From training intelligent systems to training their development tools

Golenkov V. V., Gulyakina N. A.,
Grakova N. V., Davydenko I. T., Nikulenka V. Y.
The department of Intelligent Information Technology
Belarusian state University of Informatics and Radioelectronics
Minsk, Belarus
{golen, guliakina, grakova}@bsuir.by
ir.davydenko@gmail.com, v.nikulenka@jetsys.io

Eremeev A. P.
The department of applied mathematics
National research University "MEI"
Moscow, Russia
eremeev@appmat.ru

Tarasov V. B.
The department of computer systems of production automation
Moscow state technical University N. E. Bauman
Moscow, Russia
vbulbov@yahoo.com

***Abstract*—The article considers the approach to the transition implementation from the training of intellectual systems based on knowledge to the training of tools for their development. At the same time, the architecture of such an intelligent system is considered as the basis for ensuring its flexibility and learnability. In addition, the article examines the areas of learning and self-learning of intelligent systems, as well as their ability to acquire knowledge and skills from various sources. The justification of OSTIS technology using for the development of knowledge-based intelligent systems is provided.**

***Keywords*—learnability of intellegent systems; skill; intellegent system; problem; external environment; self-training; reflexivity; unlimited learnability; flexibility; machine learning; training of intellegent systems; artificial neural network; genetic algorithm; technology of development of intellegent systems; ostis-system; self-training of intellegent systems; metaknowledge**

I. Introduction

Here are the main points of this article:

1) ***Learnability of intelligent system***, that is, its ability to acquire new ones and to improve already acquired ***knowledge*** and ***skills*** is the main characteristic of intelligent systems. *Learnability of intelligent system* creates the necessary conditions to ensure the rapid pace of ***intelligent systems*** evolution, to intensively expand the set of ***problems*** which it solves, to continuously improve the quality of the *problems* solution, to quickly adapt *intelligent systems* to changes in ***the external environment*** and operating conditions.
2) *The level of learnability of intelligent system* is determined by the level of development of the ***self-learning*** means and, first of all, by the level of its ***reflexivity***, that is, the ability for self-analysis.
3) The highest form of *learnability of intelligent systems* is their ***unlimited (universal) learnability***, assuming the ability to acquire and improve any (!) types of knowledge and skills.
4) *Learnability of intelligent system* is determined by the degree of its ***flexibility***, that is, the complexity of making various changes to the *intelligent system*, as well as the variety of types of possible changes.
5) Modern work in the field of ***machine learning*** does not consider the whole complex of problems associated with the ***training of intelligent systems***, but focuses mainly on improving various *skills*. It also applies to the training of ***artificial neural networks*** and to the training of ***genetic algorithms***.
6) The most important criterion for the quality of the proposed ***technologies for development of intelligent systems*** is *the level of learnability* of developed *intelligent systems* which it provides.
7) *Intelligent systems* developed using ***OSTIS Technology*** [1], which we call ***ostis-systems***, have a high level of *flexibility* and *unlimited learnability*.
8) ***Flexibility of ostis-systems*** is determined by:
   - the sense nature of the ***internal knowledge representation***;
   - the universality of the language of the internal ***knowledge representation***;
   - the developed level of ***the associative organization of memory*** of *ostis-systems*;
   - ***agent-oriented management of knowledge processing***, managed by ***knowledge base***.
9) ***Learnability of ostis-systems*** is determined by the ability of *ostis-system* to detect ***contradictions*** (errors), ***information holes*** and ***information waste*** that appear in the current state of *knowledge base* both as a result of acquiring *knowledge* and *skills* from outside and in the

process of solving various *problems*. In the second case, each detected *contradiction*, *information hole* or *information waste* is clearly associated with the ***information process*** that generated it to clarify the reason for its occurrence.

10) ***Unlimited learnability of ostis-systems*** is determined by:
    - universality of ***internal sense language of knowledge representation***;
    - unlimited opportunities for transition in ***internal representation of knowledge bases*** from *knowledge* to ***metaknowledge*** and, as a result, unlimited possibilities for *structuring knowledge bases*;
    - universal character of *agent-oriented model of knowledge processing*.
11) *OSTIS Technology* also has high level of *flexibility* and *unlimited learnability*, because it is implemented in the form of ***intelligent metasystem***, which is also *ostis-system.*

II. The features of intelligent systems

Transition of ***traditional computer systems*** to *intelligent systems*, that is ***to the systems based on knowledge*** is the most important tendency of transition to the next generation of the computer systems having ***semantic compatibility***, *learnability* and ability to formation of temporary collectives of computer systems in need of the collective solution of difficult tasks. At the same time it is essential to emphasize that any modern computer system can be realized in the form of functionally equivalent *systems based on knowledge*. Such transformation of modern computer systems will allow to increase their quality and competitiveness significantly. As a result of transformation of modern computer systems in functionally equivalent *systems based on knowledge* we will receive ***semantic compatible systems based on knowledge*** with *the high level of learnability*, differing among themselves in structural complexity of *knowledge bases* and functional complexity of means of processing of knowledge.

It is necessary to distinguish *traditional computer systems* using some methods of artificial intelligence (for example, artificial neural networks, genetic algorithms, nonclassical logics) from *intelligent systems*, which consists of (1) the base of all necessary *knowledge* and *skills* (2) ***interface with external environment*** and (3) ***integrated problem solver*** which consists of *skill* base acquired by intelligent system (methods for solving various classes of problems) and ***interpreter of knowledge and skills*** acquired by *intelligent system*. General architecture of *intelligent system* is shown in (Figure 1), which is explained below in formal language ***SCn-code*** [1] (Listing 1).

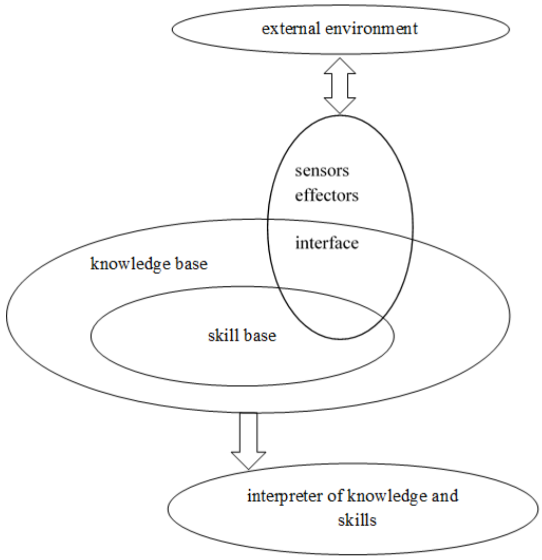

Figure 1. The architecture of intelligent system

intellectual system
= *trained system based on knowledge*

knowledge base
= *knowledge base and skills of intelligent system*
= *systematized mix of all knowledge and skills stored in memory of intelligent system*
<= *second domain**
   *knowledge base**
= *Class of information structures, each of which is knowledge base of some intelligent system*

knowledge base*
= *Binary oriented relation connecting each intelligent system with its knowledge base*

skill base
= *skill base of intelligent system*
<= *generalized inclusion**
   knowledge base
= *a systematized complex of all methods of solving various problems and classes of problems known to intelligent system*
= *part of knowledge base, which is systematized mix of all skills that intelligent system has acquired to the present moment, each of which is knowledge about how to solve specific problem or some class of problems*

memory of intelligent system
= *internal environment of intelligent system in which its knowledge base is stored and processed and which is contrasted to external environment of this intelligent system*

external environment of intelligent system
= *mix of all kinds of entities which are not signs or sign constructions stored in memory of intelligent system and*

which are in relation to the indicated internal signs, can only implement the role of their denotations (that is, the role of described entities)

interface of intelligent system
= *interface of intelligent system with its external environment*

sensors of intelligent system
= *sensory means of intelligent system that ensure reflection of events and situations occurring in external environment of intelligent system (including reflection of intelligent system behavior in its external environment), events and situations occurring in memory of this intelligent system*

effectors of intelligent system
= *effector means of intelligent system, providing transformation of planned actions specifications of intelligent system in external environment*

interpreter of knowledge and skills
= *interpreter of knowledge base and skill base of intelligent system*
= *base machine of intelligent system*
= *processor of intelligent system*
= *engine of intelligent system*
= *set of means to solve problems in intelligent systems*
= *set of means by means of which the intelligent system carries out the solution of tasks*

integrated problem solver
= *integrated problem solver of intelligent system*
<= *generalized decomposition**
{
• *skill base*
• *interpreter of knowledge and skills*
}

Listing 1. Explanation of general architecture of intelligent system in formal language SCn-code

The architecture of *intelligent system* is the basis for ensuring its *flexibility* and *learnability* due to its clear stratification to independent components, expansion and improvement of each of them sufficiently is not dependent on changes in other components.

Traditional computer systems are mainly focused on solving problems, the initial data of which are localized (fully specified). To solve the problems there is no need for additional information. The specialized solvers can exist for these problems which can use convenient specialized methods of representing and storing the processing data. In order to integrate these specialized solvers into the developing computer system it is sufficient to organize the exchange of a finite number of initial (transmitted) and returned parameters with a priori known semantic interpretation or the exchange of finite information constructions presented in a specified format.

But for *problems* solved by *intelligent systems*, in general, it is not known what data and knowledge should be used to solve these problems, what methods should be used to solve them. When you solve these problems, it is necessary to localize a fragment of the *knowledge base* which comprises the data and the *knowledge* which are sufficient for the solution of the *problem* and excluding those data and *knowledge* that are obviously not needed for this, as well as the distinguishing of the available solutions from the variety of methods that are sufficient for solving this *problem*.

The competitiveness of *intelligent system* ensures:

- permanent *training of intelligent system* during the operation;
- permanent ***self-learning of intelligent system*** during the operation;
- *learnability of intelligent system*;
- ***flexibility of intelligent system***.

The circumstances that constrain the development of *intelligent systems* market:

- inter-subject character of the development of *intelligent systems*;
- high science-intensive development of *intelligent systems*;
- the need for close cooperation with experts of applied fields;
- lack of ***general theory of intelligent systems***;
- lack of ***common technology for the development of intelligent systems*** which is available to a wide range of engineers;
- high and not always adequate social interest in artificial intelligence;
- substantially more complicated training of engineers of intelligent systems in comparison with the training of traditional computer systems engineers.

The main features of the next generation intelligent systems:

- the variety of problems types solved by the developed intelligent systems:
  - •• solving problems that require the joint coordinated use of problem-solving models variety, types of knowledge and knowledge representation models;
  - •• solving problems which a priori aren't known how to solve and the necessary knowledge and also models (principles) for solving problems aren't known;
- low deadlines and relatively low labor content for the development of the initial versions of *intelligent systems*, which are *trained* and do *self-learning* during the operation;
- high rates, low labor content and unlimited possibilities for the evolution (improvement) of *intelligent systems* during the operation with the maintaining the compatibility of these systems;
- high rates of evolution, low labor content and unlimited possibilities for the evolution of the *technology for the development of intelligent systems* with the automation of making appropriate changes to the operating *intelligent systems* while ensuring the compatibility of these systems.

III. Training of intelligent systems

Under ***the training of intelligent systems*** we don't mean only the training, for example, of *artificial neural networks* that are part of *intelligent systems*. We interpret *the intelligent system training* as a process of new knowledge and skills acquiring and improving for this system, implemented by some team of ***teacher-developers*** of the specified system.

Training of intelligent system has the following stages:

- preliminary debugging (testing and tuning) of *intelligent system* to solve some problems that is implemented before the introduction of *intelligent system* in operation. It is clear that the stage of creating (synthesizing, assembling) the initial (starting) version of this system should precede the debugging of *intelligent system*;
- *training of intelligent system* is implemented directly during its operation on the basis of a permanent analysis of the trained *intelligent system* activity.

Training of intelligent system is implemented:

- for improvement of its *knowledge base*;
- for improvement of its *integrated problem solver*;
- for improvement of its verbal and, primarily, ***user interface*** with various external subjects;
- for improvement of its sensory-effector subsystems, providing interaction with the *external environment*.

IV. Self-learning of intelligent system

Self-learning of intelligent system is a transition from a passive form of *training intelligent system*, when the trained system is treated as a passive "container" filled by *teachers-developers* with new *knowledge* and *skills*, to an active form of training, when the trained system becomes the participant of the training process. *Self-learning* of *intelligent system* is the automation of various processes aimed at training of *intelligent system* and implemented by *intelligent system.* These automated processes include:

1) permanent ***analysis of knowledge base quality*** and *integrated problem solver* of trained *intelligent system*, which has led to, for example, the identification and specification of various kinds of errors (contradictions), *information waste*, *information holes* (lack of *knowledge* and *skills*), assessment of the reliability of new acquired (including input) *knowledge* and *skills*, as well as the specification of the *knowledge* and the *skills* (who is the author, the moment of appearance in the system, the type and other);
2) Some automated types of improving the current state of *knowledge base* and *integrated problem solver* - automatic correction of some errors, the *information waste* removal, the specification of *information holes*, the systematization of acquired *knowledge* and *skills*, the extraction of implicit *knowledge* from the given (inductive inference, *self-learning* by using of precedents);
3) Coordination of the activities of *intelligent system developers (teachers)*. It means that *intelligent system* developers exchange information among themselves only through the *knowledge base* of the developed *intelligent system*. The developers of *intelligent system* become independent *agents* (subjects) of the *intelligent system* development (training), managed by the *knowledge base* of this system. The developed (trained) *intelligent system* also becomes one from these *agents*.

The development of *intelligent system* that has developed *self-learning skills* is fundamentally different from development of *intelligent system* that does not have a high level of *self-learning*. This is due to the fact that the *self-learning intelligent system* becomes one of the subjects of its own training, that is, one of its teachers. And it becomes an essential factor in increasing the effectiveness of training, because no one knows better than the trained *intelligent system* about its *knowledge base*, the *integrated problem solver*.

So ***the method of training the intelligent system*** is largely determined by the tools of *self-learning* that the trained *intelligent system* has.

Areas for *self-learning* of *intelligent system* are:

- acquisition of new *knowledge* from different sources;
- extracting implicit *knowledge* from acquired *knowledge*:
  - •• detection of regularities;
  - •• structuring *knowledge base*;
- maintaining integrity of *knowledge base* (consistency, completeness, flexibility);
- increasing efficiency of solving problems based on analysis of own activities.

V. Learnability of intelligent system

Learnability of intelligent system is a level of ability to solve *self-learning* problems, for example:

- integrate new *knowledge* and *skills* into *knowledge base*;
- identify and resolve *contradictions (*errors) in *knowledge base*;
- structure *knowledge* and *skills* for localization of solution areas of problem sections;
- identify *information holes* in *knowledge base*;
- identify and remove *information waste*;
- analyze the quality of its own activity in solving various problems and learn from its own mistakes.

It is clear that learnability of *intelligent system* is based on ***reflection***, that is, the ability to analyze and evaluate its own quality and quality of its activities, as well as the ability to learn new *knowledge* and *skills* and improve the acquired *knowledge* and *skills* quickly. For example, the learnability of artificial neural networks is determined by the availability of a method for automatically adjusting of *the neural network* which is based on the results of its testing, aimed at constructing of optimal connections' structure, setting up parameters of connections (the stochastic gradient method, the method of back propagation of the error).

Thus, *learnability* of *intelligent system* is ensured by:

1) Systematization of internal representation of *knowledge* and *skills* – all accumulated *knowledge* and *skills* should be brought into coherent system;
2) Fairly simple model for integration (immersion) of new *knowledge* and *skills* into *knowledge base*;

3) Unlimited possibilities to represent in *knowledge base* all necessary information for self-analysis, containing a sign of own ***Myself***, complete own documentation, description of its connections with other *entities* including description of its own point of view from the point of view of other subjects;
4) Ability to *reflection* and a fairly simple model for analyzing quality of the current state of *knowledge base* (quality of *knowledge base* structure, ***completeness of knowledge base***, presence and localization of detected contradictions and errors);
5) Level of development of means of detecting and eliminating unnominal (including erroneous) situations in the process of functioning of *intelligent system*;
6) Level of development of means to improve quality of the current state of *knowledge base* (improving system of accumulated *knowledge* and *skills*).

VI. Unlimited learnability of intelligent systems

The highest level of learnability of *intelligent systems* is ***unlimited learnability***, striving for permanent and unlimited reduction in those problems which *intelligent system* has not managed with or managed not well enough. Unlimited (universal) learnability of *intelligent system* is its ability to acquire (grasp) quickly and without any restrictions and apply new necessary *knowledge* and *skills* effectively, and also ability to modernize and improve the acquired *knowledge* and *skills* without any restrictions.

When unlimited learnability of *intelligent system* implies absence of any restrictions on the types of acquired *knowledge* and *skills*, it requires ensuring compatibility within *knowledge base* of all kinds of *knowledge* and *skills*, as well as all possible models for their presentation, and leads to unlimited expansion of set of problems solved by *intelligent system*.

unlimited learnability
= *universal learnability*
= *ability to acquire, extract and systematize a variety of knowledge and skills, as well as to integrate and joint use them in solving complex problems*

Listing 2. Formal interpretation of the concept of unlimited learnability

Unlimited learnability of intelligent system is provided by:

- variety of used *knowledge* and *skills*' types;
- absence of restrictions on the acquisition and deep integration of fundamentally new (previously unknown) types of *knowledge* and *skills* into *knowledge base*;
- compatibility of used *knowledge* and *skills*.

VII. Flexibility of intelligent systems

Flexibility of intelligent system is the basis of its *learnability*. *Flexibility of intelligent system* is determined by the labor content (simplicity) of making various changes in *intelligent system*, implemented at various levels of *intelligent system* during its *training*.

flexibility
= *level (degree) of flexibility*
= *modifiability*
= *ease of modification*
= *reconfiguration*
= *softness*

Listing 3. Formal interpretation of the concept of flexibility

The principal difference between the characteristic of *flexibility of intelligent systems* and the characteristic of their *learnability* lies in the following: *flexibility* determines the possibility and complexity of various transformations in the *intelligent system*. In contrast, *learnability* determines a fundamental possibility and complexity of holistic transformations of a higher level that translate the *knowledge* and *skills* of *intelligent system* to a qualitatively new content level. So, for example, for the training of *intelligent system*, it is important to bring to its *knowledge base* a new regularity, not previously known to it. Whereas flexibility of *intelligent system* should ensure a fairly simple integration of the regularity's formulation into the composition of the *knowledge base*. Thus flexibility is a kind of "syntactical" aspect of learnability, which is the basis that provides this learnability. Therefore, there may be ***flexible systems***, but ***they don't have the learnability***.

Flexibility of *intelligent system* is determined not only by the labor content of making any changes in the *intelligent system*, but also by the labor content of ensuring *the integrity* of *intelligent system* when the changes are made. For example, new added information to the *knowledge base* of *intelligent system* may conflict with the *knowledge*, which has already located in the *knowledge base*. It means that it is necessary for the maintaining of the integrity (in this case, consistency) of the *knowledge base*:

1) to try finding in the current state of the *knowledge base* all statements that, for various reasons, contradict new information (for example, there may be statements about non-existence, statements about uniqueness);
2) if no such contradictions are found, the new information can be integrated into the current state of the *knowledge base*;
3) if such contradictions are found, then it is necessary to determine for each of them which of the contradictory statements has greater reliability, then either to eliminate (destroy) the contradiction or to fix the fact of its presence with an indication of that the *intelligent system* itself agrees with and that is considered by some other explicitly indicated external subject which is the author of the information entered into the *knowledge base*.

It is much more difficult to ensure the integrity of *intelligent system*, not with the addition of something new, but with adjustment (updating, changing) any fragments of *intelligent system*. Firstly, each change can also come into conflict with the current state of *intelligent system* and the contradictions should be solved. Secondly, for the ensuring of the *intelligent system* integrity some of the changes (for example, the replacement of concepts, the change in the structure of the *knowledge base*) require the implementation of not only these changes, but also a large number of other changes that are the consequences of the former. It is clear that not only the automation of making various changes to *intelligent system* should be provided for ensuring the flexibility of *intelligent system* updating, but also the possibility of rolling back to the previous states of the system (in case of erroneous changes), which requires storing the history of its evolution in *knowledge base* of *intelligent system* (its training).

Flexibility of *intelligent systems* is provided by the ***unification*** of the basic principles of information coding in the *intelligent systems*, memory, the basic principles of *intelligent systems* memory organization, as well as the basic principles of processing information in the memory of *intelligent systems*. As the training of *intelligent systems* comes down to the expansion and improvement of the system of acquired *knowledge* and *skills*, this unification significantly reduces the variety of forms of implementation of processes aimed at *intelligent systems*' training. It is of particular importance for the realization of the unlimited learnability of *intelligent systems*.

VIII. Flexibility of ostis-systems

Flexibility of ostis-systems (systems developed using *OSTIS Technology*) is provided by:

- the basic principles of coding information in the memory of *ostis-systems* are proposed by *OSTIS Technology*, the principles underlie the ***SC-code*** (Semantic Computer Code);
- the basic principles of organizing *ostis-systems* memory which are proposed by *OSTIS Technology* - the principles of *sc-memory* organization, which provides storage and processing of ***SC-code texts***;
- the basic principles of processing information in *OSTIS Technology* in memory of *ostis-systems* (in *sc-memory*).

A. SC-code Principles

Among the basic principles of coding information in memory of *ostis-systems* that provide flexibility of *ostis-systems* are, first of all, those principles that provide ***sense information representation*** with the following characteristics:

1) Among ***the signs*** from ***ostis-system knowledge base*** there should not be pairs of synonymous signs that is signs with the same ***denotation***, ***signs denoting*** the same ***entity***.
2) There should be no duplication of information not only in the form of multiple occurrence of signs denoting the same *entities*, but also in the form of multiple occurrence of ***semantically equivalent texts*** (*knowledge*). In this case, ***the logical equivalence of texts*** is allowed.
3) Among the *signs* from *knowledge base* there should not be *any homonymous signs* that is signs, which can *denote* different *entities* in different contexts.
4) All the *signs* from *knowledge base* should not have an internal structure, which analysis is necessary for understanding ***the sign constructions*** using these *signs*. The internal structure of *signs* is necessary only in those *sign constructions* in which the multiple occurrence of signs having the same *denotation* is allowed, and the analysis of the coincidence of the internal structures of such signs is means of establishing their synonymy or supposed synonymy specified at the stage of subsequent semantic analysis of the *sign construction*. Consequently, all the signs included into *sense information representation* can be *considered* abstract, because they abstract from their internal structure, they have only one characteristic - to designate one-to-one correspondent *denotations*, are invariants of the corresponding maximal ***classes of synonymic signs***.
5) The composition of *the sign construction*, which is *sense information representation*, should not include anything but the abstract *signs* considered above.
6) Within the constructing of *sense information representation*, such constructions as words, terms (phrases) should not be only used, but such language methods as declination, conjugation, punctuation marks (separators, delimiters).
7) From the syntactic point of view, all *the signs* included into *sense information representation* should be clearly divided into two types:
   - ***signs of connections*** between described *entities* (connection is considered one of the types of described *entities*);
   - *signs of entities* that are not connections.

The important advantage of *sense information representation* is that it explicitly and clearly defines the connections between the described *entities* (including the *connections* between the *connections*) in the form of *connections between the signs(!)* of these *entities*, the semantic type of each *connection* is clearly indicated. Stress that all the *connections* of each described entity in *sense information representation* are represented only by a set of *connections' signs* incident to the sign of the indicated *entity,* i.e. a set of *connections*, where the *sign* of the described entity is one of the components. Thus, any *sign construction* can be represented (in a *semantically equivalent* form) as a set of *signs* of the described *entities* and a set of *connection's signs* connecting these described *entities* with other *entities*. At the same time, there are no restrictions on the described *entities* and on the connections between them. The described *entities* can be:

- ***material*** (physical) and ***abstract*** (virtual) - ***numbers***, ***sets***, signs of any *entities*;
- ***really existing*** and ***fictitious***;
- ***fixed (constant)*** and ***arbitrary (variables);***
- *connections* between *entities*.

It is not difficult to see that the *sign construction*, which is the *sense information representation*, can't be ***linear*** in general, because each described *entity*, which is a *denotation* of the corresponding *sign,* can be connected by an unlimited number of connections with other *entities* described in the same *sign construction*. Thus, *sense information representation* is a ***graph structure***, which has a different theoretical-graph configuration. Such *graph structures*, which have the semantic characteristics described above, are also called ***semantic networks*** [2], [3], [4]. The interpretation of *semantic networks* described above makes the possibility of fully using the theory of graphs for studying the syntactic characteristics of ***semantic networks*** and for constructing algorithms for processing *semantic networks*.

It is necessary to clearly distinguish the *semantic network*, which is internal abstract information representation in *the memory* of *intelligent system*, from its coding within the chosen version of the technical implementation of the specified memory, as well as from various versions of its visualizing for users. The semantic network and its graphic or text image are not the same. Similarly, it is necessary to clearly distinguish the *graph structure* as an abstract object from various variants of its representation.

Within *OSTIS Technology*, we speak about creating formal means of describing the meaning of various types of *knowledge* and formal means of describing *knowledge* processing at sense level. This involves the development of the corresponding standard that distinguishes the basic universal language of *semantic networks* from the whole variety of *semantic networks abstract languages*, which is called ***SC-code*** [5]. *SC-code* is a language of unified semantic networks. The texts of this language are called ***sc-texts***. The *signs* in the composition of *sc-texts* are called ***sc-elements***. The transition from the general concept of the semantic network introduced above to unified *semantic networks* (*sc-texts*) is considered as setting a number of restrictions on *semantic networks* of a general form, but such restrictions that do not reduce the semantic power of the *semantic networks language claiming universality* [6].

The consequences of the considered basic principles of building ***knowledge bases* of *ostis-systems*** are:

- unlimited possibility of transition within each *knowledge base* from *knowledge* to ***meta-knowledge***, from *meta-knowledge* to ***meta-meta-knowledge*** and so on, and, consequently, an unlimited possibility of ***structuring knowledge bases*** on a variety of features. This structuring is, first of all, necessary for localizing ***the area of the each problem solution***, that is, for establishing connections between different problems and areas of their solution (those fragments of *knowledge bases* that contain sufficient information for solving the corresponding ***problems***);
- the ability to represent (encode) any (!) ***structure*** in the *sc-memory* in the form of isomorphic *sc-text* whether:
  - ***image structure*** (the structure of the connections between different image fragments);
  - ***the syntactic structure*** of a naturally language text (the structure of connections between different fragments of text at the level of letters, words, terms);
  - *the structure* that reflects the configuration of various connections between any (!) described *entities*.
- complete independence of *sc-texts* from ***terms*** (names) attributed to various described *entities*. These terms are also considered as described *entities*, and the connections connecting these terms to the *entities* which they are assigned to are given by the relation "**to be an *external sign***";

The *SC-code* is not *language* in habitual understanding of this word, and the unified way graph (nonlinear generally) codings of information, providing submission of information of any level (primary, secondary, metalevel) and the description of *communications* between these levels. So, for example, ***primary sc-text*** can be the description of structure of some image, and various fragments of this text can be images of various observed objects which will be presented in the ***secondary sc-text*** by the signs which are obviously connected with the fragments of *primary sc-text* corresponding to them.

The same way *primary sc-text* can be the description of *structure* of the accepted text message, that is primary description of structure of some ***line of symbols***. It is obvious that some fragments of this *primary sc-text* are descriptions of structures of external signs of the corresponding described *entities*, that is structures of the words or phrases entering the message. It is obvious also that the specified described *entities* in the *secondary sc-text* representing sense of the accepted message will be designated by the ***internal signs*** which are obviously connected with the fragments of the specified *primary sc-text* corresponding to them.

B. The principles of memory organization ostis-system

Memory of ostis-system (sc-memory) is a non-linear (graph) associative restructuring (graphodynamic) memory, in which the processing of information is reduced not only to a change in the state of memory elements (any of which is a sign that is a part of the processed sign structure) but also to a change in the configuration of connections, the incidence between them.

From a formal point of view, the memory of *ostis-system* is a dynamic *sc-text*, in which the following events can occur:

- remove ***sc-node*** with the removal of all the *incident* ***sc-links*** (signs of various connections);
- remove the *sc-link* with the removal of all the *sc-links* which it is a component;
- replacing the type of *sc-text* element (for example, *sc-node* can be converted to ***sc-connector***);
- adding a new *sc-text* element with mandatory indication of the connection of the new *sc-element* generated in memory with any *sc-element* already presented (stored) in memory.

In *sc-memory* it is possible to store and process any ***external information constructions*** (terms, hieroglyphs, texts, images, video information, audio information) presented in electronic form as corresponding ***files***. At the same time each *file* in *sc-memory* is represented by the corresponding *sc-node* denoting this file (actually, the *external information structure* encoded by this file), and the specified file is considered the content of the denoted *sc-node*. Through such files, in particular, information is exchanged between *ostis-system* and external subjects (users, other *ostis-systems*).

For ensuring *flexibility* of *ostis-systems* the developed form of associative access to the any kinds of *knowledge* and *skills* stored in memory of *ostis-systems* thanks to the developed means of the specification of required *knowledge* is of particular importance.

So, for example, the procedure of associative search of *signs* on the basis of aprioristic information on communications between *signs* significantly becomes simpler. Such search is carried out by means of wave navigation on space of the connected *signs*.

Significantly also the procedure of associative search of the fragments of the stored *knowledge base* satisfying to the set *inquiries* (requirements) becomes simpler. Significantly the variety of types of such *inquiries* extends:

- *inquiriy* of fragments of the *knowledge base*, isomorphic to the set sample;
- *inquiriy* of fragments of the *knowledge base*, homomorphic to the set sample;
- *inquiriy* of the fragments of the *knowledge base* satisfying to the set not atomic logical formula that is the fragments satisfying to the set logical properties;
- request of the specification (the semantic vicinity) the set essence;
- *inquiriy* of the *entities* similar to the set essence (including to the set *knowledge base* fragment);
- *inquiriy* of communications (including similarities and differences) between the set *entities*.

C. The principles underlying the integrated solver of ostis-system problems

The integrated solver of ostis-system problems is a hierarchical system of ***agents*** that implement the *knowledge base* processing represented in *SC-code* and stored in *sc-memory*, and which interact with each other only through the specified *sc-memory*. Thus, the whole process of processing the *knowledge base* in *ostis-system* is controlled by this *knowledge base*.

These *agents*, which are called sc-*agents*, are divided into:

- ***non-atomic sc-agents***, which are collectives of sc-*agents* of lower level;
- ***Atomic sc-agents***, which are not collectives of sc-*agents*.

In its turn *atomic sc-agents* to their implementation level are divided into:

- ***atomic programm's sc-agents*** implemented in the ***basic programming language*** of *sc-agents*;
- **atomic sc-agents of the base programming language interpreter of sc-agents** (the interpreters of *knowledge base* and integrated solver of *ostis-system* problems).

In this case, each *sc-agent* has its own class of situations or events in *sc-memory* that initiate the activity of this *sc-agent*, generating the corresponding ***information process in sc-memory***, the main characteristics and the current state are described in *sc-memory* and used in this process.

Example1. Here is an example of parallel asynchronous multi-agent solution of a problem in *sc-memory*. Consider a problem of calculating the indicated number.

Calculation is the construction of representation of a number in a given system of calculus (for example, decimal).

Example of an agent-oriented model for solving problems in *ostis-system*.

Example problem: calculate the value of a number. This *problem* is a condition for the activation of a number of *agents*. It is done with the help of a meta-agent, which activates *agents*, focused on solving particular problems:

- ***sc-agent addition agent***
  - checks whether the specified number is the sum of the other numbers. Moreover, if these other numbers are calculated,then addition is performed, and if any of the terms is not known, a subtask for its calculation is generated, and the information process goes into the waiting state for solving this subtask;
- ***sc-agent subtraction agent***;
- ***sc-agent multiplication agent***;
- ***sc-agent division agent***;
- ***sc-agent involution agent***;
- ***sc-agent evolution agent***;
- ***sc-agent logarithm agent***;

Unfortunately, not every problem for calculating a number can be solved with the help of the specified set of ***arithmetic sc-agents*** and even with the help of a wider set of *sc-agents* that provide the calculation of all possible numerical functions. Examples of such problems are the problems of solving quadratic equations (Figure 2), the problem of solving systems of linear equations, etc. To solve such problems additional *sc-agents* are introduced, in particular, those that reformulate the original problem so that it can be solved, for example, only with the help of *arithmetic sc-agents*. Thus, for example, *sc-agent for solving quadratic equation* replaces the formulation of the problem of solving a quadratic equation (Figure 2) in the form shown in Figure 3.

Moreover, this kind of problems reformulation can be described in the form of ***products*** and included in *knowledge base* (actually, in skill base), and then implement *sc-agent* that provides interpretation of any products stored in ***ostis-system skill base***.

The unambiguous representation of information *in memory of ostis-system*, that is, the elimination in *SC-code* of duplication (semantic equivalence) of both *signs* and whole *sign constructions* significantly simplifies the processing of *knowledge base*.

Listing 4 provides clarifying of some concepts that underlie the organization of integrated problem solver of ostis-system.

operation
= *purposeful holistic process implemented by a certain subject within some dynamic system and effecting a change in the state of this system*
= *action*

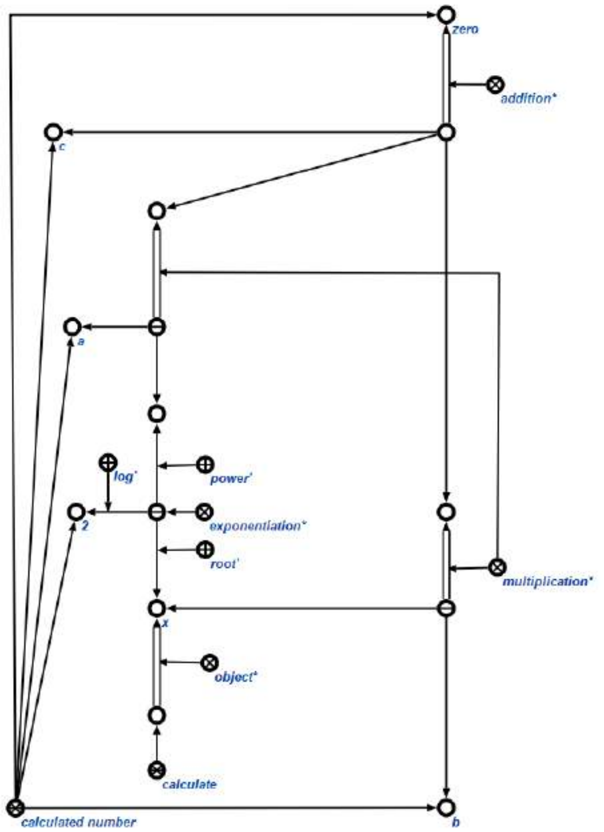

Figure 2. Formulation of the problem of solving a quadratic equation in SC-code

intelligent system operation
= operation implemented by intelligent system
⊃ operation
*<= partitioning **
{
- *internal operation of intelligent system*
  = information process in memory of intelligent system
  = operation implemented in memory of intelligent system and aimed at changing the state of the stored knowledge base
- *external operation of intelligent system*
  = operation of intelligent system implemented in its external environment
  ⊃ atomic external operation of intelligent system
    = external operation of intelligent system implemented by one of its effectors

}

problem
= pragmatic specification of some (given) operation
= formulation of problem
⊃ problem of intelligent system
  = specification of intelligent system operation

Listing 4. Clarifying concepts underlying the organization of integrated problem solver of ostis-system

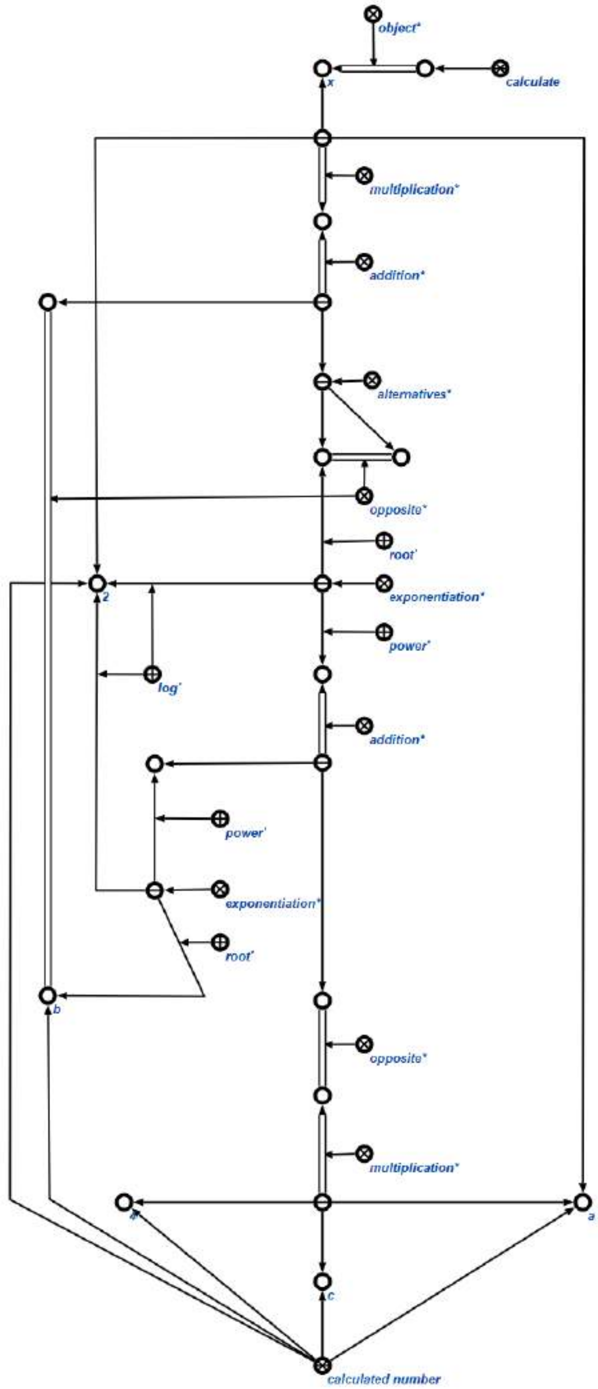

Figure 3. Result of reformulating the problem of solving a quadratic equation

Formulation of *problem*, as specification of the corresponding *operation,* can include:

- indication of ***classes of operations*** (preferably minimal) which the specified operation belongs to;
- indication of what should be done
  - •• either by indication of the ***target situation***;
  - •• or by indication of the *class of operations* which this *operation* belongs to with additional indication of objects which this *operation* is implemented with (action arguments);
- condition for initiating the specified *operation*. This condition can be either the completion of some other specified *operation,* or at least one of the specified *operations*, or the completion of all the specified *operations*, or the occurrence of the specified *situation* in *knowledge base*, or at least one of the specified *situations*, or the occurrence of the specified ***event*** in *knowledge base*, or at least one of the specified *events*, or all of these *events*;
- indication of a plan for the implemention of the specified *operation,* that is, hierarchical decomposition of this operation into operations of lower level;
- indication of an area for the implemention of the specified operation, that is, the structure included in *knowledge base*, which contains all (!) data that may be required to implement the specified operation;
- indication of temporal characteristics of the specified operation (start and / or end time, duration);
- indication of a customer for this operation;
- indication of implementers, that is, means of implemention of the specified operation (for internal specified operation of *ostis-system*, such mean is set of its several sc-*agents*);
- indication of status of the operation, that is, the indication that the specified implementer or collective of implementers may or may not, want or not want, should or should not implement the operation;
- indication of priority of the specified operation;
- indication of the state of the specified operation, which can be planned, implemented, directly implemented at the moment, interrupted.

skill
= skill of intelligent system
  ⊃ Skill of ostis-system = pragmatic specification of some (given) class of operations
= method (model) for solving problems belonging to given class of problems
= method to implement operations belonging to given class of operations
= knowledge of how to solve problems of given class
= knowledge of how to implement operations of given class

Listing 5. Formal interpretation of the concept of skill

In contrast to other types of *knowledge*, *skills* (Listing 3) have not only *denotation*al, but operational (functional) semantics, which describes what operations should be implemented when this skill is used.

Skill of ostis-system is called *knowledge*, which is a part of *knowledge base* of this system and which is sufficient to solve given class of problems (that is, to implement given class of operatons) within given *knowledge base* area using given family of *ostis-system agents*.

Each *Skill of ostis-system* is represented in *knowledge base* in the form of a specification of the *class of operations* implemented by using this skill.

Skill of ostis-system as a specification of ***some class of ostis-system operation*** includes:

- indication of the corresponding class of *ostis-system operation*;
- generalized formulation of *problems* of the corresponding class, that is, generalized specification of operations of the specified class (indicating what should be given and what is required);
- indication of area for solving all problems of given class, that is, the fragment of *knowledge base* that contains all the necessary information for solving all problems of given class;
- indication of declarative or procedural program, which interpretation guarantees implementation of all or almost all operations belonging to the given specified *class of operations*;
- means for implementation operations of given class, which are set of sc-*agents* that ensure implemention of all or almost all operations belonging to the specified *class of operations* and defining the operational semantics of the interpreted program mentioned above and, accordingly, the operational semantics of whole skill.

Stress that in most cases, area for solving problems is a certain ***subject domain*** together with its ***integrated ontology***, and interpreted program is ***logical ontology*** of the specified *subject domain*.

Hierarchy of *ostis-system skills*:

- skill of base (zero) level – the skill realized by one of the *agents* of base machine of *ostis-system*, which is the interpreter of *knowledge* and *skills* of *ostis-system* stored in its *sc-memory*;
- skill of level 1 – the skill, implemented by collective of all *agents* of base machine of *ostis-system*;
- skill of level 2 – the skill, implemented by one of *atomic program* sc-*agents*;
- skill of level 3 – the skill, implemented by collective of *atomic program* sc-*agents*;
- skill of level 4 – the skill implemented by collective consisting of *atomic program* sc-*agents* and collective consisting only of *atomic program* sc-*agents*.

Thus, activity of *ostis-system* is clearly *stratified* into several sufficiently independent levels, each of which ensures implementation of clearly defined set of operations:

- level of operations implemented by collectives of sc-*agents*;

- level of operations implemented by *atomic program* sc-*agents*;
- level of operations implemented by interpreter of *ostis-system knowledge* and *skills* when interpreting programs of *atomic program* sc-*agents*.

D. Flexibility of knowledge bases sc-models

The unambiguity of information representation in memory of *ostis-system*, that is, the elimination in *SC-code* of duplication (semantic equivalence) of both signs and whole *sign constructions* essentially simplifies the processing of the *knowledge base*.

The labor content of editing *sign constructions* stored in *sc-memory* at a semantic level essentially reduces compared, for example, with the editing of database or texts of natural languages. In particular, it is much easier to remove a sign from *sign construction* in *SC-code* without damage to its integrity.

Rules for removing *sc-elements*:

1) If *sc-element* is deleted, then all (!) *sc-links* (both atomic and non-atomic) incident to it are deleted recursively;
2) If ***sc-arc of membership*** leaving the sign of the ***formed set*** is deleted, then the indication of the formation of this set is deleted and an indication of the power of this set is added (if it was absent).

Flexibility of knowledge bases sc-models is also ensured by a clear separation of external and *internal signs* and complete independence of *knowledge bases* sc-model from the variety of forms and languages of this *knowledge base* external representation.

Here are some examples of editing *knowledge base* sc-model.

Example 2. Replacement of the term

A name (term) of each described entity is present in *ostis-system knowledge base* in one instance and is represented in *knowledge base* by *sc-node* denoting this name, which is connected by the relation external *sign with *sc-element* denoting the same entity as the specified name (Figure 4), and this *sc-element*, in turn, is connected by unlimited number of connections with *sc-elements*, which denote other *entities* described in *knowledge base*.

Therefore, replacement of a name (term) of an entity is implemented only in one place of *knowledge base*.

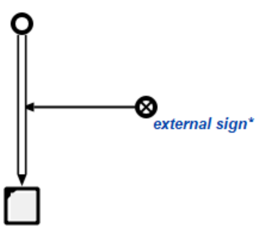

Figure 4. The relationship of a term with a sc-element that identify the same entity

Example 3. Replacement of concepts

Instead of the concept of *partitioning* * of set, proceed to use the following concepts:

- *union** of sets (covering a set);
- *family of pairwise disjoint sets*.

For such replacement, you need a concept that becomes unused to define through the used concepts (Figure 5).

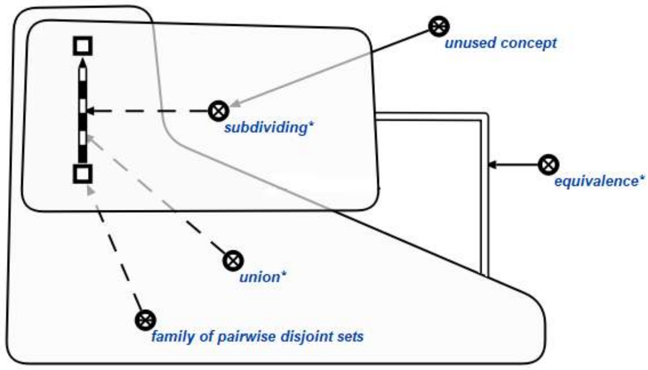

Figure 5. Definition for concept of partitioning*

The purpose of this kind of replacement of concepts is to minimize the number of defined concepts (some defined concepts are made unused). To eliminate the concept means to remove all arcs of membership from it, leaving the concept as unused.

1) removal of all arcs of memdership emerging from it;
2) construction of logically equivalent specifications for each instance of the concept presented in *knowledge base*;
3) inclusion of this concept in the number of unused concepts.

E. Flexibility of ostis-system integrated solver is provided by:

- autonomy of *sc-agents*;
- interaction of *sc-agents* only through *sc-memory*;
- using the emergence in *sc-memory* of formulations of initiated problems as the main events initiating activity of the corresponding *sc-agents*;
- developing of powerful language of the formulations of various kinds of problems on the basis of *SC-code*;
- clear separation of variety of various skills of the solution of the tasks stored by *ostis-system*, from the interpreter of these skills and ensuring independence of the specified skills of variety of options of realization of the specified interpreter;
- the fact that the skills acquired by *ostis-system* are a part of her *knowledge base* and are stored in her memory, therefore, them easily to fill up and correct.

Correction of *knowledge base* is based on the fact that there is no duplication of information both at *sc-element* level and at sc-construction level in *knowledge base*, it means that all such corrections are of local character (within the semantic area of fragment correction is not required global revision). But it is necessary to pay for such locality, it should be constantly maintained. This requires:

- constantly improve structure of *knowledge base* (system of subject domains and ontologies);
- use special sc-agents in *knowledge base* of *ostis-system* to perform permanent analysis of the quality of *knowledge base* for the presence and elimination of synonymous sc-elements and semantically equivalent sc-constructions (not to be confused with logical equivalence);
- each *sc-agent* within each *atomic sc-process* complies with certain rules of *sc-memory* usage that are common for all *sc-agents* of *ostis-system*
  - •• removes *information waste* (information necessary only for performance of the specified atomic process);
  - •• ensures that synonyms and semantic equivalent *sc-texts* weren't generated;
  - •• ensures that that the *atomic processes* which are carried out by him didn't interfere with each other.

F. Flexibility of ostis-systems

in general lies in the local character of various kinds of *intelligent system* transformations. It means that consequences of transformations should be considered within the known and clearly defined parts of system (fragments of *knowledge base* of integrated solver). When strict requirements presented by *OSTIS Technology* to the integrity of *intelligent systems* are observed, you can implement any(!) changes in *ostis-system knowledge base*, in sc-model of its integrated solver in base interpreter of sc-model of *intelligent system*. Compliance with these requirements ensure locality of these changes, that is, complete independence of some changes from others.

Localization of problem-solving processes in *ostis-system* is implemented due to the fact that *knowledge base* of *ostis-system* has hierarchical structure that provides clear allocation of solving area for each problem of the fragment of *knowledge base* which is sufficient for solving one form of problem or another. The structure of *knowledge base* is based on hierarchical system of *subject domains* and corresponding ontologies.

IX. Learnability of ostis-systems

Learnability of ostis-systems is expressed in the following:

- detection and localization of contradictions in *knowledge base*;
- elimination of some contradictions in *knowledge base*;
- detection and localization (specification) of *information holes* in *knowledge base*;
- detection and elimination of *information waste* in *knowledge base*;
- analysis of the solver performance in solving various kinds of problems;
- detection of contradictions (errors) that appeared in *knowledge base* as a result of the activity of sc-*agents* specifying specific problems and specific atomic information processes (as sources of errors);
- detection of contradictions, which are consequences of activity of *knowledge base* developers , with indicating execution time and authors of these actions and with organization of interaction with the developers to eliminate these contradictions;
- constant updating of enumeration of problems which could not be solved or succeeded, but not as desired (insufficiently accurate, insufficiently fast, insufficiently transparent, insufficiently simple, insufficiently clear);
- permanent improvement of *knowledge base* structure, the purpose of which is to increase accuracy of information resources localization sufficient for solving various problems solved by *intelligent system*, including fundamentally new problems for it;
- permanent search and improvement of ways and effective models for solving fundamentally new problems for *intelligent system*, which are necessary to be solved during functioning of *intelligent system*.

A. Ability to acquire new knowledge generated by SC-agents of ostis-system in internal sense language

To acquire new *knowledge* generated by some sc-agent of *ostis-system* and, accordingly, located in its *sc-memory*, is to immerse (integrate) the specified *knowledge* into *knowledge base* of *ostis-system*. The most important advantage of *SC-code* is that this procedure of immersing (integrating) new *sc-texts* into *knowledge base* is reduced to constructoin of correspondence of synonymy between *sc-elements* (both *sc-nodes* and *sc-connectors*) of new immersible *sc-text* and *sc-elements* that are part of the current state of *knowledge base*, followed by gradual gluing (identification) of synonymous *sc-elements*. The construction of this correspondence of synonymy is implemented, firstly, by the coincidence of main names * (names, *external signs*) corresponding to some *sc-elements*, and, secondly, on the basis of statements describing existence and uniqueness of *entities* possessing given characteristics. Stress that not all *sc-elements* are named, that is, having corresponding external names (*external signs*). Only the used concepts should be named, each of which should be researched at least in one of the selected *subject domains*. In addition, several names can correspond to some *sc-elements*, one of which is allocated as the main name * of the corresponding *sc-element*, which the lack of synonymy and homonymy towards the *main names** of other *sc-elements* is guaranteed for.

Example 4. An example of immersing a new *sc-text* in *knowledge base*. Figure 6 shows *sc-text* that is in the current state of *knowledge base*. Figure 7 shows *sc-text* that is immersed in *knowledge base* (introduced into *knowledge base*). Figure 8 shows the result of gluing *sc-elements* by the coincidence of their *main names**, after that several new *entities* appeared in the current state of *knowledge base* – a new triangle and two bisectors and their intersection point, a new circle corresponding to the new triandle.

Figure 9 shows the final result of immersion of considered new text in *knowledge base*, derived on the basis of using the following statements about uniqueness stored in *knowledge base* of ostis system:

1) Each *triangle* has a single *bisector* having a common element, which is the *vertex* of this triangle.
2) Every two *triangles* having the same *vertices* are the same triangle.
3) Each *triangle* has a unique inscribed circle.
4) All bisectors of one *triangle* intersect at one *point*.
5) The point of intersection of its bisectrix and the center of the inscribed circle coincide for every triangle.
6) Each two-fold constant positive *arcs of membership* going out of the same set that is not a multiset is the same arc.
7) Membership *bisectrix** does not have multiple pairs.

Comparing Figure 8 and Figure 9, it is easy to see that there was not really much new information contained in the new immersed *sc-text* – there is only another bisector of the already known triangle and that the center of the inscribed circle coincides with the point of intersection of its bisectors.

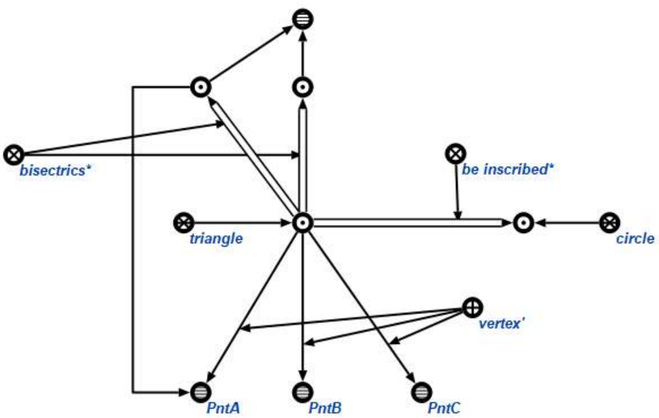

Figure 6. *Sc-text* that is in the current state of *knowledge base*

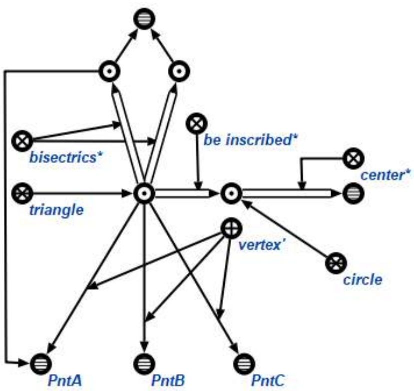

Figure 7. *Sc-text*, immersed in *knowledge base*

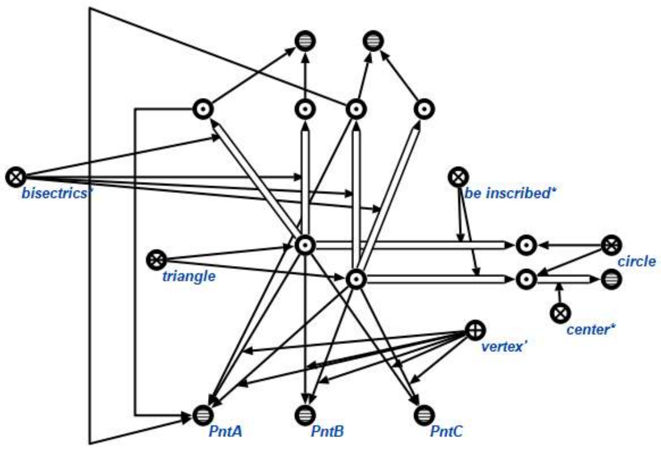

Figure 8. The result of gluing *sc-elements* by the coincidence of their main names

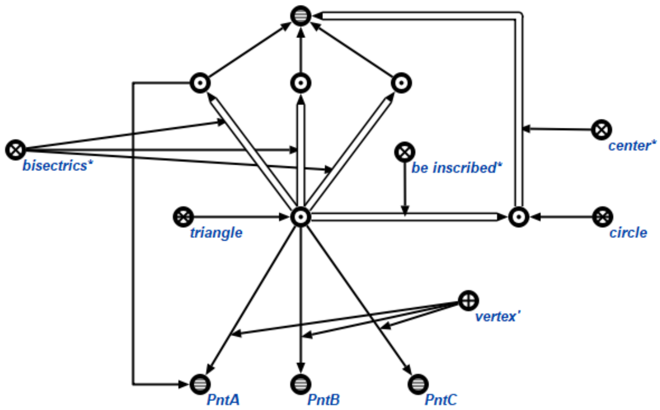

Figure 9. The result of gluing unnamed synonymous *sc-elements* (the final result of immersing introduced sc text into *knowledge base*)

B. Ability to acquire new knowledge and skills coming from outside in formal languages

As the main languages providing the representation of *external texts* for *ostis-system* are used:

- ***SCg-code*** (Semantic Computer graphical Code), which provides graphic representation of *SC-code* texts, in which *sc-nodes* are represented as circles, squares, closed lines representing signs of sc-structures whose images are limited by these lines, and *sc-connectors (sc-arcs* and sc-edges) are represented as lines connecting *sc-node* images;
- ***SCs-code*** (Semantic Computer string Code), which provides string (linear) representation of *SC-code* texts and used by *ostis-system* for input and output of relatively short messages;
- ***SCn-code*** (Semantic Computer natural Code), which provides intuitively understandable formatted representation of large *source code*.

The syntax and declarative semantics of these languages are close to the syntax and semantics of *SC-code*. Thanks to this, the syntactic and semantic analysis of input texts is substantially simplified, as well as the translation of *sc-texts* in specified external languages when messages addressed to users are generated. Proximity of the mentioned external languages to *SC-code*, first of all, is that all used names are the *main names** (main *external signs**) of the corresponding *sc-elements*. Stress that the list of used external formal languages for *ostis-systems* is easily extended, because it is not very difficult to describe syntax and semantics of any external formal language using the *means of SC-code* within ***interface knowledge base***, and it is not difficult to develop a family of *sc-agents* that provide:

- making ***syntactic analysis*** of external source code based on the description of syntaxis of the corresponding external language. The result of this analysis is *sc-text* describing the *syntactic structure* of the external source code with such a degree of detailing that is sufficient for the subsequent ***semantic analysis*** of this code;
- making *semantic analysis* of external source code based on the description of semantics of the corresponding external language.

The result of this analysis is construction of *sc-text* semantically equivalent to external source code. After that, the immersion of *sc-text* described above into the current state of *knowledge base* is implemented.

But a better understanding by *ostis-system* of the accepted external message implies the implementation of the following additional actions:

- Coordination of *the concepts* used in the accepted external message with the concepts used in the current state of *knowledge base* of *ostis-system* (see example 3);
- Inclusion of acquired *knowledge* into general system of accumulated *knowledge*, in particular, into hierarchical system of the selected *subject domains* and corresponding *ontologies*.

C. Ability to acquire new knowledge and skills coming from outside in non-formal languages

Acquisition (understanding) of *knowledge* entering *ostis-system* in ***natural languages*** consists of the same stages as the process of understanding external texts of formal languages considered above, but fundamentally differs from the understanding of external texts of formal languages due to the fact that the syntax and semantics of natural languages have a significantly higher level of complexity compared with the syntax and semantics of formal languages. It means that the development of ***knowledge base of ostis-system natural-language interface***, as well as the development of system of *sc-agents* of the interface is very complex problems.

Nevertheless the principles which are the cornerstone of development of ***verbal interfaces of ostis-systems*** have the serious advantages which are that external (including, and natural language) texts are considered by *ostis-system* as a part of her external environment and can be described in memory of *ostis-system* in her internal language (in a *SC-code*). Thus, the structure of the external text can be described with any extent of specification means of a *SC-code*. Therefore, ***semantics*** of the external text will represent a morphism between the *sc-text* describing structure of the source text with the required extent of specification, and the *sc-text* which is semantic equivalent to the specified source text.

That fact, as the description of structure of the source text, and the *sc-text* semantic equivalent to him, and the description of a morphism between them are presented in one language (*SC-code*) and are in one memory (*sc-memory*), creates good prerequisites for effective realization of semantico-syntactical approach to the analysis of the source text assuming lack of clear split of process of the analysis of the source text in time for a stage of parse and a stage of the *semantic analysis*. Such approach allows to reduce labor input and the syntactic and *semantic analysis* thanks to preliminary results of the semantic analysis when performing parse and vice versa.

D. Ability to acquire knowledge coming from outside through sensors of ostis-systems

Advantage of *ostis-systems* in the analysis of primary sensor information, as well as in the analysis of external texts, is defined by that, as intellectual (in that number, semantic) the analysis of this information it is easy to present primary sensor information, secondary sensor information and result in memory of *ostis-system* in the form of *sc-texts* that allows to pass easily from the "syntactic" level of sensor information to "semantic" level. It is relevant and for different identification of emergency situations in difficult, for example, production, systems to which it is necessary to react quickly, and for ***the semantic analysis*** of images and scenes.

It is necessary to pay attention to analogy of transition from the description of *syntactic structure* of the source external text to sense of this text (that is to semantic equivalent *sc-text*) and transition from the description of structure of primary and secondary sensor information to the description of communications between those objects which are reflected in this sensor information.

Thus, recognition of various objects in a flow of sensor information is an analog of transition from the phrases which are a part of the source external text, and, more precisely, from the *sc-texts* describing structure of these phrases to the *sc-elements* designating the same *entities*, as the specified phrases.

E. Ability to detect and eliminate contradictions in knowledge base

Associative search for sc-structures that satisfy statements about non-existence and uniqueness is generalized description of errors.

Example 5. Here is an example of contradictions that are the consequence of the actions of *knowledge base* developers.

Suppose that in the fragment of *knowledge base* shown in Figure 10, *knowledge base* developer introduced updated

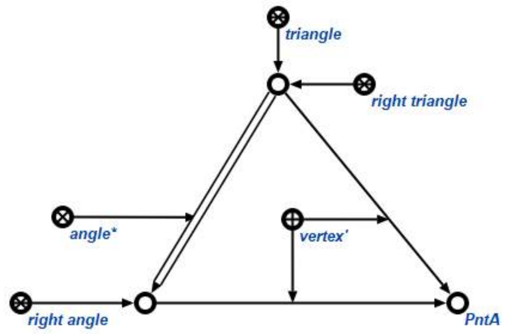

Figure 10. *Knowledge base* fragment about triangle

information about the magnitude of the considered angle with the vertex at point A and let this magnitude be 88^0.

Then:

- the considered angle ceases to be right;
- the considered triangle ceases to be right;
- if the magnitudes of remaining angles of the considered triangle are known, then the sum of the angles of the triangle becomes less than 180^0, and this contradicts the well-known theorem. It means that the magnitudes of the other angles of this triangle should be clarified.

F. Ability to search and eliminate information holes in knowledge base

Search and elimination of *information holes* in *knowledge base* is implemented on the basis of:

- generalized description of required typical specifications for different classes of *entities*;
- generalized quasi-plicative statements describing not what must exist in given infinite *subject domain* in given conditions, but what must be present in *knowledge base* in given conditions.

The following provisions are the basis of detection and specification of *information holes* in *knowledge base*:

- any information should be included in any *subject domain* and/or any specific ontology;
- there is typical structure of mandatory and desirable specification of each instance of this class for each entity class;
- these classes have hierarchy (!) with inheritance of typical structures of specifications of their superclasses' instances;
- unambiguous specifications play important role (for example, representation of numbers in different systems of calculus); classes of *entities* that have typical specifications, in particular, include:
  - •• *entity*,
  - •• *set*,
  - •• *structure*,
  - •• *subject domain*,
  - •• *concept*,
  - •• *relation*,
  - •• *binary relation*,
  - •• *external information construction*,
  - •• *person*,
  - •• *number*,
- any ***concept*** should
  - •• included in any ***subject domain*** as the researched (class of researched objects, researched characteristic, researched relation);
  - •• be defined or explained;
  - •• have set of propositions describing the regularities which are satisfied by instances of this concept;
  - •• have typical examples (instances).

G. Ability to search and eliminate information waste in knowledge base

The specified ability is realized, first of all, by each sc-agent, because each of them after the completion of each actions is obliged to "clean up after itself", that is, to remove all auxiliary structures created only for implementation of corresponding action. In addition, special sc- *agents* are used to eliminate *information waste*. Activity of these sc-*agents* is the opposite of activity of sc-*agents* for removing *information holes*, but it is organized similarly on the basis of generalized quasi-replicative statements which describe that should be removed from the current state of *knowledge base* in given conditions.

H. Ability to systematize the acquired knowledge and skills

As a result of such systematization, each acquired *knowledge* and, in particular, skill should be included in one or several structures that are the results of decomposition of *knowledge base* of *ostis-system* on various grounds. One of the main types of such decomposition is logical-semantic decomposition of *knowledge base*, the result of which is hierarchical system of *subject domains* and corresponding ontologies.As a result of such systematization, each acquired *knowledge* and, in particular, skill should be included in one or several structures that are the results of decomposition of *knowledge base* of *ostis-system* on various grounds. One of the main types of such decomposition is logical-semantic decomposition of *knowledge base*, the result of which is hierarchical system of *subject domains* and corresponding ontologies.

X. Unlimited learnability of ostis-systems

Unlimited learnability of ostis-systems is provided by:

1) Universality of *SC-code*, that is, ability to represent any (!) types of *knowledge* and *skills* in *SC-code*.
2) Universal character of the ***agent-oriented organization of knowledge processing***, which allows on its basis to interpret any (!) models of information processing (sequential and parallel, synchronous and asynchronous, functional, production, genetic, neural network, logical - clear and fuzzy, accurate and plausible , deductive, inductive, abductive, argumentative).
3) ***Compatibility*** of all *knowledge*, *skills*, models, information processing used in *ostis-system* in appropriate coordination with used concepts.

All variety of *knowledge* types and variety of problem-solving models types can be formally described in the form of hierarchical system of interrelated *subject domains* and ontologies. In other words, any kind of *knowledge* can be treated as a language that is sublanguage of *SC-code*, the union of the texts (*sc-texts*) of which is the corresponding *subject domains*, and the description of semantics of which is ontology of the specified *subject domain*. As for problem-solving models, each of them is formally defined by (1) a ***language of problems***, which is sublanguage of *SC-code*, and in particular, describing the ***decomposition of problems into subproblems*** (the union of the texts of such problems language is ***subject domain of problems*** solved within the corresponding *subject domain*) and (2) a set of sc-*agents* that ensure all problems solution within the specified *subject domain* of problems, that is, ensure interpretation of the specified problems language and define ***the operational semantics*** of this language.

Since all kinds of *knowledge* and problem-solving models can be represented using *SC-code* as a part of integrated *knowledge base* of *ostis-system* stored in *sc-memory* of this system, and since different kinds of *knowledge* and different kinds of problem-solving models will intersect if they contain signs with the same *denotations*, the joint use of different types of *knowledge* and problem-solving models is not very difficult. Ensuring the joint use of different types of *knowledge* and different types of problem-solving models makes it possible not only to compensate for disadvantages of some models with the advantages of others, but also to significantly expand the range of solved problems, firstly, due to different problem-solving models provide solutions of different types of problems in general (sometimes even intersecting), and, secondly, due to possibility of solving complex problems which can not be solved in principle with the help of one model and can be solved only by the joint use of different problem-solving models.

XI. Training and self-learning of ostis-systems

Training ostis-systems is a process of improving (designing) *ostis-system*, implemented during its operation.

The efficiency and, first of all, the speed of *training of ostis-systems* is defined not only by the high level of *learnability of ostis-systems*, but also efficiency of ***the ostis-systems training technique*** realized by group of *developer teachers*. In spite of the fact that there are general principles of the organization of project management, accounting of specifics of designed projects (in this case, *ostis-systems*) will allow to increase efficiency of such management significantly. In particular, when training *ostis-systems* there can be very useful that experience which the mankind had accumulated when training children.

When training *ostis-systems* increase in level of activity of the trained *ostis-system* in the course of the training is perspective. It has to be carried out (1) in the direction of development of means of self-training and (2) in the direction of increase in level of automation and intellectualization of management of activity of *developer teachers*.

In this case the developer of *ostis-system* becomes the external agent working on the *knowledge base* of this system and interacting with other developers through the same memory in which the specified *knowledge base* is stored.

XII. The relationship between flexibility and learnability of intelligent systems

Flexibility is a system attribute that characterizes the ability (freedom) to make the system changes that are being implemented or t *Flexibility* is characteristic of system that characterizes possibility (freedom) of introducing certain changes into system, which is implemented either by system's teacher-developers or by the system itself.

Learnability is characteristic of system that characterizes ability to use its own flexibility (that is, freedom of self-change) so that quality of system increases. In particular, learnability is ability to compensate for the negative consequences of teacher-developers actions, fraught with destruction of system integrity.

The simplest form of training *intelligent system* is loading new *knowledge* and / or *skills* into its memory, or "manually" editing *knowledge* and *skills* which are already in memory. But even such simple form of training requires from *intelligent system* not only appropriate level of *flexibility*, that is, providing quite convenient possibility of "manual" replenishment and editing of *knowledge* and *skills* of *intelligent system*, but also appropriate level of their *learnability*.

Principle possibility "manually" and without any restrictions to replenish and correct programs and processed data stored in memory is present in modern computer systems. But, firstly, you can not follow every byte in large systems; secondly, "manual" intervention in large system (for example, changing relational database schema) can have difficult predictable consequences and is powerful source of errors introduced into system.

Consequently, it is important not only to ensure high level of *flexibility* in large computer systems and, in particular, to automate changes introduced to system, but also to ensure sufficient level of *learnability* to prevent negative consequences of these changes.

This is the transition from systems with high level of *flexibility* to *trained* systems. Stress that transition to trained systems is natural standard for evolution of computer systems when their level of complexity increases (with increase in amount of processed data and number of used programs).

Here are some examples of the relationship between the features of *flexibility* and the features of *learnability* of *intelligent systems*:

1) Artificial neural network:
   - *flexibility*: easy ability to change parameters of artificial neural network;
   - l*earnability*: the presence of procedure that changes parameters of artificial *neural network* so that the efficiency of solving problems for which it is intended increased (for example, procedure for back propagation of errors);

2) *ostis-system*:
   - *flexibility*: there are ample opportunity for editing *knowledge base* (including its restructuring, reconfiguration) and high level of such editing automation;
   - *learnability*: there are developed means that ensure the integrity of *knowledge base* (its consistency, completeness, clarity);
3) *ostis-system*:
   - *flexibility*: there are advanced forms of associative access to the searched (askable) fragments of *knowledge base* due to the lack of information duplication in *knowledge base* (that is, the lack of synonymy of internal *skills* and semantic equivalence of internal *sign constructions*);
   - *learnability*: significant simplification of means for analyzing quality of the current state of *knowledge base*;
4) *ostis-system*:
   - *flexibility*: local character of all changes made in *ostis-system* (and, first of all, in its *knowledge base*) due to clear stratification of *ostis-system*, which ensures high degree of independence of changes in different parts of *ostis-system*;
   - *learnability*: significant simplification of means ensuring integrity of *ostis-system*;
5) *ostis-system*:
   - *flexibility*: possibility within internal sense code of unlimited transition from *knowledge* to *meta-knowledge*;
   - *learnability*: ability to specification and systematization of acquired *knowledge* and *skills* and, as a result, ample opportunities for decomposition and structuring of *knowledge base* and *skills* on various grounds;
6) *ostis-system*:
   - *flexibility*: unification of internal representation of *knowledge* and *skills* in memory of *ostis-system* on the basis of hierarchical system of *subject domains* and ontologies, and as a result, simplification of procedures for editing *knowledge base* due to significant reduction in the number of pairs of logically equivalent fragments of *knowledge base*, and also due to the rule of inheriting characteristics with the construction of specifications for instances of concepts that are subclasses of other concepts known to the system;
   - *learnability*: ability to ensure compatibility of *knowledge bases* of various *ostis-systems* and, as a result, to ensure compatibility of these systems;
7) *ostis-system*:
   - *flexibility*: possibility of detailed logging in ostis-system memory of all internal processes occurring in this memory;
   - *learnability*: significant simplification of means to establish causes and sources of emergence of various kinds of contradictions (including conflicts between internal information processes);
8) *ostis-system*:
   - *flexibility*: possibility to change parameters of *integrated problem solver* (in particular, level of each internal agent of *ostis-system* activity);
   - *learnability*: constantly improved procedure aimed at improving performance and quality of functioning of *integrated problem solver* of *ostis-system* (in particular, by redistributing levels of internal *agents* activity).

XIII. Flexibility and learnability of OSTIS technology

Flexibility and unlimited learnability of ***OSTIS Technology*** are determined by the fact that the specified technology is implemented in the form of *ostis-system*, that is, it is implemented using the same *OSTIS Technology*, the support of which it implements. The specified *ostis-system* is called ***metasystem IMS.ostis*** (Intelligent Meta System Open Semantic Technology *Intelligent systems*).

Thus, we focus not only on creating *OSTIS Technology*, but rather on creating process that supports high rates of permanent evolution of this technology.

XIV. Training and self-learning of OSTIS technology

Training and *self-learning* of *OSTIS Technology* (that is, permanent improvement of metasystem *IMS.ostis* during its operation) have a number of features:

1) *IMS.ostis* training (improvement) is implemented within open project that lets anyone to enter the number of developers of this *metasystem*.
2) ***metasystem IMS.ostis*** *self-learning* can be implemented on the basis of analysis of ***child ostis-systems*** operation and evolution building with the help of this metasystem. But it requires the organization of effective interaction between metasystem ***IMS.ostis*** and the child *ostis-systems* generated by it.
3) Rapid evolution of *OSTIS Technology* requires coordination's automation of versions of various ***ostis-systems components*** with current version of *OSTIS Technology* and, in particular, automation of outdated reusable components replacement with current versions of these components in child *ostis-systems*.

XV. Conclusion

Stress that transition from traditional computer systems is the final stage in the evolution of computer systems in the direction of increasing their flexibility, learnability, providing their unlimited (universal) learnability and logical-semantic compatibility.

Learnability of *intelligent systems*, being an important characteristic of *intelligent systems*, creates good background for significant expansion of their life cycle and ensuring high level of ***competitiveness*** in comparison with traditional computer systems.

But to ensure competitiveness of *intelligent systems*, in addition to their learnability, a well-thought-out ***methodology for their training*** (improvement) is needed. ***Management of project aimed at training intelligent system*** has serious features (primarily due to the specifics of object of training). Especially when this is about open projects.

In addition, one of the subjects of *intelligent system* training project can be intelligent metasystem *IMS.ostis*. But for this it is necessary to develop principles of interaction of *intelligent systems* (in the transition to collectives of *intelligent systems*). At the same time, if high rate of *IMS.ostis* evolution is ensured, then the result is high competitiveness not only of the technology, but also of *intelligent systems* developed on its basis.

High level of *ostis-system learnability* is the basis for ensuring high rate of *OSTIS Technology* evolution, as this technology is also implemented as *ostis-system* (*IMS.ostis*). It is essential that high rate of *OSTIS Technology* evolution is factor in accelerating (!) rate of various ostis systems evolution.

The most important direction of evolution of *OSTIS technology* is expansion and improvement of structure of ***library of reusable components*** of *ostis-systems* that provides essential decrease in labor input of development of ostis-systems.

Competitive technology for development of *intelligent systems* should have high rate of unlimited evolution (improvement, training) and should be oriented towards development of *intelligent systems* that have high level of learnability. But for this, ***general theory of compatible (joint used) knowledge and problem-solving skills***, as well as ***general theory of unlimited trained intelligent systems***, should be developed.

High rate of *OSTIS Technology* evolution, open nature of this technology, and open nature of participation in its development (within open-source project *IMS.ostis*) ensure high competitiveness of *OSTIS Technology.*

Acknowledgment

The work is executed at support of the grant BRFFR-RFBR No. Ф16Р-101 "Development of intelligent learning systems, and synergetic training organizations based on open semantic technologies, ontology engineering and patterns of understanding", and No. Ф16Р-102 "Formalization of temporal reasoning in intelligent systems".

References

[1] (2017, Dec.) The IMS.OSTIS website. [Online]. Available: http://www.ims.ostis.net

[2] E. F. Skorokhod'ko, *Semanticheskie seti i avtomaticheskaya obrabotka teksta [Semantic networks and automatic text processing]*. Kiev: Nauk. dumka, 1983, (in Russian).

[3] L. S. Bolotova, *Sistemy iskusstvennogo intellekta: modeli i tekhnologii, osnovannye na znaniyakh: uchebnik [Systems of artificial intelligence: the models and technologies based on knowledge: textbook]*. M.: Finansy i statistika, 2012, (in Russian).

[4] G. S. Osipov, *Metody iskusstvennogo intellekta [Methods of artificial intelligence]*. M.: FIZMATLIT, 2015, (in Russian).

[5] V. V. Golenkov, N. A. Gulyakina, and T. A. Gavrilova, "Unifitsirovannyi sposob abstraktnogo kodirovaniya semanticheskikh setei [unified way of abstract coding of semantic networks]," in *X mezhdunarodnaya nauchnaya konferentsiya imeni T.A.Taran «Intellektual'nyi analiz informatsii IAI-2010» [Intellectual information analysis]*, Kiev, May 2010, p. 46–52, (in Russian).

[6] V. V. Golenkov and N. A. Gulyakina, "Strukturizatsiya smyslovogo prostranstva [structurization of semantic space]," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, Minsk, Feb. 2014, p. 65—78, (in Russian).

[7] V. B. Tarasov, *Ot mnogoagentnykh sistem k intellektual'nym organizatsiyam [From multi-agent systems to intelligent organizations]*. M.: Editorial URSS, 2002, (in Russian).

[8] S. I. Nikulenkov, *Tulup'ev A.L. Samoobuchayushchiesya sistemy [Self-learning systems]*. M.: MTsNMO, 2009, (in Russian).

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (2nd Edition, Complete Draft)*. London: The MII Press, 2017.

[10] A. P. Eremeev and A. A. Kozhukhov, "Implementation of reinforcement learning tools for real-time intelligent decision support systems," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, Minsk, Feb. 2017, pp. 183–186.

[11] P. Flakh, *Mashinnoe obuchenie. Nauk i iskusstvo poctroeniya algoritmov, kotorye izvlekayut znaniya iz dannykh [Machine learning. The sciences and the art of constructing algorithms that extract knowledge from data]*. M.: LVR Press, 2015, (in Russian).

[12] A. A. Barsegyan, M. S. Kupriyanov, V. V. Stepanenko, and I. I. Kholod, *Tekhnologii analiza dannykh: Data Mining. Text Mining, OLAP [Data analysis technologies: Data Mining. Text Mining, OLAP]*. Spb.: BKhV-Peterburg, 2007, (in Russian).

ОТ ОБУЧЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ К ОБУЧЕНИЮ СРЕДСТВ ИХ РАЗРАБОТКИ

Голенков В.В., Гулякина Н.А.,
Гракова Н.В., Давыденко И.Т., Никуленко В.Ю.
Белорусский государственный университет
информатики и радиоэлектроники
Минск, Беларусь
Еремеев А.П.
Национальный исследовательский университет
«МЭИ», Москва, Россия
Тарасов В.Б.
Московский государственный технический
университет им. Н.Э.Баумана, Москва, Россия

В статье рассматривается подход к реализации перехода от обучения интеллектуальных систем, основанных на знаниях, к обучению средств их разработки. При этом архитектура такой интеллектуальной системы рассматривается как основа обеспечения её гибкости и обучаемости. Кроме того, в статье рассматриваются направления обучения и самообучения интеллектуальных систем, а также их способность приобретать знания и навыки из различных источников. Даётся обоснование применения технологии OSTIS для разработки интеллектуальных систем, основанных на знаниях.