Semantic models, method and tools of knowledge bases coordinated development based on reusable components

Irina Davydenko
Belarusian State University
of Informatics and Radioelectronics
Minsk, Belarus
ir.davydenko@gmail.com

***Abstract*—The paper considers the approach to the development of knowledge bases based on semantic models. The problem of collective use of various types of knowledge applied for solving complex problems is considered. To solve the problem was proposed the knowledge base and methods and tools for developing knowledge bases. Feature of this approach is the coordinated development of knowledge bases of developers, as well as the use of reusable components.**

***Keywords*—semantic network; knowledge base; knowledge base structuring; coordinated development; intelligent system**

I. Introduction

The evolution of information technologies has led to the accumulation of large volumes of heterogeneous, weakly structured information. For effective use of this information to solve various probelms, it is necessary to create knowledge bases, which allow to systematize the knowledge stored in them, and also provide an opportunity to work effectively with them for both computer system and human. The emergence of knowledge bases led to the emergence of specialized models and knowledge representation languages, as well as software tools that allow to develop knowledge bases, including distributed development teams. The volume and heterogeneity of the information stored in the knowledge base leads to the need for its structuring, i.e. the allocation of interrelated fragments in it in order to improve the efficiency of its processing, as well as for didactic purposes.

The knowledge base is a key component of such class of computer systems, as knowledge-based systems, the development of which is one of the promising areas in the field of artificial intelligence [1]. The quality of the developed systems of this class is determined, inter alia, by the quality of the knowledge base and by the variety of types of knowledge stored in it. Knowledge-based systems are now used in a wide variety of areas of human activity - medicine, automation and production management, training, design automation tools and many others [2].

In this article, problems in the knowledge bases development from the perspective of providing the learnability of knowledge-based systems as a necessary property of intelligent systems will be considered. To solve these problems, a semantic model of the knowledge base is proposed that provides the coordinated use of different types of knowledge and models of their representation, the ability to present multilevel meta-knowledge and the knowledge base structuring, as well as the method and tools for knowledge bases development based on this model.

II. Problems in the field of knowledge engineering

The value and quality of the processed information is determined by the possibility of its effective use in a variety of models of problem solving. In turn, the problem solving models themselves should be able to use any information stored in the knowledge base, which can be useful for solving each specific problem.

Expanding the scope of knowledge-based systems has led to the need to support the solution of complex problems. Under the complex problem we will understand the problem, the solution of which involves the use of formalized knowledge of various types and different models of their processing, which in turn requires the compatibility and integration of the knowledge used, as well as models for their processing.

Examples of complex tasks include the following:

- the task of understanding of the meaning of the text of a natural language;
- the task of understanding of the meaning of the handwritten natural language text;
- the task of understanding of the meaning of a voice message;
- the task of semantic analysis of the image;
- the task of automating adaptive learning;
- the task of planning behavior in intelligent robots;
- the task of complex automation of various enterprises.

Nowadays the main approach to the implementation of systems capable of complex problems solving is the development of hybrid systems, one of the main problems in the development of which is the collective use of different types of knowledge in the interests of compensating for deficiencies and combining the advantages of heterogeneous models to solve the problem [3]. In addition, the use of various types of knowledge in problems solving allows to increase the number of classes of problems solved in comparison with the number of classes of problems solved by systems using a limited set of knowledge types. Such an increase is achieved through the combination of different approaches in solving the sub-problems within the problem, which becomes possible due to the unification of different types of knowledge representation within the same system. The relevance of this problem in the case of databases is considered in [4]. As a consequence, the actual task is to develop a common unified formal basis for different types of knowledge representation within a single system and ensuring their collective use in complex problems solving. This work focuses on the problem of ensuring the compatibility of various types of knowledge, including meta-knowledge, which is proposed to be solved by unifying the representation of different types of knowledge within the same knowledge base [5], [6], [7], [8], [9].

The work is devoted to solving of problems related to the development of models, methods and tools for creating knowledge bases of computer systems capable of complex problems solving.

Let's consider in more detail a complex problem on an example of a problem of automation of the batch production enterprise.

Complex automation of the enterprise requires the following information in the appropriate system:

- equipment nomenclature and configuration (to solve the problems of the current situation analyzing and the planned changes simulating, to evaluate the efficiency of the current equipment configuration, to analyze the possibility of its optimization);
- generalized business processes (description of products, stages of production);
- production plans (list and volume of products);
- current business processes (state of specific equipment, current production stage);
- specifications of emergency situations, ways of their elimination;
- current processes related to logistics;
- information on personnel, accounting
- etc.

The description of all listed types of knowledge in one system is necessary because they are closely related to each other and often describe the same objects, but from different aspects. In order for this description to work within the same system, all fragments must be consistent among themselves, in addition, the description should be structured in various aspects. For example, a description of the same department in terms of current production processes, logistics, personnel, emergency situations - different views on the same complex object.

Various kinds of meta-information about the same objects are used to solve various production tasks. For example, in the event of an emergency situation in the workplace, different users may have various questions to the automation system, the answer requires the presence of a different kind of knowledge in the system (Figure 1).

In order for the company to adapt to the market requirements, modern technologies, the situation with supplies and logistics, changes in legislation, it is permanently required to supplement the knowledge base of the automation system with new types of knowledge, including new concept systems (new workshop, new equipment, new type of manufactured products, new laws, etc.) directly in the operation of such a system (Figure 1). In other words, an intelligent system capable of complex problems solving must be *learnable*. By ***learnability*** in this case we will understand the ability of the system to acquire new knowledge and skills in the process of its operation, while preserving the correctness and integrity of the knowledge base. In turn, the learnability causes the requirement of ***flexibility*** to such a knowledge base, that is, reducing the complexity of making changes to the knowledge base.

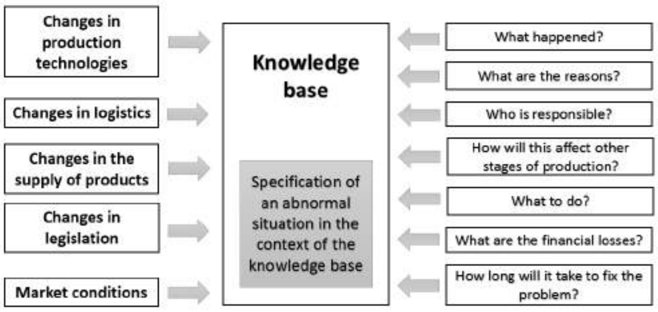


Figure 1. Examples of requests for a production automation system and factors affecting the knowledge base changes

The foregoing allows us to formulate requirements to the knowledge base of systems capable of complex problems solving:

- The possibility of coordinated use of different types of knowledge within the same knowledge base, including, during complex problems solving.
- The knowledge base must have a structure that takes into account the different aspects of the described entities specification.
- Possibility of description in the knowledge base of meta-knowledge (systematization and structuring of the knowledge base requires advanced tools of meta-description, i.e. tools of transition from the description of knowledge to the description of metaknowledge).
- convenience of the knowledge base processing, which implies, among other things, the possibility of narrowing and expanding the scope of problem solving, if necessary, as well as the ability to take into account meta-information of various kinds during processing.
- flexibility of the knowledge base, i.e. the possibility of adding to the knowledge base of new fragments, including new types of knowledge, without making changes

to the existing knowledge base structure directly in the process of system operation.
- in order for the system to be able to analyze and, among other things, verify and optimize its own knowledge base and processes of its development, the structure of this knowledge base, the history of its changes and plans for its evolution should be described by the same means as the domain part of knowledge base. The ability of the system to analyze its own components, in particular, the knowledge base, will be called **reflexivity**.

The creation and evolution of a knowledge base that meets these requirements is a labour intensive and time-consuming process that requires a high level of knowledge engineers experience, which entails a deficiency of specialists in the field of knowledge engineering and a high cost of knowledge bases and, consequently, knowledge-based systems.

Taking into account the requirements for knowledge bases, we will also formulate the requirements for the technologies for their development:
- provide the ability to create, using the specified models, tools and methods, knowledge bases that meet the above requirements and are compatible with each other;
- reduce labor intensity and shorten the development time of the knowledge bases;
- ensure high rates of evolution of models, tools and methods for the knowledge bases development;
- reduce the requirements for knowledge bases developers by providing information support.

To date, theoretical studies in the development of formal knowledge representation models and formal models for problems solving have led to a wide variety of such models.

Nowadays, there are dozens of knowledge representation models, but most of them are based on the main four: semantic networks, frames, production and logical models [10].

Each of the described models has its advantages and disadvantages, and when developing knowledge bases, the knowledge engineer must make a choice in favor of the most suitable model, because different types of knowledge require the use of different models of knowledge representation. The existing tools for knowledge bases development, as a rule, are focused on the use of one of the above-mentioned models. However, when creating knowledge-based systems, especially when solving complex problems, it becomes necessary to represent different types of knowledge within the same knowledge base, especially knowledge at the intersection of sciences, which none of the above-mentioned models taken separately can provide [7], [6]. In addition, the choice of the knowledge representation model limits a variety of problem solving models that can be used in the developed system.

Due to that, there is a need to create a universal model of knowledge representation that would make it possible to represent any kind of knowledge in a unified, easy-to-process form. Approaches to solving this problem are considered in the papers [6], [11], [12].

However, for coordinated use of the different types of knowledge in solving complex problems, it is not enough to develop only a knowledge representation model itself, it is necessary to develop unified methods of representation for each kind of knowledge that are agreed upon within the chosen model of knowledge representation.

Each knowledge representation model corresponds to a set of knowledge representation languages that implement these models.

In the knowledge representation languages, as a rule, the syntactic and semantic components are divided. The syntax specifies the rules by which the construction of the given language are built, and the semantics determines the rules for the specified constructions interpreting.

At the moment, different languages for knowledge representation are used: CycL [13] (the knowledge representation language based on ontologies and used within the Cyc project), IDEF5 [14] (Integrated Definitions for Ontology Description Capture Method - an ontological research standard for visualizing data, obtained as a result of processing ontological queries in a simple natural graphical form), Prolog [15] (predicate based language of mathematical logic of Horn clauses, which is a subset of the first order predicates logic), CLIPS [16](C Language Integrated Production System - a language of knowledge representation based on logical rules that use the same name program shell for expert systems creating), and others.

Particular attention should be paid to the tools offered in the scope of the Semantic Web, because of their thoroughness and prevalence. This direction is actively developed by the W3C consortium, whose main task is the development of standards for the Semantic Web [17], [18].

The means of the Semantic Web approach are a set of methods and technologies designed to present information in a form suitable for machine processing. Information is presented in the form of a semantic network, specified by means of ontologies. Standardization of information representation allows a computer system to obtain various factual information and make logical conclusions based on it. The use of W3C standards in the development of intelligent applications in recent decades has become very popular.

As part of the Semantic Web for the presentation of knowledge, the following have been developed:
- Resource Description Framework (RDF) and languages that provide RDF data representation;
- RDF Schema metadata presentation tools;
- principles of representation of knowledge in the form of ontologies and ontology description languages (OWL Lite, OWL DL, OWL Full);
- the SPARQL query language for RDF data storing;
- and a number of other standards.

Efficient knowledge storages based on RDF are also developed, such as Sesame, HyperGraphDB, Neo4j, Virtuoso, AllegroGraph, which provide storage and access to data using the SPARQL query language. To edit ontologies created on the basis of W3C standards, a large number of editors have been created that have a fairly wide functionality [19].

The knowledge representation tools offered in the Semantic Web approach have some disadvantages, one of which is their limitations, in particular, the lack of the ability to describe metacommunications, the means of fuzzy knowledge describing, the inability to describe the properties of entire entity classes, the impossibility of exceptions to certain rules describing and other [20].

Despite the successes achieved in the field of creating knowledge bases, there are the following problems:

- laboriousness of simultaneous use of models representing various types of knowledge;
- incompatibility of already developed components of knowledge bases leads to the need to re-develop existing solutions;
- changes to the knowledge base may necessitate significant changes in the structure of the knowledge base, especially in the case of dynamic knowledge bases;
- despite the existence of sufficiently developed tools of knowledge bases development, they do not fully provide comprehensive support (including information) to the team of developers at all stages of the knowledge base development, and also do not have sufficient flexibility and extensibility;
- existing facilities are oriented, as a rule, to a specific knowledge storage format, which makes it difficult to transfer the already developed knowledge base to another model interpretation platform.

The main reason for all these problems is the lack of unification of the representation of various types of knowledge, including meta-knowledge, within the same knowledge base, and the lack of a knowledge base structuring model that allows to structure the knowledge base on various aspects simultaneously.

III. The proposed approach

Among the approaches to solving these problems are the following:

- development of models to represent different types of knowledge, development of standards for knowledge representation;
- development of knowledge base structuring models that would allow to separate independent fragments of the knowledge base in such a way as to minimize the number of approvals in the collective development of knowledge bases and, as a result, to reduce the complexity of developing and improving of the knowledge base;
- ensuring the compatibility of the already developed knowledge base components for their reuse, and the creation of libraries of such components in order to reduce labor costs and shorten the development of knowledge bases;
- use of automation and information support tools for developers, focused on the collective development of knowledge bases, providing integrated support for all development stages to reduce labor costs, shorten development time, and reduce requirements for knowledge base developers.

However, existing implementations of these approaches are usually aimed at solving of any one of these problems and do not take into account the need to solve all of these problems in a complex.

The approach proposed in this paper is based on the idea of systems constructing based on semantic networks, proposed in [21]. These ideas served as the basis for the creation of ***OSTIS Technology***, which is a complex of models, tools and methods for intelligent systems development, as well as for the permanent updating and improvement of this technology.

OSTIS technology is based on using as a method of encoding information of unified semantic networks with a basic set-theoretic interpretation of their elements. This method is called ***SC-code*** (*Semantic Code*), and semantic networks represented in *SC-code* are called *sc-graphs* (*sc-texts* or texts of SC-code). Elements of such networks, represented in the SC-code, will be called *sc-elements*, in turn, the nodes of such networks will be called *sc-nodes*, the connections between them are *sc-connectors* (sc-arcs, sc-edges).

A key feature of *SC-code* is the joint use of the mathematical apparatus of graph theory and set theory. This allows, on the one hand, to ensure the strict and versatility of formalization tools, on the other hand - to ensure the convenience of storing and processing of information presented in this form.

For visualization of *SC-code* texts, external languages such as *SCg* (Semantic Code graphical), *SCn* (Semantic Code natural) and *SCs* (Semantic Code string) are used [**?**].

The model of an entity recorded by *SC-code* means is called the *sc-model of the specified entity*.

One of the features of the *sc-model of the knowledge base* is the lack of synonymy between sc-elements, which should be identified and eliminated by gluing together synonymous sc-elements, or by explicitly indicating the fact of their synonymy. [12]

To solve the problem of coordinated use of different types of knowledge within the knowledge base, compatibility and flexibility of knowledge bases, as well as high labor costs and long terms of their creation, it is proposed to develop:

- A unified semantic model of the knowledge base that ensures the unification of the representation of various types of knowledge and the possibility of using a wide range of knowledge base structuring types through:
  - internal representation of the knowledge base in the memory of the intelligent system in the form of a semantic representation using formalized semantic network;
  - separation of a hierarchical system of subject domains and an explicit representation of ontologies that describe the semantics of all subject domains and corresponding languages in the knowledge base.

  Use of these principles will ensure the possibility of coordinated use of different types of knowledge, the compatibility of the knowledge bases being developed among themselves, as well as their flexibility.

- The method of coordinated development of knowledge bases built on the specified model, based on the formal ontology of the design actions of knowledge base developers, implementing a component approach and oriented towards the collective development of knowledge bases. The presence of such a method will ensure the correctness and consistency of the project activity of developers directly in the process of the knowledge base use.
- a library of reusable components of knowledge bases, including components search tools based on their specifications. The presence of such a library will significantly reduce the labor costs for the development of knowledge bases.
- tools of automating the activity of knowledge base developers, as well as their information support, including the knowledge base development automation system and the developer consulting services subsystem within the IMS metasystem that implement the proposed method and provides coordination, verification and editing of knowledge base fragments directly in the process of using it. The availability of such tools will reduce the time required to create knowledge bases and requirements for their developers. These tools are expected to be developed using OSTIS Technology, which in turn will ensure flexibility of the tools themselves.

IV. Semantic model of the knowledge base

To implement the proposed approach, it is necessary:

1) to provide a unified basis for the presentation of various types of knowledge, which involves the development of entity specification tools in the knowledge base within a certain ontology of representation;
2) to provide the possibility of an unrestricted transition from knowledge to meta-knowledge due to the possibility of allocating of some integral fragment of the knowledge base and considering it as a specification object;
3) to construct a formal model of knowledge, identify the typology of knowledge;
4) to construct formal models of the most important types of knowledge, which are the basis for the specification of different kinds of entities.

From a formal point of view, the sc-model of the knowledge base is a set of sc-elements.

In order to transform a different kind of knowledge stored in the memory of a computer system into a single well-structured knowledge base, it is necessary to bring all these diverse types of knowledge to a *common syntactic and semantic foundation* based on some *universal ontology of representation*. In this paper, the ***ontology of sc-elements*** is the role of such an ontology of the representation, within the framework of which the typology of entities described in the knowledge base is specified, as well as the typology of signs included in the knowledge base, which reflects the character of the relationship of these signs with the current state of the knowledge base. An example of the specification of entities using the *ontology of sc-elements* in the knowledge base is shown in Figure 2.

The development of such an ontology allows us to describe the syntactic and semantic properties of *sc-elements* (that is, the signs of entities described in the knowledge base) within a single knowledge base, which in turn allows us to provide a property of *reflexivity* for systems based on the proposed approach. In particular, the ontology under consideration allows us to describe the properties of not only the objects of the external world, but also of the internal signs (*sc-elements*) themselves. For example, when describing objects in a dynamic domain, it is necessary to describe, on the one hand, the temporal properties of the entities themselves (*a past entity*, *a present entity*, *a future entity*), on the other hand, the temporal properties of the signs relative to the current state of the knowledge base (*a sign represented in the current state of the knowledge base*, *a sign not represented in the current state of the knowledge base*, etc.).

Existing approaches to the development of knowledge bases are based on the examination of specific elements of the knowledge base (classes, instances, relations, etc.) as objects of the specification. However, with the accumulation of large amounts of information in the knowledge base, it becomes necessary to allocate entire fragments of the knowledge base and be able to specify them, treating them as separate entities. This is necessary to ensure the possibility of an unrestricted transition from knowledge to meta-knowledge. In the framework of this paper, such a fragment of the knowledge base is called a ***structure*** (*sc-structure*). Each ***structure*** is a *sc-element*, denoting some text of *SC-code*, which can later be a specification object, including being part of other structures, be connected with other entities by different relations.

From a formal point of view, the *structure* is treated as a set, the elements of which are all the *sc-elements* that make up the fragment of the knowledge base designated by the given structure.

For the specification in the knowledge base of various structures, their typology was developed and the roles of the elements that make up the structure were determined.

The fact that as a formal basis for representation of knowledge in the *SC-code* the theory of graphs and set theory are used, allows us to analyze not only the external connections of the considered fragment with other elements of the knowledge base, but also to analyze the internal structure of these fragments with the necessary degree of detail, i.e. identify in the knowledge base the analogies, similarities, differences, build different types of correspondence between fragments.

In one of the *SC-code* external representation languages - the *SCg* - *structure* can be represented by explicitly indicating all the pairs of elements membership to the *structure* (see Figure 3a), and also as a contour containing all the elements that make up the structure (see Figure 3b).

The concept of *structure* is a formal basis for the *semantic*

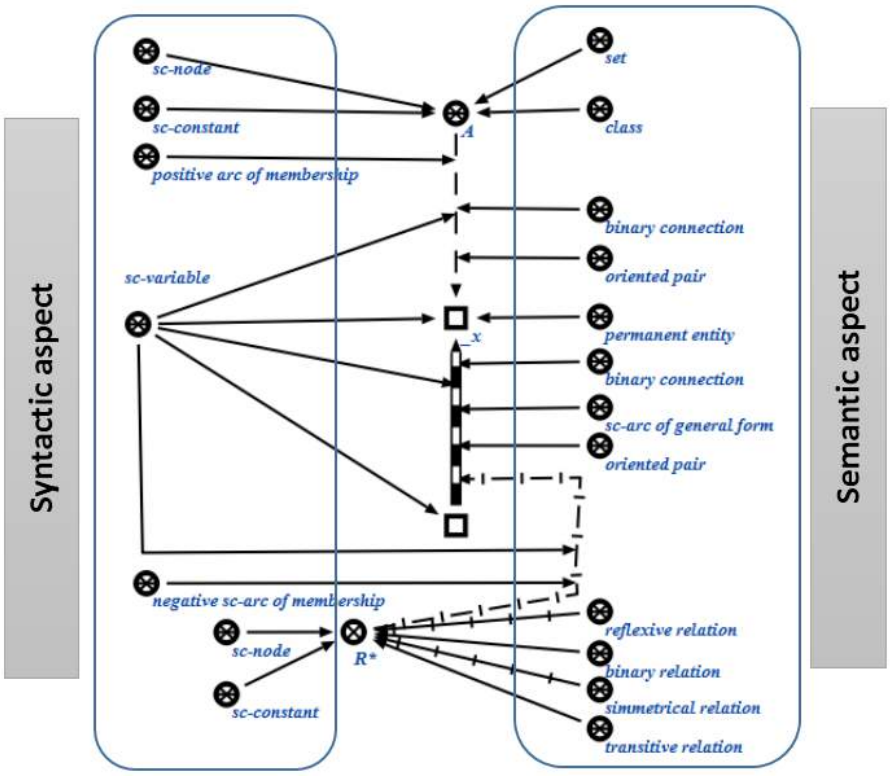


Figure 2. Syntactic and semantic aspects of the description of entities stored in the knowledge base

model of the knowledge base and models for its structuring in various aspects. In the works [22], [23], various relations defined on the structures are considered.

On the basis of the concept of structure, the concept of ***knowledge*** as the most important kind of entities described in the knowledge base is clarified.

Within the proposed knowledge base model, each *structure* will be called ***knowledge*** if and only if:

- for any connection (including the *sc-connector*) that is part of this structure, this structure includes all components of this connection with explicit indication of the membership of the specified components in the specified connection;
- for any sign of the final structure included in the structure under consideration, this structure also includes all components of this finite structure with explicit indication of the membership of these components in the specified final structure;
- for any sc-element that is part of the structure under consideration, this structure includes the signs of all concepts for which the indicated sc-element is a member, with explicit indication of this membership.

The most commonly used types of knowledge within the proposed knowledge base model are the *semantic neighbourhood*, *factual knowledge*, *comparison*, *knowledge base section*, *subject domain*, *ontology*, *task*, *program*, *plan*, *solution*, *statement*, *definition*, *reasoning*, etc.

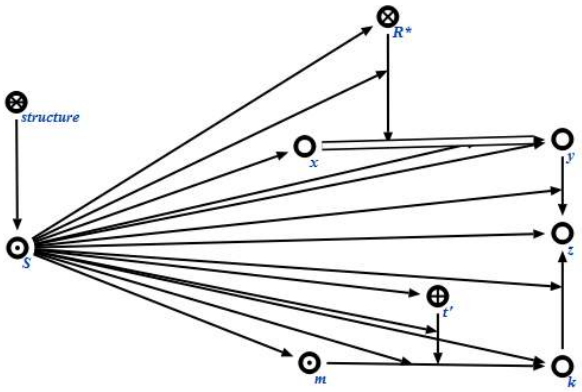
The formal refinement of the concepts of *structure* and *knowledge* makes it possible to provide the possibility of an unrestricted transition from knowledge to the corresponding *metaknowledge*. This property is achieved due to the ability to designate some fragment of the knowledge base with one sc-element, and, accordingly, consider it as a single whole, specify its properties and connections with other fragments. In turn, such a specification can be further considered as an object of formal description (Figure 4).

For the specification of individual entities within the knowledge base, the concept of a ***semantic neighbourhood*** is introduced.

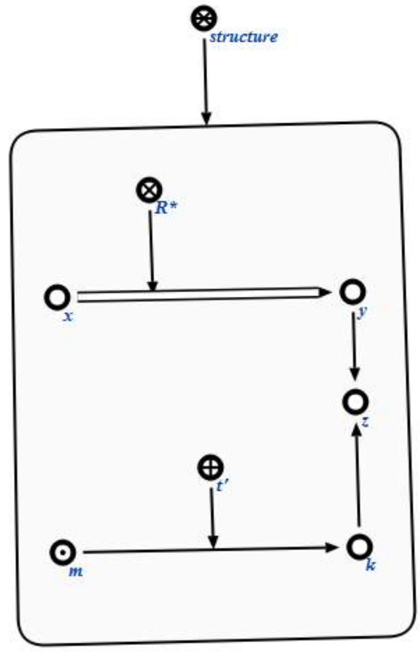
A ***semantic neighbourhood*** is a specification of some entity for a specific set of characteristics, which is essentially a collection of metainformation. Formally, the set of such attributes is determined by the set of relations and classes to which the described entity belongs and is the basis for semantic neighbourhoods classifying.

In general, a set of characteristics specifying entities belonging to certain classes will be different. In addition, it often becomes necessary to specify the same entity in various aspects and explicitly capture these aspects in the knowledge base.

For example, some person can be described from a professional (job, position, professional skills), medical (sex, weight, height, illness), civil (family status, nationality, age, attitude



a)



b)

Figure 3. An example of a complete and abbreviated representation of a structure in the SCg-code

to military service), etc. points of view (see Figure 5). The ability to describe the various properties of the same object is achieved through the allocation of various classes of semantic neighbourhoods and the selection of a set of characteristics that determine a particular class of semantic neighbourhoods.

An example of a semantic neighbourhood is shown in Figure 5.

The most important types of knowledge in terms of the process of knowledge bases development are ***subject domains*** and ***ontologies***. In the framework of the proposed approach, they are the basis for creating formal means of describing different types of knowledge in the knowledge base.

The concept of the *subject domain* is the most important in the field of knowledge engineering, allowing you to focus on a particular class of entities being researched and a specific family of relations that are specified on the specified class. Thus, abstraction is carried out from the rest of the investigated world.

Consideration of the structure of the knowledge base in relation to the *subject domain* allows us to consider the objects under study at different levels of detail. Detailed analysis of the researched objects can be carried out both within the original subject domains and within a system of independent, but related subject domains.

Each subject domain focuses on the description of the relations of the corresponding class of researched objects. Formally, the subject domain model is defined as follows:

$$M_{SD} = (SD_N, SD_E, SD_R, SD_O), \quad (1)$$

where SD_N – non-empty set of elements of the subject domain (carrier);
SD_E – set of roles of elements within the subject domain;
SD_R – set of relations between subject domains;
SD_O – set of classes of ontologies specifying subject domains.

An example of a formal specification of a subject domain is shown in Figure 6.

For a formal specification (description of properties) of the relevant subject domain, oriented to describing the properties and interrelations of the concepts that make up the specified subject domain, such type of knowledge is used as ***ontology***.

Within the framework of the proposed approach, the concept of *ontology* is refined and their typology based on the selected typology of *semantic neighbourhoods* is introduced. This approach to the allocation of ontology classes is based on the approach to the classification of ontologies, depending on the set of the relations used to describe the entity properties.

From the formal point of view, within the framework of the proposed knowledge base model, ***ontology*** will be interpreted as the result of the set-theoretical union of semantic neighbourhoods of one type. Depending on the properties of the subject domain concepts under consideration, which are described in the ontology, i.e. type of unioned semantic neighbourhoods, the following types of ontologies distinguished: *structural specification of the subject domain*, *set-theoretic ontology*, *logical ontology*, *terminological ontology*, *ontology of problems and problems solutions*, *ontology of classes of problems and methods for problems solving*, etc. In more detail this typology was considered in [22], [23].

Explicit allocation of *ontologies* in knowledge bases of

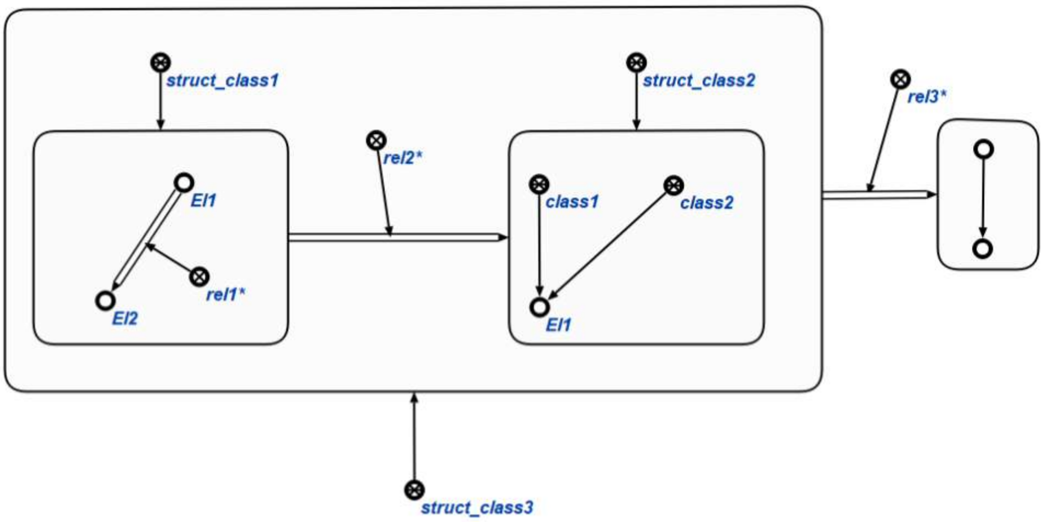


Figure 4. Example of metaknowledge description in SC-code

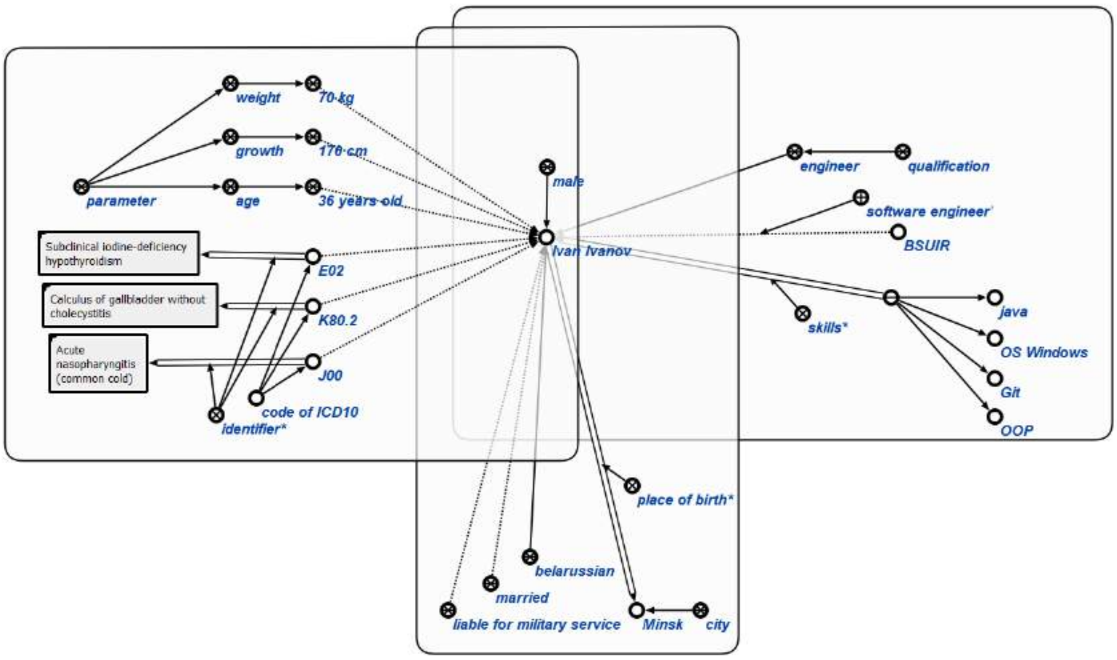


Figure 5. Example of the description of the semantic neighbourhood in the SC code

intelligent systems is necessary in order to:

- commit the agreed current version of the interpretation (clarification) of all the concepts used;
- ensure a clear organisation of the continuous process of development and harmonisation of the system of concepts used. This, in turn, requires a fairly detailed documentation (logging) of all changes in the system of concepts.

The main goal of structuring is to divide the specification of the described world into parts. This allows us to abstract each of the parts from those details that are not essential for solving the current problem, by analogy with experiments in various natural sciences.

One of the structuring way, which makes it possible to localise the domain of finding ways to solve problems, is the structuring based on the hierarchy of subject domains and their ontologies. In this case, the search area for solving a problem can be one or more subject domains that are sufficient to solve a given class of problems. At the same time, if necessary, the scope of the problem solution search can expand up to the

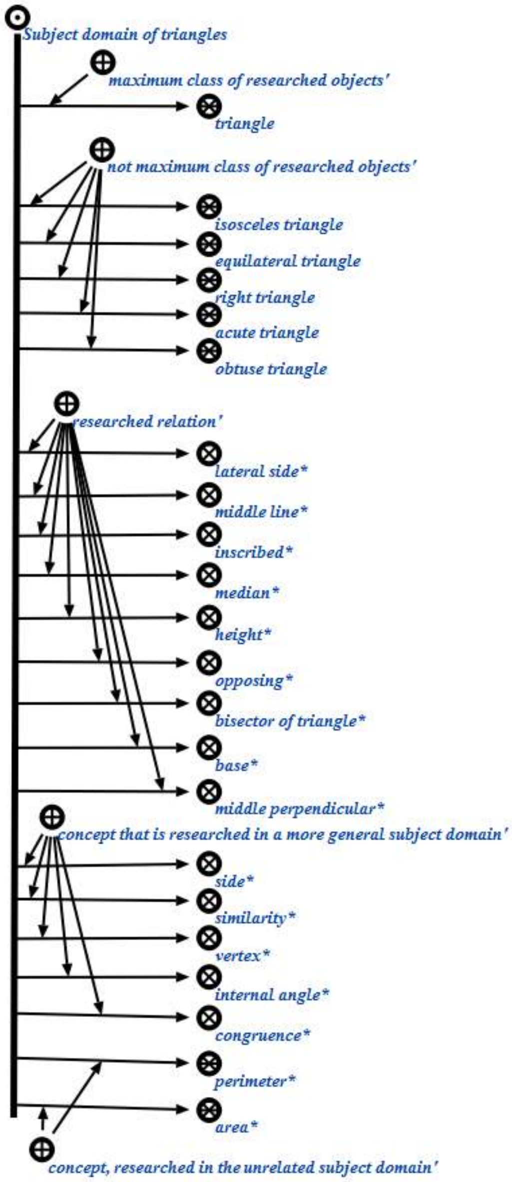


Figure 6. The example of a subject domain specification in the SC code

whole knowledge base.

Another important structuring way is the structuring of the knowledge base from the perspective of the development process (Figure 7), within which sections are identified that describe the part of the knowledge base available to the end user and parts of the knowledge base available to different categories of developers.

In addition, in the knowledge base it is necessary to describe the dynamics of the specified domain itself as a model of the external world fragment (i.e., the description of the past, present and future state of various external entities), and the dynamics of the knowledge base itself (i.e., the history of its development, plans for completion, current state), which is also one of the ways of the knowledge base structuring (Figure 7).

Using the models discussed earlier, a ***formal model of the knowledge bases structuring*** has been developed:

$$M_{STR} = (STR_S, STR_C, STR_R), \tag{2}$$

where STR_S – set of knowledge base sections;
STR_C – set of allocated classes of knowledge base sections, determined by a set of structuring characteristics;
STR_R – set of relations specifying the knowledge base sections, including - decomposition of sections into subsections.

This model is considered in more detail in [22]. An example of the structuring of the knowledge base on various aspects is shown in Figure 8.

The proposed model of structuring based on the allocation of sections and the formation of their hierarchy makes it possible to structure the knowledge base on the basis of an arbitrary set of characteristics, i.e. structuring the knowledge base from different points of view, while combining them within a single knowledge base.

An important feature of this approach is the description of the entire structure of the knowledge base with the help of SC-code in the same knowledge base, which in turn ensures the reflexivity of the intelligent system, that is, the ability to analyse the structure of its own knowledge base, for example, to identify various contradictions.

On the basis of the above results, a ***semantic model of the knowledge base*** is constructed that satisfies the above requirements:

$$M_{KB} = (M_S, ONT_R, ONT_{HL}, \{M_{STR1}, M_{STR2}, ...M_{STRn}\}), \tag{3}$$

M_S – set of structures stored in the knowledge base;
ONT_R – the ontology of sc-elements, which is the ontology of representation in the framework of the proposed approach;
$ONT_{HL} = \{ONT_{STR}, ONT_K, ONT_{SN}, ONT_{SD}, ONT_O\}$ – set of top-level ontologies,
ONT_{STR} – the ontology of the Subject domain of structures,
ONT_K – the ontology of the Subject domain of knowledge,
ONT_{SN} – the ontology of the Subject domain of semantic neighbourhoods
ONT_{SD} – ontology of the Subject domain of subject domains
ONT_O – ontology of the Subject domain of ontologies,
M_{STRi} – model of the knowledge base structuring with the i-th way.

In accordance with the proposed model, adding a new entity to the knowledge base requires specification of this entity using one or more concepts from the ontology of representation or the set of top-level ontologies. The specification of the entity in this case means the indication of the inclusion or belonging to any of the classes of the mentioned ontologies.

In turn, the addition of a new kind of knowledge is reduced to:

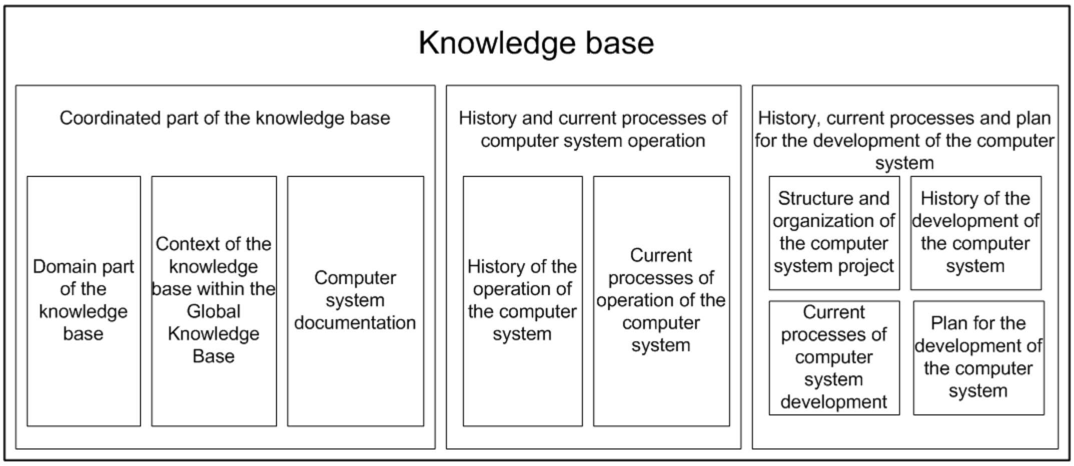


Figure 7. Structuring the knowledge base in terms of the development process

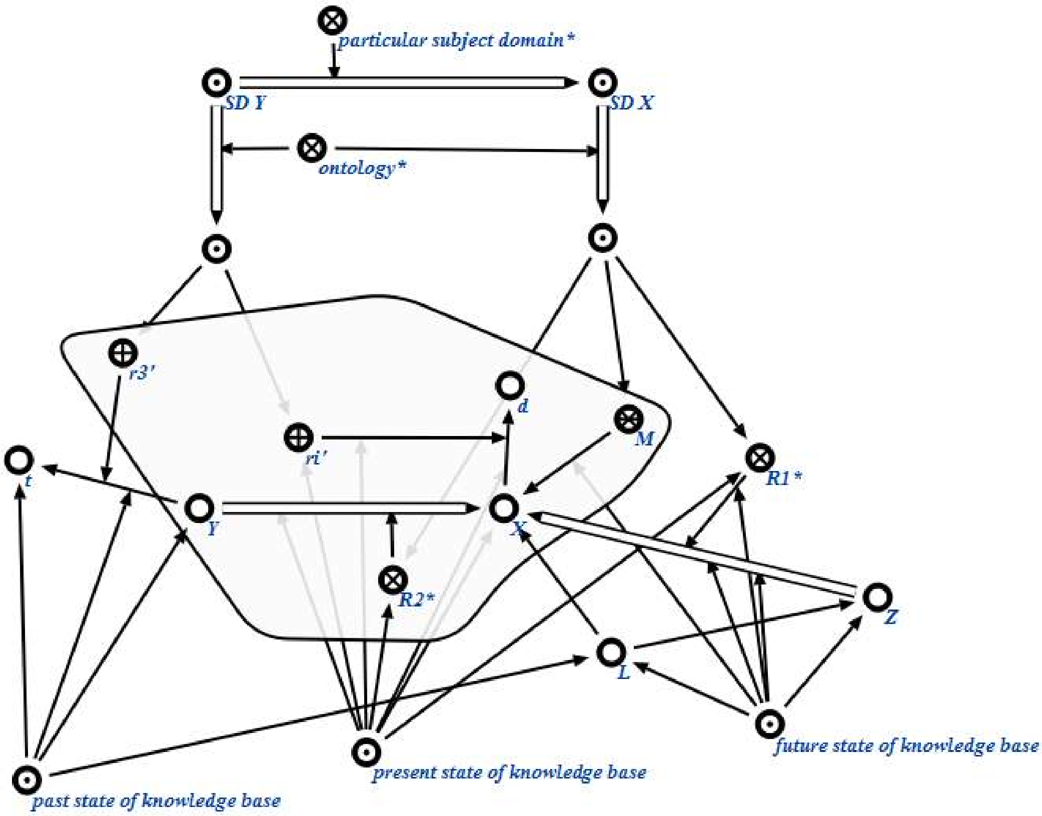


Figure 8. An example of different structuring ways combining in one knowledge base, represented in SC-code

- construction of a model of the relevant subject domain (or several subject domains), within the framework of which the introduced concepts are reseached - i.e. constructing of a structural specification of the subject domain, including indicating the relations of the considered subject domain with others already existing in the knowledge base model;
- building a family of ontologies of various types, specifying the developed subject domains.

The knowledge bases developed with the help of the proposed model have the following main advantages:

- the consistent use of different types of knowledge by using as a formal basis for information encoding of unified semantic networks with a basic set-theoretic interpretation, the syntax and semantics of which are specified within the *ontology of sc-elements*, which is the ontology of representation in the framework of the proposed approach, and use of the developed set of *top-level ontologies* (ontology of the subject domain of structures, ontology of the subject domain of sets, ontology of subject domain of connections and relations, ontology of subject domain of subject domains, etc.), specifying the most common kinds of knowledge, on the basis of which other types of knowledge may be described;
- the possibility of structuring from the point of view of various aspects and description in the knowledge base of meta-knowledge through the use of a *structure model* that allows to consider and specify a fragment of the knowledge base as a single whole, which in turn enables the transition from knowledge to meta-knowledge to an unlimited number of levels;
- convenience of the knowledge base processing due to the possibility of metainformation use during the processing, as well as the possibility of localising the area of finding ways to solve problems within one or several subject domains explicitly specified in the knowledge base;
- the flexibility of the knowledge base, i.e. its ease of modification, due to the explicit specification in the knowledge base of subject domains and their ontologies, which allows you to localise those fragments of the knowledge base in the process of the knowledge base developing and changing, which will also need to be changed.
- the ability to analyse, including, verify, adjust and optimise the structure of the knowledge base, history and plans for its development by the same means as the domain part of the knowledge base by using unified tools to describe all the listed information. At the same time, the possibilities of such an analysis can be easily extended by specifying in the knowledge base the information needed for analysis, for example, the specification of incorrect structures classes.

The principles of representation of various types of knowledge, using the ontology of sc-elements and a set of ontologies of the upper level, developed within the framework of the *knowledge base semantic model* allowed to solve a number of problems related to the presentation of knowledge within the Semantic Web, in particular, using the standards RDF and OWL 2. Let us consider several examples in more detail.

Example 1. In RDF and OWL 2, the principle of set normalisation is not used, which is one of the fundamental principles of the SC-code. The principle of the sets normalisation assumes that each sc-element (including the sc-arc) is treated either as a sign of the terminal entity (for example, the sign of a concrete number or the sign of a concrete material entity) or as a sign of a set whose elements are, in turn, only sc-elements. This principle ensures strict formalisation and unambiguous interpretation of each element of the knowledge base, which in turn increases the convenience of information processing and reduces the number of concepts.

Figure 9 shows an example of using the "color" concept in RDF (Figure 9a) and SCg (Figure 9b). As can be seen from Figure 9a, the specific color (Red) is denoted by the node of the semantic network and is associated with the object being characterised by the relation "have color". When trying to apply the principle of normalization, it becomes obvious that a particular color is not a terminal entity (there is no "red" entity), and should be treated as a set of all elements with this attribute, in this case, having a red color (Figure 9b). Thus, the relation "have color" is redunant. Analogous arguments lead to the fact that the relations "sex", "length" and other relations that describe the properties of objects in a certain parameter space turn out to be redundant. In turn, the use of the sc-arc of membership instead of the arc of the relation "have color" simplifies the processing of such a construction, since the semantics of such an arc is unambiguously interpreted by the processing means.

Example 2. The RDF and OWL2 tools allow you to describe relation properties, but do not allow you to specify individual connections of a specific relation, although sometimes such a need arises. Let us consider the example given on the official website of the W3C consortium [24], in which it is necessary to characterise the degree of certainty when exhibiting a certain diagnosis to the patient. As can be seen from Figure 10a, to record this information by means of RDF, it is necessary to introduce an additional node of the semantic network, denoting a specific diagnosis (in fact - a connection of the "diagnosis" relation) and relations "have diagnosis", "have value", "have probability". In addition, that such a description is cumbersome, the description of the diagnosis in the case where it is necessary to indicate the probability and in the case when it is not necessary to do so is fundamentally different (in the second case, only the "diagnosis" relationship between the patient and the value of the diagnosis is used). From the point of view of the set-theoretic interpretation underlying the SC-code, each relation connection (for example, the sc-arc) is treated as a set of related elements, which in turn is an element of the corresponding relation. This approach allows us to specify both the relation itself and separately each connection of the relation, while adding any specification does not lead to changes in the representation of the original relationship

Figure 9. An example of the "color" concept use

(Figure 10b). In this case, as in the previous example, the relation "have probability" turns out to be redundant.

Example 3. The set-theoretic approach, underlying the SC-code, allows, if necessary, to specify in the knowledge base not only binary connections, which are usually represented by sc-arcs or sc-edges, but also various capacity links, denoted by sc-nodes. However, such sc-nodes do not need to be named, unlike nodes in RDF-triples. Each such link can then be specified in the necessary way.

This approach solves one of the problems described on the W3C website [25], related to the presentation of the fact that some set is the union of a family of pairwise disjoint sets. In the variant of representation on OWL 2 (Figure 11a), the "union" relation is used for this, but this relation cannot be described by RDF triples, in addition, in this case it is necessary to indicate the fact of disjointness for each pair of sets. The SC-code means allow to introduce quasi-binary relations connecting some sc-element and some set (connection) of other sc-elements. For example, to solve the problem under consideration, we can define the relation "subdividing*" connecting some set and a family of pairwise disjoint sets (Figure 11b). A formal definition of this relation through other relations (for example, "union" and "pair of disjoint sets") can also be written in the SC-code, and not be specified separately for each use of this relation.

Next, we will consider some advantages of different types of knowledge representation using the models proposed in this work in relation to the classical models of representation (frames, products, logical models). Figure 12 shows an example of representing of the knowledge base fragment using frames (Figure 12a) and the first-order predicate logic language (Figure 12b) [26].

Figure 13 shows an example of a SC-code construction, semantically equivalent to the fragments shown in Figure 12.

Figure 14 shows an example of formalisation in SC-code of the following production rules:

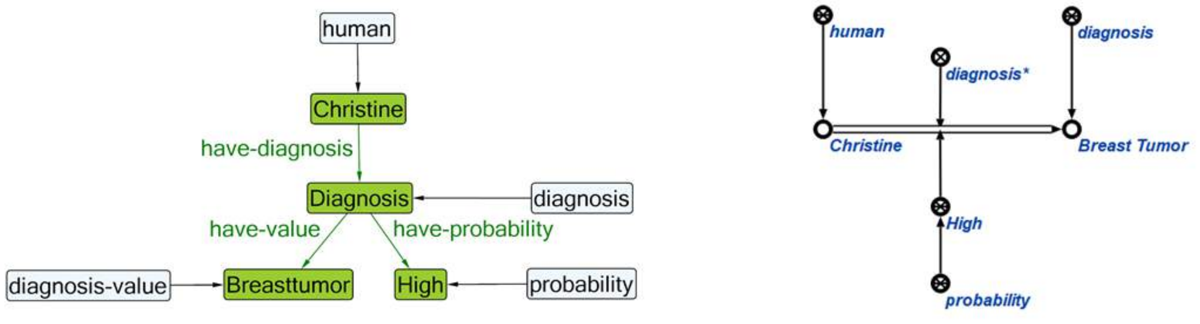
- IF *X* is a bird, then *X* is an animal;
- IF *X* is a bird, then *X* has two legs;
- IF *X* is an animal, then *X* has two legs OR *X* has four legs.

These examples allow us to formulate the following shortcomings of classical models:

- each classical model is heavily oriented to presenting knowledge of a particular types. So, with the help of frames it is inconvenient to represent logical rules and patterns; the product model is convenient for the presentation of rules, but requires the presence of some additional language for describing the factual information, in addition, the production model does not allow describing complex logical statements containing several quantifiers at different levels; the logical model is oriented to the representation of strict facts and statements, and in the classical version it does not allow to describe knowledge that does not have sufficient completeness, accuracy and correctness;
- the form of presentation of some information in the classical models is largely determined by the syntax of the chosen model, not by the sense of the information presented. So, for example, the presentation of even simple factographic information in the production model obliges the knowledge engineer to formulate all knowledge in the form of rules like «IF ..., THEN ...», although this is not always convenient for the developer and especially for the domain expert;
- the syntax of some classical models in some cases (for example, in the case of frames) is oriented on its perception by a person, not by a machine, in others (for example, in logical models) on the contrary. Thus, classical models do not allow to provide knowledge representation in the form of both human-friendly and convenient for storage and processing by a machine.
- in many cases, classical models (for example, frames) do not have sufficient strictness of representation and allow you to write the same information in different ways, which in the future can lead to incompatibility of fragments of knowledge bases developed by different developers.

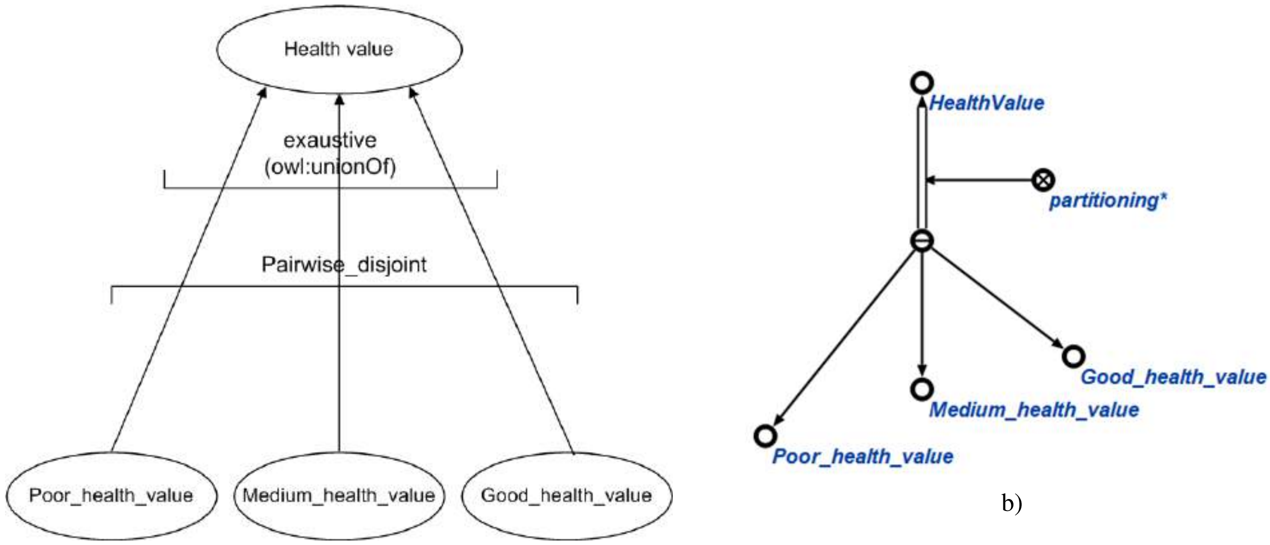
In turn, the models constructed on the basis of the SC-code, proposed in this paper, allow to eliminate the indicated shortcomings, in particular:

- the use of the apparatus of graph theory and set theory as a basis for representation of knowledge allows to ensure, on the one hand, strictness in presentation, on the other hand, presentation visibility and convenience



a) b)

Figure 10. Connection specification example



a) b)

Figure 11. An example of a description of disjoint subsets of a set

of information storing and processing in the computer system memory;

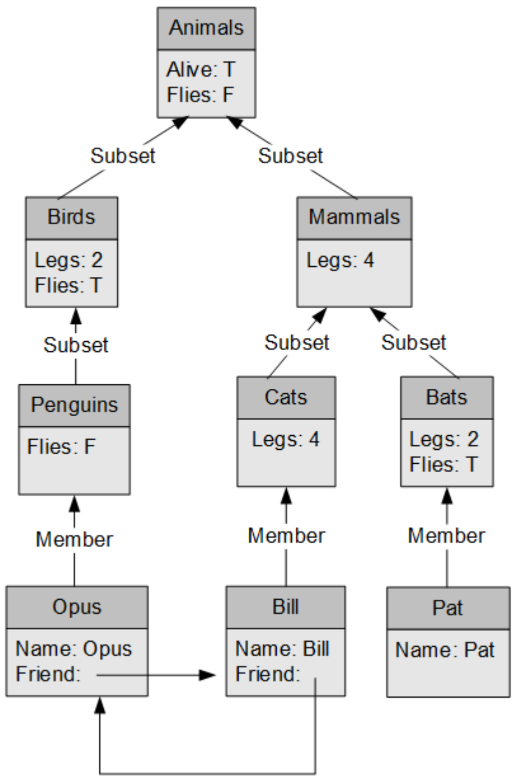
- the proposed models allow to represent different types of knowledge in a unified form (see Figures 13, 14);
- SC-code has a relatively small alphabet and simple syntax, all models built on its basis use the same basic alphabet and syntax, only a set of key nodes is expanded, thus the convenience of storage and processing is not violated. The meaning of the stored information is fixed by the configuration of the connections between sc-elements that are built in accordance with the mentioned syntax;
- The formalism used does not require the developer to bring the information presented to any special kind, determined by the syntactic features of the language. The only condition is the interpretation of all the described entities as particular for some of concepts from the representation ontology considered, based on the formalism of set theory, which in turn ensures the unambiguity of the semantics of each entity described in the knowledge base.

V. Method and tools for development of knowledge bases

Developing a knowledge base is a labour intensive and time-consuming process. Among the ways to reduce the timeframe for creating knowledge bases, the main ones are to ensure the joint development of knowledge bases by a distributed team of developers, automating their activities, and also reusing the already developed knowledge base components. [27], [28], [29], [30], [31], [32], [33], [34], [35], [36].

However, it is impossible to fully automate the process of developing the knowledge base, because some stages, such as the formation of a system of concepts, require the concerted efforts of a number of developers and experts and are subjective.

In addition, in the process of using any knowledge-based system, there is a permanent need to improve its knowledge base: the addition of new knowledge, the removal of irrelevant information, the search for and correction of errors and inaccuracies. The actualisation of information stored in



a)

Rel(Alive,Animal,T)
Rel(Flies,Animal,F)

Birds ⊂ Animals
Mammals ⊂ Animals

Rel(Flies,Birds,T)
Rel(Legs,Birds,2)
Rel(Legs,Mammals,4)

Penguins ⊂ Birds
Cats ⊂ Mammals
Bats ⊂ Mammals
Rel(Flies,Penguins,F)
Rel(Legs,Bats,2)
Rel(Flies,Bats,T)

Opus ∈ Penguins
Bill ∈ Cats
Pat ∈ Bats
Name(Opus,"Opus")
Name(Bill,"BIll")
Friend(Bill,Opus)
Friend(Opus,Bill)
Name(Pat,"Pat")

b)

Figure 12. An example of a knowledge base fragment presented with the help of frame and logic models

the knowledge base in accordance with the current state of the described subject domains (especially dynamic ones) may require such significant changes as the replacement of one conceptual system with another, including the introduction of new concepts, the removal of outdated concepts, the redefinition of existing concepts, etc. Due to the fact that standards, existing in various fields of application of knowledge-based systems, and also requirements and technologies are constantly changing, the process of evolution of the knowledge base must be carried out continuously directly during the operation of such a system [37]. The vast majority of knowledge base development tools do not provide such an opportunity, while strict dividing the processes of developing, improving and maintaining knowledge bases of knowledge-based systems.

A. Library of reusable components

To reduce labour costs in the development of knowledge bases of knowledge-based systems, it is proposed to use already developed fragments of knowledge bases or the whole knowledge bases of any systems. To organise the storage and search of such components, a ***library of knowledge bases reusable components***, which is part of the *IMS metasystem* [**?**], is proposed in this paper.

The *library of knowledge bases reusable components* includes a set of such components, means of such components specification and tools of automating the search for components based on their specifications (Figure 15).

Each ***knowledge bases reusable component*** is a structure either explicitly represented in the current state of the sc-memory or an incompletely formed structure which, if necessary, can be completely formed by combining its parts indicated by any decomposition relation, for example, a subdividing, or the inclusion relation, and which can be used within another knowledge-based system.

Each *knowledge bases reusable component* has a formal specification, that is, some semantic neighbourhood that characterizes this component. On the base of the formal specification, search for a suitable component in the library is carried out, comparation of it with other components, and so on.

The main semantic classes of knowledge bases reusable components stored in the library of knowledge base components are:

- semantic neighbourhoods of different entities;

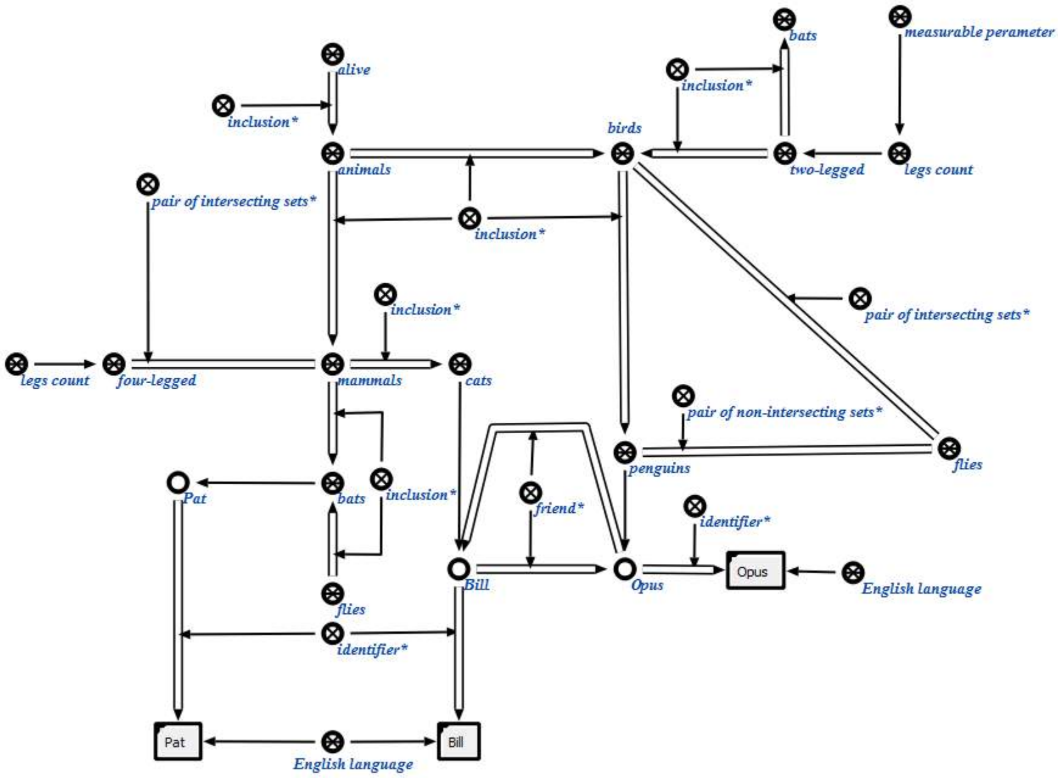


Figure 13. SC-code construction, semantically equivalent to the given examples

- ontologies of different subject domains;
- specifications of formal languages for describing different subject domains;
- sections of the knowledge base of various semantic types (including non-atomic ones);
- knowledge bases of entire subsystems that provide solutions to various tasks;
- and etc.

Integration of a *knowledge bases reusable component* into a system is reduced to merging key nodes by identifiers and eliminating possible duplications and contradictions that could arise if the developer of the system manually made any changes to its knowledge base.

The components search automation tools include tools for finding dependencies between components, searching for components in which the specified concepts are described, as well as searching for components by a fragment of their specification, etc.

More detailed knowledge bases component development based on *OSTIS Technology* is considered in [38].

B. The method of coordinated development of knowledge bases

When developing knowledge bases, it is necessary at every stage to ensure the semantic compatibility of knowledge bases and their components, that is, all concepts used must be treated equally in different components. Especially this task is relevant in terms of collective development, as well as in situations where the system of concepts used is changing. To ensure such compatibility it is necessary to use ontologies, as well as to commit protocols of coordinated changes in the knowledge base.

To solve these problems, a ***method of creating knowledge bases by a team of developers*** based on a formal model of the project activity of various knowledge base developers, each of which can play a certain role in the development process, is proposed. The construction of such a model and the clear allocation of classes of such actions allows to automate the process of collective development, as well as minimize the overhead costs for coordinating the activities of various developers. The main attention in the proposed method is given to the processes of harmonising the interpretation of certain concepts within the framework of the created knowledge base.

This method assumes two main stages - the *creation of a start-up version of the ostis-system* (including its knowledge base) and the stage of the knowledge base development itself.

The process of creating the start-up version of the ostis-system (system, built with OSTIS Technology) can be divided into four main stages:

- selection and installation of an interpretation platform for the ostis-system model;
- installation of the *Core of sc-models of knowledge bases* from the library of knowledge bases reusable components, which contains ontologies of the most common

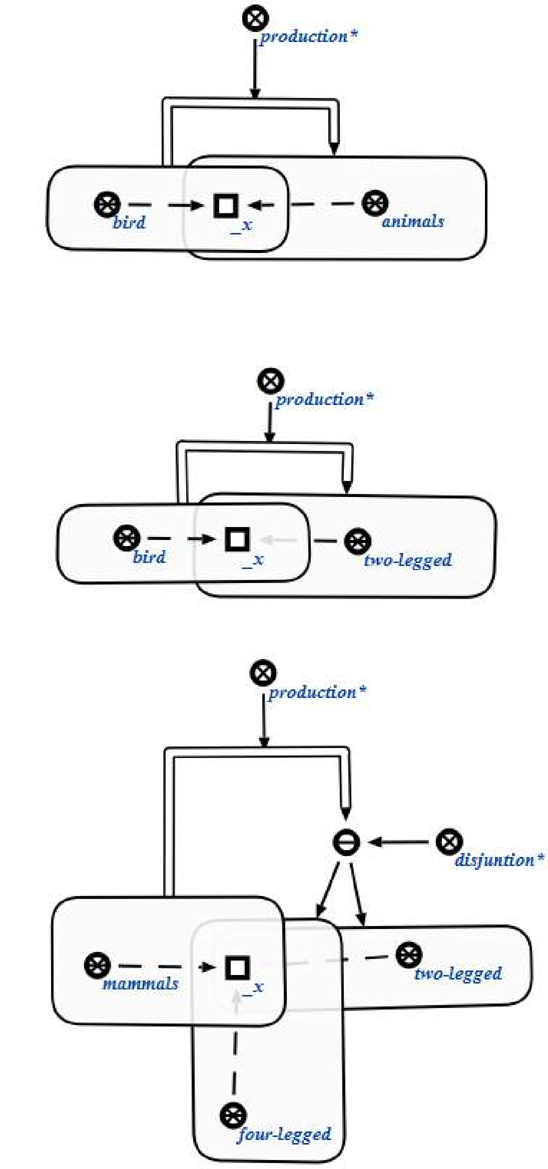


Figure 14. An example of formalisation of productions in SC-code

subject domains, i.e. top-level ontologies (for example, the *Subject domain of numbers and numeric structures*, the *Subject domain of ontologies*, the *Subject domain of logical formulas and logical ontologies*, the *Subject domain of connections and relations*, the *Subject domain of parameters and quantities*, etc.);

- installation of the *Core of the knowledge base processing machine* from the library of the reusable components of the knowledge base processing machines [39], that is, a set of basic reusable components of the knowledge base processing machines required to run the start-up version of the ostis-system;
- installation of the Core of the sc-models of interfaces [40], that is, a set of basic reusable components of the sc-models of interfaces required for running the start-up version of the ostis-system;
- installation of a system for the collective development of knowledge bases support.

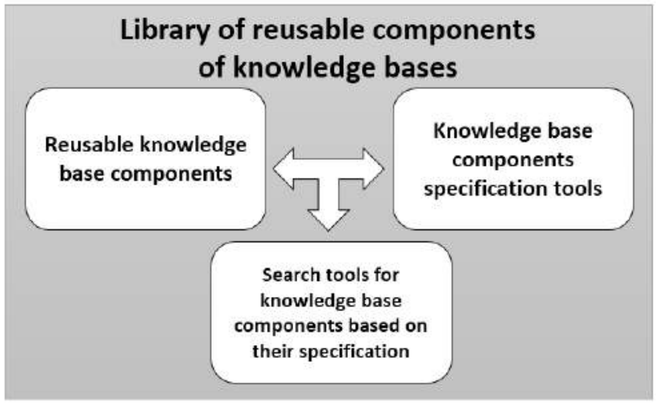


Figure 15. The structure of the library of reusable components of knowledge bases

After the basic configuration of the initial version of the ostis-system is build, the stage of development of the knowledge base begins, which includes the following stages:

1) formation of the initial structure of the knowledge base, which assumes:
   - formation of the structure of knowledge base sections;
   - identification of the described subject domains;
   - construction of a hierarchical system of described subject domains;
   - building a hierarchy of knowledge base sections within the subject domain of the knowledge base, taking into account the hierarchy of subject domains built at the previous stage.
2) identification the components of the knowledge base that can be taken from the library of reusable components of knowledge bases, and their inclusion into the developed knowledge base.
3) formation of project tasks for the development of missing fragments of the knowledge base and distribution of tasks among developers.
4) development and coordination of knowledge base fragments, which, in turn, can be included in the library of knowledge bases reusable components.
5) verification and debugging of the knowledge base.

It should be noted that during the development of the knowledge base, steps 3-5 are performed cyclically.

To ensure the property of the intelligent system's reflexivity, in particular, the possibility of the analysis automating of the history of the knowledge base evolution and generating plans for its development, all activities related to the development of the knowledge base must be specified in the same knowledge base by the same means as the domain part. To solve this problem, a ***formal ontology of the developers activity*** aimed at the development and modification of knowledge bases has been developed. This model is based on the *model of activity of various subjects* proposed in [41].

To organise the coordinated project activity on the creation of knowledge bases within the framework of this ontology, the roles of the participants in the development process (administrator, manager, expert, developer), the classes of actions performed by them, as well as the means for specifying the mentioned actions are introduced. This model is considered in more detail in [22].

The process of creating and editing the knowledge base of the ostis-system comes to the formation of *proposals for editing* a particular section of the knowledge base by developers and the subsequent consideration of these proposals by the knowledge base administrators. In addition, it is assumed that, if necessary, experts can be involved to verify the incoming proposals for editing the knowledge base, and the management of the development process is carried out by the managers of the relevant knowledge-base development projects (see Figure 16). At the same time, the formation of project tasks and their specification is also carried out with the help of the mechanism of proposals for editing the relevant section of the knowledge base. Thus, all information related to the current processes of developing the knowledge base, history and plans for its development is stored in the same knowledge base as its domain part, i.e. the part of the knowledge base available to the end user.

This approach provides wide possibilities for automating the process of knowledge bases development, as well as subsequent analysis and improvement of the knowledge base.

The developed method of coordinated construction and modification of knowledge bases on the basis of a formal ontology of project activity using reusable components allows to ensure the correctness and consistency of the project activity of developers directly in the process of using the knowledge base. At the same time, the discussion and the reconciliation process takes place in the same memory of the computer system where the knowledge base is also stored.

C. Knowledge Base Development Tools

To reduce the complexity of the process of knowledge bases development and reducing requirements for developers, *tools for automating the development of knowledge bases and information support for knowledge base developers* have been built.

The *tools of information support for developers* are implemented in the form of intelligent *metasystem IMS* (Intelligent MetaSystem) [**?**], which is also built using OSTIS technology. At each moment of time, the metasystem contains the models, tools and methods of developing computer systems based on OSTIS technology that have been accumulated and formalized to date. All the models, methods and tools presented in this dissertation work are formally described in the knowledge base of this system. In this way, it is possible to continuously update these results.

Tools of knowledge bases development automation are implemented in the form of a ***system for the knowledge bases collective development support***. An important aspect of supporting the creation of knowledge bases is to support the activity of knowledge base developers directly in the process of operating the system being developed. Thus, the system for supporting the collective development of knowledge bases is built in as a subsystem in each system being developed.

Based on the analysis of similar tools (Protégé [42], Co4 [43], NeON [44], etc.), the following additional requirements were formulated for the functionality of the *system for supporting the collective development of knowledge bases*, taking into account the needs of developers of knowledge bases and the identified shortcomings of the analogues considered:

- providing the possibility of both manual and automatic editing of knowledge bases;
- ensuring the possibility of automatic verification of the knowledge base, including the analysis of the correctness and completeness of the knowledge base;
- ensuring the possibility of creating a knowledge base by a distributed team of developers, including a mechanism for reconciling changes to the knowledge base, as well as a mechanism for storing the history of changes introduced with authority specification.

The implementation of these capabilities implies the refusal to work with the source code of the knowledge base. In this case it is assumed that all changes are made directly to the memory of the system, where the entire knowledge base is stored, which makes it possible to develop the knowledge base of the computer system in the process of its operation.

As it was said before, the system of support of collective development of knowledge bases is constructed as an ostis-system and has the appropriate architecture (figure 17)

The semantic model of the knowledge base of the system of support of collective development of knowledge bases includes sections containing all the knowledge necessary to support the development and evolution of the knowledge base:

- a set of top-level ontologies that are necessary for the functioning of the support system for the process of developing knowledge bases and are the basis for building the knowledge bases of the systems being developed;
- formal ontology of the subject domain of activities aimed at the development and evolution of knowledge bases, including a description of the typology of the knowledge bases developers roles, the classification of developers' actions, as well as formal means of specifying proposals for editing the knowledge base;
- ontology of the subject domain of problem structures in knowledge bases, that is, structures that describe incomplete, incorrect or excessive information in the knowledge base;
- tools of specifying changes and transient processes in the knowledge base.

Tools for processing the knowledge base of the system of support of collective development of knowledge bases is

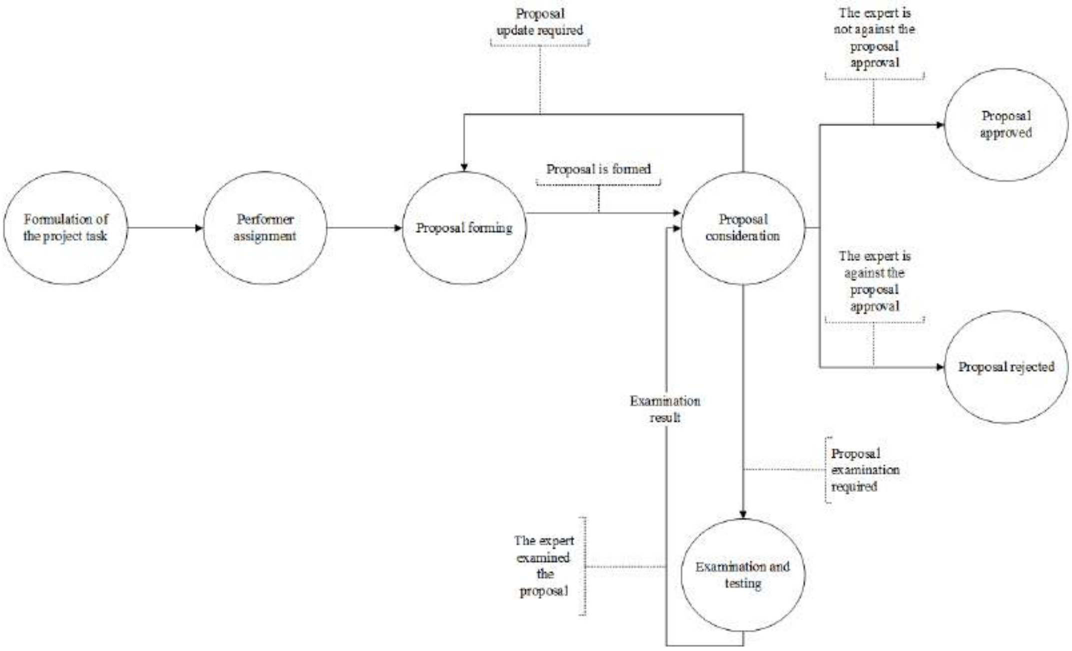


Figure 16. The mechanism for knowledge base fragments coordination

Figure 17. Architecture of the system of support of collective development of knowledge bases

a group of agents [45], each of which automates actions belonging to one of the knowledge bases developers actions classes discussed above. The tools for knowledge processing includes the following agents:

- class of agents for the verification of knowledge bases;
- class of knowledge base editing agents;
- class of automation agents for the activity of the knowledge base developer;
- class of automation agents for the activity of the knowledge base administrator;
- class of automation agents for the activity of the knowledge base manager;
- class of automation agents for the activity of the knowledge base expert;
- class of agents for calculating the characteristics of the knowledge base.

The *user interface of the system of support of collective development of knowledge bases* is represented by a set of interface commands that allow developers to initiate the activity of the required agent that is part of this system [40]. This set completely corresponds to the above set of the knowledge processing agents.

The developed tools for automating the process of building and modifying knowledge bases implement the proposed method and ensure coordination, verification and editing of knowledge base fragments directly in the process of their using.

VI. Conclusion

The proposed models, methods and tools were used to develop the knowledge bases of a number of systems, in particular, the IMS metasystem, as well as a number of applied intelligent reference systems in various subject domain, such as geometry, discrete mathematics, numerical models, chemistry, etc. , as well as in the development of a prototype of the batch production enterprises automation system [46].

Acknowledgement

This work was supported by BRFFR-RFFR (No F16R-101), BRFFR-SFFRU (No F16K-068).

References

[1] T. Gavrilova, D. Kudryavtsev, and D. Muromtsev, *Inzheneriya znanii. Modeli i metody: Uchebnik [Knowledge engineering. Models and methods: Textbook]*. SPb.: Lan', 2016.
[2] G. Alor-Hernández and R. Valencia-García, Eds., *Current Trends on Knowledge-Based Systems*. Springer, 2017.
[3] A. Kolesnikov, I. Kirikov, and S. Listopad, *Gibridnye intellektual'nye sistemy s samoorganizatsiei: koordinatsiya, soglasovannost', spor [Hybrid intelligent systems with self-organization: coordination, consistency, dispute]*. M.: IPI RAN, 2014.
[4] L. Kalinichenko, *Metody i sredstva integratsii neodnorodnykh baz dannykh [Methods and tools for integrating heterogeneous databases]*. M.: Nauka, 1983.

[5] Y. Val'kman, V. Gritsenko, and A. Rykhal'skii, *Model'no-parametricheskoe prostranstvo. Teoriya i prilozhenie [Model-parametric space. Theory and application]*. Kiev: Naukova dumka, 2010.

[6] Y. Zagorul'ko, "O kontseptsii integrirovannoi modeli predstavleniya znanii [on the concept of an integrated knowledge representation model]," *Izvestiya TPU*, vol. 322, no. 5, pp. 98–103, 2013.

[7] A. Kleshchev, *Semanticheskie porozhdayushchie modeli. Obshchaya tochka zreniya na freimy i produktsii v ekspertnykh sistemakh [Semantic generative models. General point of view on frames and products in expert systems]*, Vladivostok, 1986.

[8] V. Rubashkin, "Ontologii: ot informatsionno-poiskovykh tezaurusov k inzhenerii znanii [ontology: from information-search thesauri to knowledge engineering]," in *Desyataya natsional'naya konferentsiya po iskusstvennomu intellektu s mezhdunarodnym uchastiem KII-2006 (Obninsk, 25-28 sentyabrya 2006 g.)*. Moskva: Fizmatlit, 2006.

[9] G. Shchedrovitskii, *Sintez znanii: problemy i metody [Synthesis of knowledge: problems and methods]*. M.: Nauka, 1984, pp. 67–99.

[10] T. Gavrilova and V. Khoroshevskii, *Bazy znanii intellektual'nykh sistem: Uchebnik [Knowledge bases of intellectual systems: Textboo]*. SPb.: Piter, 2001.

[11] T. H. R. Chan, "Design an integrated knowledge representation (ikr) model," Degree of Master of Philosophy, Department of Manufacturing Engineering and Engineering Management, May 2006.

[12] V. Ivashenko, "Modeli i algoritmy integratsii znanii na osnove odnorodnykh semanticheskikh setei [models and algorithms of knowledge integration based on homogeneous semantic networks]," avtoref. dis… kand. tekhn. nauk: 05.13.17, V.P. Ivashenko ; Uchrezhdenie obrazovaniya «Belorusskii gosudarstvennyi universitet informatiki i radioelektroniki», Minsk, 2014.

[13] (2016) The syntax of cycl. [Online]. Available: http://www.cyc.com/documentation/ontologists-handbook/cyc-basics/syntax-cycl

[14] P. Benjamin, C. Menzel, R. Mayer, F. Fillion, M. Futrell, P. S. deWitte, and M. Lingineni, *IDEF5 Method Report*. College Station : Knowledge Based Systems, Inc., 1994.

[15] *ISO/IEC 13211-1:1995 Information technology – Programming languages – Prolog – Part 1: General core [Electronic resource]*, International Organization for Standardization Std., mode of access: https://www.iso.org/obp/ui/#iso:std:iso-iec:13211:-1:ed-1:v1:en. — Date of access: 08.08.2016.

[16] (2017) Clips - a tool for building expert systems. [Online]. Available: http://clipsrules.sourceforge.net/

[17] (2017, Dec.) The world wide web consortium. [Online]. Available: https://www.w3.org

[18] V. Khoroshevskii, "Prostranstva znanii v seti internet i semantic web (chast' 1) [knowledge spaces on the internet and semantic web (part 1)]," *Iskusstvennyi intellekt i prinyatie reshenii*, no. 1, pp. 80–97, 2008.

[19] "Ontology Tools [Electronic resource]," *Open Semantic Framework*, mode of access: http://wiki.opensemanticframework.org/index.php/ Ontology_Tools. - Date of access: 21.09.2016.

[20] S. Gorshkov, *Vvedenie v ontologicheskoe modelirovanie [Introduction to ontological modeling]*. Ekaterinburg: OOO «TriniData», 2016.

[21] V. Golenkov, "Grafodinamicheskie modeli i metody parallel'noi asinkhronnoi pererabotki informatsii v intellektual'nykh sistemakh [graphodynamic models and methods of parallel asynchronous information processing in intelligent systems]," Dis… d-ra tekhn. nauk: 05.13.11, 05.13.17, Minsk, 1996.

[22] I. Davydenko, N. Grakova, E. Sergienko, and A. Fedotova, "Sredstva strukturizatsii semanticheskikh modelei baz znanii [means of structuring semantic models of knowledge bases]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BGUIR. Minsk : BGUIR, 2016, pp. 93–106.

[23] I. Davydenko, "Ontology-based knowledge base design," in *Open semantic technologies for intelligent systems*, V. Golenkov, Ed., BSUIR. Minsk : BSUIR, 2017, pp. 57–72.

[24] "Defining n-ary relations on the semantic web [Electronic resource]," *The World Wide Web Consortium (W3C)*, mode of access: https://www.w3.org/TR/swbp-n-aryRelations/#pattern1. — Date of access: 30.09.2016.

[25] (2017) Representing specified values in owl: "value partitions" and "value sets" in the world wide web consortium (w3c). [Online]. Available: https://www.w3.org/TR/2005/NOTE-swbp-specified-values-20050517/

[26] S. Rassel and P. Norvig, *Artificial Intelligence. The modern approach*. Moskva : Vil'yams, 2006.

[27] I. Efimenko and V. Khoroshevskii, *Ontologicheskoe modelirovanie ekonomiki predpriyatii i otraslei sovremennoi Rossii: Chast' 1. Ontologicheskoe modelirovanie: podkhody, modeli, metody, sredstva, resheniya [Ontologicheskoe modelirovanie ekonomiki predpriyatii i otraslei sovremennoi Rossii: Chast '1. Ontologicheskoe modelirovanie: podkhody, modeli, metody, sredstva, resheniya]*, F. Aleskerov, V. Podinovskii, and B. Mirkin, Eds. M.: Izd. dom Vysshei shkoly ekonomiki, 2011, (Preprint / WP7/2011/08 (ch. 1)).

[28] V. Gribova, A. Kleshchev, F. Moskalenko, V. Timchenko, L. Fedorishchev, and E. Shalfeeva, "Platforma dlya razrabotki oblachnykh intellektual'nykh servisov [platform for developing cloud-based intelligent services]," in *XV natsional'naya konferentsiya po iskusstvennomu intellektu s mezhdunarodnym uchastiem KII-2016 (3–7 oktyabrya 2016 g., g. Smolensk, Rossiya). Trudy konferentsii. V 3-kh tomakh*. Smolensk: Universum, 2016, pp. 24–33.

[29] E. J. Reddy, C. Sridhar, and V. P. Rangadu, "Knowledge based engineering: Notion, approaches and future trends," *American Journal of Intelligent Systems*, no. 5 (1), pp. 1–17, 2015.

[30] H. Eriksson, Y. Shahar, S. Tu, A. Puerta, and M. Musen, "Task modeling with reusable problem-solving methods," *Artificial Intelligence*, no. 79 (2), pp. 293–326, 1995.

[31] R. Mizoguchi, J. Vanwelkenhuysen, and M. Ikeda, *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing*. Amsterdam : IOS Press, 1995, ch. Task ontology for reuse of problem solving knowledge, pp. 46–59.

[32] A. N. Borisov, "Postroenie intellektual'nykh sistem, osnovannykh na znaniyakh, s povtornym ispol'zovaniem komponentov [building knowledge-based intelligent systems, reusing components]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, V. G. (otv. red.) [i dr.], Ed. Minsk: BGUIR, 2014, pp. 97–102.

[33] B. Wielinga and A. Schreiber, *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing*. Amsterdam : IOS Press, 1994, ch. Reusable and sharable knowledge bases: a European perspective, pp. 110–120.

[34] R. Hartung and A. Håkansson, "Using meta-agents to reason with multiple ontologies," *Agent and Multi-Agent Systems: Technologies and Applications. KES-AMSTA 2008. Lecture Notes in Computer Science*, vol. 4953, pp. 261–270, 2008.

[35] A. Gladun and Y. Rogushina, "Repozitorii ontologii kak sredstvo povtornogo ispol'zovaniya znanii dlya raspoznavaniya informatsionnykh ob"ektov [ontology repositories as a means of re-using knowledge to recognize information about objects]," *Ontologiya proektirovaniya*, no. 1, pp. 35–50, 2013.

[36] C. Debruyne, P. D. Leenheer, and R. Meersman, *On the Move to Meaningful Internet Systems: OTM 2009 Workshops*. Heidelberg : Springer, 2009, ch. Method and Tool for Fact Type Reuse in the DOGMA Ontology Framework, pp. 1147–1164.

[37] V. Lapshin, *Ontologii v komp'yuternykh sistemakh [Ontologies in computer systems]*. M.: Nauchnyi mir, 2010.

[38] I. Davydenko, "Tekhnologiya komponentnogo proektirovaniya baz znanii na osnove unifitsirovannykh semanticheskikh setei [the technology of component design of knowledge bases on the basis of unified semantic networks]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BGUIR. Minsk : BGUIR, 2013, pp. 185–190.

[39] D. Shunkevich, I. Davydenko, D. Koronchik, I. Zhukov, and A. Parkalov, "Sredstva podderzhki komponentnogo proektirovaniya sistem, upravlyaemykh znaniyami [tools for supporting the component design of knowledge-driven systems]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BGUIR, 2015, pp. 79–88.

[40] D. Koronchik, "Semanticheskie modeli mul'timodal'nykh pol'zovatel'skikh interfeisov i semanticheskaya tekhnologiya ikh proektirovaniya [semantic models of multimodal user interfaces and semantic technology of their design]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. Minsk: BGUIR, 2012, pp. 339–346.

[41] D. Shunkevich, A. Gubarevich, M. Svyatkina, and O. Morosin, "Formal'noe semanticheskoe opisanie tselenapravlennoi deyatel'nosti razlichnogo vida sub"ektov [a formal semantic description of the purposeful activity of various types of sub-objects]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. Minsk: BGUIR, 2016, pp. 125–136.

[42] (2016) Protege. ontology editor and framework for building intelligent systems. [Online]. Available: http://protege.stanford.edu

[43] A. Gomez-Perez and M. Suarez-Figueroa, "Scenarios for building ontology networks within the neon methodology," in *Proceedings of the Fifth International Conference on Knowledge Capture (K-CAP 2009)*. New York : ACM, 2009, pp. 183–184.

[44] J. Euzenat, *10th Knowledge Acquisition, Modeling and Management for Knowledge-based Systems Workshop (KAW'96)*. Banff : Alberta, 1996, ch. Corporate memory through cooperative creation of knowledge bases and hyper-documents, pp. 1–18.

[45] D. Shunkevich, "Ontologicheskoe proektirovanie mashin obrabotki znanii [ontological design of knowledge processing machines]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. Minsk: BGUIR, 2017, pp. 73–94.

[46] I. Davydenko, V. Golenkov, V. Taberko, D. Ivanyuk, K. Rusetskii, D. Shunkevich, V. Zakharov, V. Ivashenko, and D. Koronchik, "Proektirovanie predpriyatii retsepturnogo proizvodstva na osnove ontologii [designing enterprises of prescription production on the basis of ontology]," *Ontologiya proektirovaniya*, vol. 7, no. 2(24), pp. 123–144, 2017.

СЕМАНТИЧЕСКИЕ МОДЕЛИ, МЕТОД И СРЕДСТВА СОГЛАСОВАННОЙ РАЗРАБОТКИ БАЗ ЗНАНИЙ НА ОСНОВЕ МНОГОКРАТНО ИСПОЛЬЗУЕМЫХ КОМПОНЕНТОВ

И.Т. Давыденко
Белорусский государственный университет
информатики и радиоэлектроники
г. Минск, Беларусь

База знаний является ключевым компонентом такого класса компьютерных систем, как системы, основанные на знаниях, разработка которых является одним из перспективных направлений в области искусственного интеллекта. Качество разрабатываемых систем такого класса определяется, в том числе, качеством базы знаний и разнообразием видов знаний, хранимых в ней.

Расширение областей применения систем, основанных на знаниях привело к необходимости поддержки решения комплексных задач. Под комплексной задачей будем понимать задачу, решение которой предполагает применение формализованных знаний различного вида и различных моделей их обработки, что, в свою очередь, требует обеспечения совместимости и интеграции используемых знаний, а также моделей их обработки.

Настоящая работа посвящена решению задач, связанных с разработкой моделей методов и средств создания баз знаний компьютерных систем, способных решать комплексные задачи.