ is of a weak morphism.

The calculation of subordination relations provides the instances of ontology with generic and specific concepts. The entities that are formed during the calculation form a class of subordination objects, and this class is replenished with zero-level objects. Subordination relations in conjunction with the instance class make it possible to construct a simple and complex conceptualization of ontology objects.

5. 7. Basic connection maps

Mappings of connections of different ontological instances form complex constructions. They are formally representable as one-dimensional or multidimensional list concepts. The result of the mapping can be heterogeneous structures endowed with actions, which, for example, helps form signs with actions that are necessary to generate the criteria for generating of knowledge.

Calculation of connection of ontology instances. The calculation of the class of maps $\mathfrak{A}^0$ in the domain *Dom* $\mathfrak{A}^0 = \mathfrak{C}_0 \cup \mathfrak{X}_0$ is the following:

– the mapping $\eta_j \in \mathfrak{A}^0$ of the connection of instances $a_i, a_k \in \mathfrak{C}_0 \cup \mathfrak{X}_0$ of an abstract ontology forms connecting entities by the rules:

1) $(a_k, a_m)\eta_j = (a_k, a_m)\varphi_j = b_{kmj}, \;\; b_{kmj} = (a_k a_m)_j;$

2) $$(a_k(x_k), a_m(x_m))\eta_j = \\ = ((a_k, a_m)\varphi_{j,1}, (x_k, x_m)\varphi_{j,2}) = \\ = ((a_k a_m)_{j,1}, (x_k x_m)_{j,2}) = b_{kmj}(x);$$

where the inducing mappings φ_j, $\varphi_{j,1}$, $\varphi_{j,2}$ are of the mappings $\eta_j \in \mathfrak{A}^0$;

– for the empty element $a_0 \in \mathfrak{C}_0 \cup \mathfrak{X}_0$, there happens a degeneration of the entities:

– $(a_0, a_i)\,{}^l\eta = a_i$ and $(a_k, a_0)\,{}^r\eta = a_k$, ${}^l\eta, {}^r\eta \in \mathfrak{A}^0$;

– $\forall \eta_j \in \mathfrak{A}^0$, $(a_0, a_0)\eta_j = a_0$;

– the $\mathfrak{F}_\eta^0 = \{b_j; (a_k, a_m)\eta_j = b_j, (\forall a_k, a_m \in \mathfrak{C}_0 \cup \mathfrak{X}_0) \; \& \; (\forall \eta_j \in \in \mathfrak{A}^0)\}$ family of connection structures and $\mathfrak{C}_0 \cup \mathfrak{X}_0 \subset \mathfrak{F}_\eta^0$, and $\mathfrak{F}_\eta^0 \setminus (\mathfrak{C}_0 \cup \mathfrak{X}_0) \overset{\rightarrow}{\subset} \mathfrak{O}_0^0$; $|b|$ is the length of the construction $b \in \mathfrak{F}_\eta^0$, $0 \leq |b| \leq 2$; if $b = a_0$, then $|b| = 0$; the family $\mathfrak{F}_\eta^0$ along the length of the structures is partially arranged orderly.

Theorem 7. The mappings of the class $\mathfrak{A}^0$ possess the properties of a strong morphism.

Indeed: the composition of the mappings $\eta_1 \bullet \eta_2$, $\eta_1, \eta_2 \in \mathfrak{A}^0$ is possible; the composition of the mappings is not commutative, but it is associative; ${}^l\eta$ and ${}^r\eta$ are the left and the right units of the class.

Extended connection mappings. Extended connection mappings can act in the basis classes $\mathfrak{C}_0$, $\mathfrak{X}_0$ and $\mathfrak{U}$ in combination with any symbols of the carrier of the OCS, for example, symbols of operators, relations, etc. Thus, if the symbol of the operator is $\circ \in \mathfrak{D}^0$ and $d_j \in \mathfrak{C}_0 \,|\, \mathfrak{X}_0 \,|\, \mathfrak{U}$, then the connection mapping $\eta_k \in \mathfrak{A}^0$ is true:

– the constructions $(\circ, d_j)\eta_k = \circ d_j$ or $(d_j, \circ)\eta_k = d_j\circ$ determine the object $b_{\circ,j}$; if $d_j = d_0$ is empty, the constructions $\circ d_j$ and $d_j \circ$ determine the object $b_{\circ,j} = \circ$;

– if $d_j \in \mathfrak{C}_0 \cup \mathfrak{X}_0$, then $\{b_{\circ,j}\} = \mathfrak{F}_{\eta,\circ}^0$ is a family of elementary entities with actions, and objects $b_{\circ,j}$ allow implementing an action by internal performers of the class Z_v, i. e. $\overset{z_k}{\circ} d_j$ and $d_j \overset{z_m}{\circ}$, where $z_k, z_m \in Z_v$; if $d_j \in \mathfrak{U}$, then $\overset{z_k}{\circ} d_j, d_j \overset{z_m}{\circ} \overset{\rightarrow}{\in} \mathfrak{U}$ and if $d_j \in \mathfrak{C}_0 \cup \mathfrak{X}_0$, then $\overset{z_k}{\circ} d_j, d_j \overset{z_m}{\circ} \overset{\rightarrow}{\in} \mathfrak{U}$;

– $\mathfrak{F}_{\eta,h}^0 = \{b_{k,h} : (h_i, h_q)\eta_k = b_{k,h}, h_i, h_q \in \mathfrak{C}_0 \cup \mathfrak{X}_0 \,|\, \mathfrak{D}^0 \,|\, D, \eta_k \in \mathfrak{A}^0\}$, where D is the class of various symbols of the OCS carrier; $\mathfrak{F}_{\eta,\circ}^0 \subset \mathfrak{F}_{\eta,h}^0$ and $\mathfrak{F}_{\eta,h}^0 \overset{\rightarrow}{\subset} \mathfrak{O}_0^0$.

Calculations of the connection mapping of the OCS structure allow constructing lists, tables, etc. of knowledge objects with included operators, which is important for the further organization of complex knowledge, complex properties, indicators, criteria, etc.

The above part of the calculation shows how the base class of zero-level objects $\mathfrak{O}_0^0$ is formed. The above calculation is the basis for forming the expansion of the taxonomy of ontology.

5. 8. The designing of objects of higher levels

The considered zero-order calculations and the generated class of zero-level objects allow us to construct objects, different relations and maps of higher levels. The proposed approach is constructively transparent and accessible to be implemented by an external performer, and the process of constructing the levels of objects is based on the iterative methodology of the formation of IO constructions. This technique is applicable for the task of forming classes of higher-level concepts.

Let the mappings $\beta_i, \beta_j \in \mathfrak{Y}^0$ and their dependence $(\beta_i, \beta_j)\phi$, $\phi \in \mathfrak{D}^0$ be determined on the class of objects $\mathfrak{O}_0^0$ of the zero order. It is necessary to use the mappings β_j to construct the class of objects $\mathfrak{O}_0^k$ of the *k*th level.

Definition 7. The ϕ-iteration in $(\beta_i, \beta_j)\phi = \psi_j$ of the initial objects $b_i^0 \in \mathfrak{O}_0^0$, formed by the mappings β_i, generates the IO objects $b_j^k \in \mathfrak{O}_0^k$ if it is possible to build the sequence $(b^0, b^1, ... b^{k-1}, b^k)$, which is directed and bound by the mappings $\psi_j : (b_r^s, b_q^s) \rightarrow b^{s+1}$.

While objects of the higher levels and their classes are formed, there are features that are reflected in the calculation:

– any class $\{\mathfrak{O}_0^i; i \geq 0\}$ contains an empty element $b_0 \in \mathfrak{C}_0 \cup \mathfrak{X}_0$;

– the admissibility of the mapping ψ_j on the objects $b_k, b_m \in \mathfrak{O}_0^{k-1}$ is determined by the two-valued predicate $\mathfrak{p}$;

$$(b_k, b_m)\psi_j = \begin{cases} (b_k, b_m)_j, & \text{if } \mathfrak{p}(b_k, b_m, \psi_j) = 1, \\ \text{no}, & \text{if } \mathfrak{p}(b_k, b_m, \psi_j) = 0; \end{cases}$$

– the ϕ-iteration does not change the properties of the objects on which it is defined, and there is a correspondence $(b_k, b_m)_j \multimap_h (g_k, g_m)_{j,h}$, where $b_k \multimap_{h_1} g_k$, $b_m \multimap_{h_2} g_m$;

– the mappings $\beta_j \in \mathfrak{R}_V^0 \,|\, \mathfrak{A}^0$ on identical objects construct identical entities $(b_k, b_k)_j$;

– for the mappings $\beta_j \in \mathfrak{R}^0 \,|\, \mathfrak{A}^0$ there happens a degeneration of the entities $(c_0, b_m)^l\beta_j = b_m$, where ${}^l\beta_j$ is the left unit of the mapping β_j.

Examples of higher-level constructions. Let us consider examples of varieties of formal entities as well as their use in forming constructions of classes of the *k*-th level $\mathfrak{O}_0^k$. One of these types of constructing is the ontological entities formed by the compositions $\phi = \bullet$ of the mappings $v \in \mathfrak{R}_v^0$ and $\rho \in \mathfrak{R}_V^0$ on objects of the class $\mathfrak{D}_0^{k-1}$:

$$f_1 = ({}_{(b_i, b_q)_k}(b_r, b_h)_m)_s, \quad f_2 = ({}_{b_i}(b_r, b_h)_m)_s,$$

$$f_3 = ({}_{(b_i, b_q)_k} b_r)_s, \quad f_4 = ({}_{b_q} b_h)_s, \ \ldots,$$

$$f_m = \left({}_{({}_{(b_i, b_q)_k} b_r)_w} ({}_{b_i}(b_r, b_h)_m)_p \right)_s, \ \ldots,$$

$$f_n = \left({}_{({}_{b_q} b_l)_j} \left({}_{({}_{b_r} b_h)_d} \left({}_{({}_{(b_i, b_q)_k} b_r)_w} ({}_{b_i}(b_r, b_h)_m)_x \right)_p \right)_n \right)_s \tag{3}$$

and so on.

The entities f_r of expression (3) are used to arrange construction families of knowledge networks so that

$$\mathfrak{F}_{\mathfrak{R}_\pi^0}^k = \left\{ f_r; (b_i, b_q)\beta_j = f_r, \ \forall(\beta_j \in \mathfrak{R}_\pi^0 \ \& \ b_i, b_q \in \mathfrak{D}_0^{k-1}) \right\},$$

where $(\pi = v_g, \beta_j = v_j) \mid (\pi = V_g, \beta_j = \rho_j)$.

The families $\mathfrak{F}_{\mathfrak{R}_\pi^0}^k$ allow establishing complex hybrid networks of ontological objects, such as $\mathfrak{F}_{\mathfrak{R}_g^0}^k = \mathfrak{F}_{\mathfrak{R}_{v,g}^0}^k \cup \mathfrak{F}_{\mathfrak{R}_{V,g}^0}^k$, for which $\mathfrak{F}_{\mathfrak{R}_g^0}^k \vec{\subset} \mathfrak{D}_0^k$.

Another type is the composition $\eta_i \bullet \eta_j$ that organizes the iterative process of constructing structures; if the objects $b_0, b \in \mathfrak{D}_0^{k-1}$ are the constructions $f_0 = b_0$, $f_1 = b$, then the set of entities

$$f_n = (f_{n-2}, f_{n-1})\eta_j, \tag{4}$$

forms the family of constructions of the connection $\mathfrak{F}_\eta^k$, where $\mathfrak{F}_\eta^k \vec{\subset} \mathfrak{D}_0^k$.

If the composition of the mappings $\eta_i \in \mathfrak{A}^0$ and $\beta_j \in \mathfrak{R}^0$ is considered, according to which the iterations are formed similarly to formulae (4), a hybrid family of the entities $\mathfrak{F}_{\eta,g}^k$ is constructed so that $\mathfrak{F}_\eta^k \subset \mathfrak{F}_{\eta,g}^k$; in particular, at $\beta_j \equiv \rho_j \in \mathfrak{R}_V^0$ and $\eta_i \bullet \rho_j$, we have the following:

1) $((b_1, b_2)\eta_i, b_3)\rho_j = ({}_f b_3)_j, \quad b_3 \neq b_0$,
2) $(b_1, (b_2, b_3)\eta_i)\rho_j = ({}_{b_1} f)_j$,
3) $((b_1, b_2)\eta_i, (b_3, b_4)\eta_m)\rho_j = ({}_{f_1} f_2)_j$,

where $b_1, b_2, b_3, b_4 \in \mathfrak{D}_0^{k-1}$.

The composition $\eta_i \bullet \rho_j$ in question specifies the construction of the entities that is directed and bound by the relation $(\prec)$; one of the entities can be of form (4), and their totality forms a hybrid family of entities $\{f_r\} \vec{\subset} \mathfrak{F}_{\eta,g}^k$.

Let $\circ \in \mathfrak{D}^0$ be an operator; then the relation $\rho \in \mathfrak{R}_V^0$ forms the object $b \in \mathfrak{D}_0^{k-1}$ as subordinated to the action $\circ \in \mathfrak{D}^0$ by the form ${}_b\circ\rho = {}_b^\rho\circ$, where ${}_b^\rho\circ \vec{\in} \mathfrak{D}^k$. The connection mapping $\eta_\circ$ also generates the constructions $(\circ, f^0)\eta_\circ = \circ f^0$, $(\circ, f^0)\eta_\circ = (\circ, (f_1, f_2)\eta)\eta_\circ$, where $f^0, f_1, f_2 \vec{\in} \mathfrak{F}_{\eta,g}^0 \mid \mathfrak{Z}_{\mathfrak{R}_\mapsto^0}^0$, in which the properties $p_i \subset \mathfrak{Z}_{\mathfrak{R}_\mapsto^0}^0$ are used to form given criteria. The mapping $\eta_\circ$ and the dependence $(\eta_\circ, \eta_j)\phi$ form an iterative procedure for constructing structures, and their totality forms the family $\mathfrak{F}_{\eta,\circ}^k$.

Complex ontological constructional entities entail a definite order of representation and performance that are organized using elements of the D symbol class by connection mappings. As a result, a family $\mathfrak{F}_{\eta,D}^k$ is generated, which, in conjunction with the families examined, forms a hybrid family of connected structures $\mathfrak{F}_{\mathfrak{A}^0}^k = \mathfrak{F}_{\eta,g}^k \cup \mathfrak{F}_{\eta,\circ}^k \cup \mathfrak{F}_{\eta,D}^k$, $\mathfrak{F}_{\mathfrak{A}^0}^k \vec{\subset} \mathfrak{D}_0^k$.

Thus, the study has shown how classes of entities of an arbitrary level are formed in the constructive ontological structure on the basic carrier and the base signature class. A visual diagram of the formation of level classes of objects is given in Fig. 1.

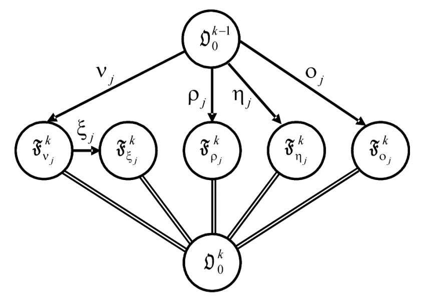

Fig. 1. A diagram for the formation of classes of objects of the k-th level of the zero order

Fig. 1 shows that the generation of the class $\mathfrak{D}_0^k$ by the class $\mathfrak{D}_0^{k-1}$ is performed by the mappings $\psi_j = (\beta_i, \beta_j)\phi$, provided that the map β_i acts on the class of objects $\mathfrak{D}_0^{k-1}$. An OCS constructs both ontology objects and new mappings. Objects are constructed on the carrier using mappings of the class $\mathfrak{R}^0$. The class $\mathfrak{R}^0$ $(v_j, \rho_j, \eta_j \in \mathfrak{R}^0)$ is expanded on the basis of constructing by means of mappings of the class $\mathfrak{D}^0$ $(\varphi \in \mathfrak{D}^0)$ and operators of the class $\mathfrak{D}^0$ $(o_j \in \mathfrak{D}^0)$. Thus, everything that is constructed on the class $\mathfrak{D}_0^{k-1}$ by the relations $(v_j, \rho_j, \eta_j \in \mathfrak{R}^0)$ and the operators $(o_j \in \mathfrak{D}^0)$ is included in the class $\mathfrak{D}_0^k$. Since φ, according to the above text of the paper, is a composition of the mappings η and ρ (compositions $\phi = \bullet$ of the mappings $v \in \mathfrak{R}_v^0$ and $\rho \in \mathfrak{R}_V^0$), what is the role of the mappings η_j and o_j in the generation of the class $\mathfrak{D}_0^k$ by the class $\mathfrak{D}_0^{k-1}$ described here? If Fig. 1 presents two ways of generating the class $\mathfrak{D}_0^k$ by the class $\mathfrak{D}_0^{k-1}$, then this should be noted in this text fragment.

5. 9. An example of forming the structure of a concept system of ontologies other than taxonomic

Means of OCSM are designed to automate structurally complex modeling areas, where a structure of the concept system, other than taxonomic, can be formed. In order to show the importance of such needs in constructive-synthesizing models, let us consider an example of solving problems of ontological support by means of OnCs in the processes of international cargo transportation (ICT) specified in [30]. Here one of the key management tasks is to select a scenario (Sh+) that provides ICT operations with minimal G(Sh+) risks. To solve the multicriteria task of planning ICT on the basis of the method of analytical hierarchies, models of direct and inverse (Fig. 2) processes of planning are presented, and they are given as abbreviated linear ordered structures, reflecting only the levels of the models. The focus of the models is the problem $G(Sh+)$, level U_1. In Fig. 2, in the direct planning process, the other levels mean the following: U_{12} – stages of the ICT procedures, U_{13} – actors, U_{14} – risks, and U_{15} – estimated scenarios; in the opposite direction: U_{22} – highlighted scenarios, U_{23} – problem-means, U_{24} – actors, U_{25} – actors' risks, and U_{26} – a set of scenarios.

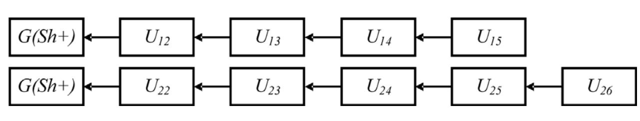

Fig. 2. Models of direct and reverse processes of international cargo transportation planning

Actors A_1, A_2 of the levels U_{13} and U_{24} belong to the universal set A_0, $A_1 \subseteq A_0$, $A_2 \subseteq A_0$. The same is true for scenarios $Sh_{15}, Sh_{22}, Sh_{26} \subseteq Sh_0$, nodes U_{15}, U_{22}, U_{26}, and risks U_{14}, $U_{25} \subseteq R_0$, which are included in the same sets but may not coincide. The analysis shows that the system of concepts corresponding to the model nodes cannot be represented by a taxonomy but will have loops, as shown in Fig. 3.

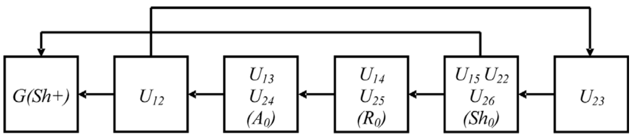

Fig. 3. A variant of the conceptual model of ontology for the support of ICT

Here the hierarchy of the levels of the objects in Fig. 2 is supplemented by the connections "Means" ← "Actors", and also G(Sh +) ← "Scenarios", forming a network. For the concepts of the models in Fig. 2 and Fig. 3, there is a partial intersection of the nodes; in the various control tasks in the ICT ontology, the elements of the nodes are not identical.

Another natural example of the need for universal means of forming network structures is the problem of designing and conceptual modeling of partial *n*-ary relations that connect more than two categories of concepts. Such a relation is represented, for example, by "Supply", including the classes "Cargo", "Owners", and "Carriers".

The developed procedures for the support of OCSM, in particular in the conceptual modeling of the concepts of the ICT area, ensure the implementation of such requirements, significantly expanding the area of information modeling of systems.

6. Discussion of the results of developing and researching ontological support for constructive-synthesizing modeling

Opportunities to achieve the stated goal of research have been implemented through the use of key methodological principles and standards for creating ontologies of subject domains. At the same time, the study used the currently known general methods of modeling complex objects and processes presented in IT, taking into account the specifics of goals, tasks and data aimed at different users. The possibility of reusing OCSM in a new environment, as well as in other applied OnSs, is provided because the focus of its attention is the task of creating powerful means of conceptual modeling.

The peculiarity of OCSM is determined by the specialization of information technologies proposed by the recursive principle of multilevel development of ontology concepts, starting from some zero level. Specialized are interpretation procedures and other actions that are performed using a class of internal and external performers.

The peculiarity of the presented study is the development of the CSM methodology and the means of its ontological support. This is due to the fact that CSM can be used uniformly to create a wide class of complex information systems. The presented developments are a continuation of the research using the means of constructive-synthesizing modeling to solve the following tasks:

– rational distribution of power for train traction recovery in DC systems [16];

– adaptation of data structures in RAM [17];

– improvement of data structures in problems of plagiarism detection [18];

– adaptation of compression algorithms to archived data [19];

– improvement of the process of ranking alternatives by the method of hierarchy analysis [20].

In terms of a shortcoming of the developed research on the ontological support of CSM, it should be noted that at the present time there is no practical implementation and application of software tools created on the basis of OCSM and their evaluation in comparison with other methods of creating modern intellectual IT. These issues require further consideration.

It should be noted that research and development are currently aimed at improving the existing automated control systems (ACSs) by Ukraine's freight traffic through a set of analytical servers developed on the basis of the CSM methodology. Their implementing requires the systematization and unification of the classes of the main analytical tasks of the automated control systems in the automation field, as well as the development and harmonization of conceptual models and their comprehensive approbation.

This study has suggested the possibilities for assessing the complexity of both information technology models and their applied ontological support. This line of research will simplify existing IT models and further develop design patterns, taking into account the simplest models and ontologies. Here, in order to obtain practically acceptable results, it is necessary to develop metrics for the complexity of IT in the chosen area of automation, as well as to provide the availability of a sufficient representative set for ACSs.

7. Conclusions

1. A methodology has been developed to support design-and-production modeling procedures using the applied system of ontological constructive-synthesizing modeling, OCSM, based on the constructivism paradigm with respect to all domain components, the revealed properties and unified models for representing notions, concepts and basic relations under CSM.

2. A single universal model of an ontological constructive structure has been devised to contain primary classes of ontology instances, active binding operators of actions and performers, classes of comparative properties, subordination and evolutionary development principles. At the same time, the corresponding specialized tools for implementing OCSM have been formed; they are the generation and construction of ontological objects of the zero and higher orders, the formation of their properties and indicators, and the binding of ontology concepts.

3. Within the framework of the methodology of OCSM, the study has developed models of the basic carrier of ontology instances, relations classes for the formation of properties of object classes, construction of individual properties and binding of concepts, basis maps of the connection, as well as procedures for constructing objects of the zero and higher levels. They are designed to automate structurally complex modeling areas that are distinguished by the uncertainty of the structure of conceptual models of systems of notions of subject areas as well as their distinctive features, providing also the possibility of forming systems of concepts other than taxonomic.

4. The properties of the main system-forming relations in the course of CSM have been revealed and studied. The research has proven that it is necessary to introduce constructed relations the structure of which is not known in advance, including recursive relations. The paper suggests models of generating ontology objects and classes by comparison relations as well as means and procedures for their implementation in OCSM.

References

1. Palagin, A. V. Ontologicheskaya kontseptsiya informatizatsii nauchnyh issledovaniy [Text] / A. V. Paladin // Kibernetika i sistemnyy analiz. – 2016. – Vol. 52, Issue 1. – P. 3–9.
2. Frye, L. TRIDSO: Traffic-based Reasoning Intrusion Detection System using Ontology [Text] / L. Frye, L. Cheng, J. Heflin // Journal of Research and Practice in Information Technology. – 2014. – Vol. 46, Issue 4. – P. 215–233.
3. Manuja, M. Intelligent text classification system based on self-administered ontology [Text] / M. Manuja, D. Garg // Turkish Journal of Electrical Engineering & Computer Sciences. – 2015. – Vol. 23. – P. 1393–1404. doi: 10.3906/elk-1305-112
4. Luke, S. Ontology-Based Knowledge Discovery on the World-Wide Web [Text] / S. Luke, L. Spector, D. Rager // AAAI Technical Report WS-96-06. – 1996. – P. 96–102.
5. Spinning the Semantic Web: bringing the World Wide Web to its full potential. [Text] / D. Fensel, J. A. Hendler, H. Lieberman, W. Wahlster (Eds.). – Mit Press, 2005. – 272 p.
6. Bechhofer, S. OWL: Web ontology language [Text] / S. Bechhofer // Encyclopedia of Database Systems. – Springer, 2009. – 80 p.
7. Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing? [Text] / T. R. Gruber // International Journal of Human-Computer Studies. – 1995. – Vol. 43, Issue 5-6. – P. 907–928. doi: 10.1006/ijhc.1995.1081
8. Noy, N. F. Ontology Development 101: A Guide to Creating Your First Ontology [Electronic resource] / N. F. Noy, D. L. McGuinness // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. – 2001. – Available at: http://protege.stanford.edu/publications/ontology_development/ontology101.html
9. Guarino, N. Formal ontology and information systems [Text] / N. Guarino // Proceedings of FOIS. – 1998. – Vol. 98. – P. 81–97.
10. Nardi, J. C. A commitment-based reference ontology for services [Text] / J. C. Nardi, R. de Almeida Falbo, J. P. A. Almeida, G. Guizzardi, L. F. Pires, M. J. van Sinderen et. al. // Information Systems. – 2015. – Vol. 54. – P. 263–288. doi: 10.1016/j.is.2015.01.012
11. Kazi, Z. Ontology-Based System for Conceptual Data Model Evaluation [Text] / Z. Kazi, L. Kazi, B. Radulovic, M. Bhatt // International Arab Journal of Information Technology. – 2016. – Vol. 13, Issue 5. – P. 542–551.
12. Alexopoulos, P. Utilizing Imprecise Knowledge in Ontology-based CBR Systems by Means of Fuzzy Algebra [Text] / P. Alexopoulos, M. Wallace, K. Kafentzis, D. Askounis // International Journal of Fuzzy Systems. – 2010. – Vol. 12, Issue 1.
13. Titenko, S. V. Ontologicheski-orientirovannaya sistema upravleniya kontentom informatsionno-uchebnyh Web-portalov [Text] / S. V. Titenko // Obrazovatel'nye tekhnologii i obshchestvo. – 2012. – Vol. 15, Issue 3. – P. 522–533.
14. Shynkarenko, V. I. Konstruktivno-produktsionnye struktury i ih grammaticheskie interpretatsii. I. Obobshchennaya formal'naya konstruktivno-produktsionnaya struktura [Text] / V. I. Shynkarenko, V. M. Il'man // Kibernetika i sistemniy analiz. – 2014. – Vol. 50, Issue 5. – P. 8–16.
15. Shynkarenko, V. I. Constructive modelling for zone of recovery energy distribution of dc traction [Text] / V. I. Shynkarenko, O. I. Sablin, O. P. Ivanov // Science and Transport Progress. Bulletin of Dnipropetrovsk National University of Railway Transport. – 2016. – Issue 5 (65). – P. 125–135. doi: 10.15802/stp2016/84036
16. Shinkarenko, V. I. Konstruktsionno-produktsionnaya model' struktur dannyh na logicheskom urovne [Text] / V. I. Shynkarenko, V. M. Il'man, G. V. Zabula // Problemi programuvannya. – 2014. – Issue 2-3. – P. 10–16.
17. Shynkarenko, V. I. Konstruktivno-produktsionnaya model' grafovogo predstavleniya teksta [Text] / V. I. Shynkarenko, E. S. Kuropyatnik // Problemi programuvannya. – 2016. – Issue 2-3. – P. 63–72.
18. Shynkarenko, V. I. Modelirovanie protsessa adaptatsii algoritmov szhatiya sredstvami konstruktivno-produktsionnyh struktur [Text] / V. I. Shynkarenko, T. N. Vasetskaya // Kibernetika i sistemniy analiz. – 2015. – Issue 6. – P. 19–34.
19. Shynkarenko, V. I. Modelirovanie protsessa ranzhirovaniya al'ternativ metodom analiza ierarhiy sredstvami konstruktsionno-produktsionnyh struktur [Text] / V. I. S Shynkarenko, T. N. Vasetskaya // Matematychni mashyny ta systemy. – 2016. – Issue 1. – P. 39–47.
20. Zholtkevich, G. N. Predstavlenie poluskhem predmetnyh oblastey informatsionnyh sistem sredstvami relyatsionnyh baz dannyh [Text] / G. N. Zholtkevich, T. V. Semenova, K. A. Fedorchenko // Visnyk Khark. nats. un-tu im. V. N. Karazina. Seriya: Matematychne modeliuvannia. Informatsiyni tekhnolohiy. Avtomatyzovani systemy upravlinnia. – 2004. – Vol. 629, Issue 3. – P. 11–24.
21. Pancerz, K. Ant-based extraction of rules in simple decision systems over ontological graphs [Text] / K. Pancerz, A. Lewicki, R. Tadeusiewicz // International Journal of Applied Mathematics and Computer Science. – 2015. – Vol. 25, Issue 2. doi: 10.1515/amcs-2015-0029
22. Pancerz, K. Paradigmatic and Syntagmatic Relations in Information Systems over Ontological Graphs [Text] / K. Pancerz // Fundamenta Informaticae. – 2016. – Vol. 148, Issue 1-2. – P. 229–242. doi: 10.3233/fi-2016-1432
23. Grabusts, P. Ontology-Based Classification System Development Methodology [Text] / P. Grabusts, A. Borisov, L. Aleksejeva // Information Technology and Management Science. – 2015. – Vol. 18, Issue 1. doi: 10.1515/itms-2015-0020

24. Kogut, P. UML for ontology development [Text] / P. Kogut, S. Cranefield, L. Hart, M. Dutra, K. Baclawski, M. Kokar, J. Smith // The Knowledge Engineering Review. – 2002. – Vol. 17, Issue 01. doi: 10.1017/s0269888902000358
25. Bova, V. V. Komp'yuternaya ontologiya: zadachi i metodologiya postroeniya [Text] / V. V. Bova, D. V. Leshchanov, D. Yu. Kravchenko, A. A. Novikov // Informatika, vychislitel'naya tekhnika i inzhenernoe obrazovanie. – 2014. – Issue 4. – P. 44–55.
26. Thomsen, E. Ontological Support for Living Plan Specification, Execution and Evaluation [Text] / E. Thomsen, F. Read, W. Duncan, T. Malyuta, B. Smith // Semantic Technology in Intelligence, Defense and Security (STIDS), CEUR. – 2014. – Vol. 1304. – P. 10–17.
27. Gonen, B. Ontological Support for the Evolution of Future Services Oriented Architectures [Text] / B. Gonen, X. Fang, E. El-Sheikh, S. Bagui, N. Wilde, A. Zimmermann // Transactions on Machine Learning and Artificial Intelligence. – 2014. – Vol. 2, Issue 6. doi: 10.14738/tmlai.26.784
28. Grishin, M. V. Tools ontological support of the design template equipment in the aircraft production [Text] / M. V. Grishin, S. N. Larin, P. I. Sosnin // V Mire Nauchnykh Otkrytiy. – 2015. – Issue 4. – P. 10. doi: 10.12731/wsd-2015-4-1
29. Breitsprecher, T. Towards Ontological Support for Principle Solutions [Text] / T. Breitsprecher, M. Codescu, C. Jucovschi, M. Kohlhase, L. Schröder, S. Wartzack // Mechanical Engineering. InFOIS. – 2014. – Vol. 5. – P. 427–432.
30. Khalipova, N. V. Vektorna optymizatsiya v zadachakh udoskonalennia mizhnarodnykh transportnykh system [Text] / N. V. Khalipova. – Dnipropetrovsk: Universytet mytnoi spravy ta finansiv, 2016. – 271 p.

UDC 519.216

DOI: 10.15587/1729-4061.2017.118265

TECHNOLOGY AND SOFTWARE TO DETERMINE ADEQUATE NORMALIZED CORRELATION MATRICES IN THE SOLUTION OF IDENTIFICATION PROBLEMS

U. Sattarova

Doctor of Philosophy in Computer Science, Associate Professor

Department of Information technologies and systems

Azerbaijan University of Architecture and Construction

Ayna Sultanova str., 11, Baku, Azerbaijan, AZ 1073

E-mail: ulker.rzaeva@gmail.com

Показано, що при класичному підході при нормуванні оцінок в діагональних елементах кореляційних матриць відсутні похибки від перешкод, а в інших же елементах ця похибка, навпаки, виникає. В результаті поліпшення обумовленості матриці від переходу до нормованих кореляційних матриць не спостерігається. Пропонуються технологія, софт для усунення цього недоліку і аналізу обчислювальних експериментів

Ключові слова: реальний сигнал, перешкода, кореляційна функція, нормована кореляційна матриця, вхідний – вихідний сигнал

Показано, что при классическом подходе при нормировании оценок в диагональных элементах корреляционных матриц отсутствуют погрешности от помех, а в остальных же элементах эта погрешность, наоборот, возникает. В результате улучшения обусловленности матрицы от перехода к нормированным корреляционным матрицам не наблюдается. Предлагается технология, софт для устранения этого недостатка и анализа вычислительных экспериментов

Ключевые слова: реальный сигнал, помеха, корреляционная функция, нормированная корреляционная матрица, входной – выходной сигнал

1. Introduction

It is known [1] that one of the main challenges in solving problems of automated control of industrial facilities is establishing the quantitative interrelations between input and output noisy signals characterizing the processes in those objects both in statics and dynamics. Such interrelations are called static and dynamic characteristics, respectively. These characteristics can be determined from differential equations of control objects. However, those differential equations are often unknown, which is why statistical methods are widely used – they make it possible to determine dynamic characteristics during normal operation of objects [1–3]. In practice, such dynamic characteristics as impulsive admittance and transfer functions of linear systems are determined by applying to their input artificial stimulation of a certain type (impulse, step function, sinusoids) and measuring the response. However, in that case, random uncontrollable disturbances are superimposed on these impacts. As a result, it proves impossible to precisely determine dynamic characteristics based on typical input signals [1–3].

2. Literature review and problem statement

The statistical correlation method for determining these dynamic characteristics is based on the solution of an integral equation that includes the correlation functions $R_{XX}(i\Delta t)$ and $R_{XY}(i\Delta t)$ of the input $X(i\Delta t)$ and output $Y(i\Delta t)$ signals. It allows us to obtain the dynamic characteristics of an object without disturbing its normal operation mode. Therefore, statistical