Methods and tools for ensuring compatibility of computer systems

Vladimir Golenkov, Natalia Guliakina, Irina Davydenko
Belarussian State University
Informatics and Radioelectronics
Minsk, Belarus
golen@bsuir.by, guliakina@bsuir.by, davydenko@bsuir.by

Aleksandr Eremeev
National Research University "MEI"
Moscow, Russia
eremeev@appmat.ru

***Abstract*—The paper discusses the main current problems in the modern computer systems development, in particular – the problem of ensuring information compatibility of computer systems. An approach to their solution, based on the use of the Open Semantic Technology for Intelligent System Design (OSTIS), is proposed.**

***Keywords*—semantic computer system, semantic technology, hybrid systems, computer systems compatibility, OSTIS Technology, SC-code, ontology**

I. Introduction

Until now, traditional information technologies and artificial intelligence technologies have evolved **independently of each other**.

Now is the time for **fundamental rethinking** of the experience of using and evolving traditional information technologies and their **integration** with artificial intelligence technologies. This is necessary to eliminate a number of shortcomings of modern information technologies.

The experience of using computer systems to automate various types of human activity shows that automation of disorder leads to even more confusion, and illiterate automation is worse than its absence. Moreover, if automation requires the use of methods and artificial intelligence, the consequences of illiterate automation can be even more devastating.

This means that before proceeding with the automation of any activity (and, especially, with the use of artificial intelligence), it is necessary to build a qualitative formal model of this activity (that is, a sufficiently detailed holistic description of it, but without excesses).

In our report at the conference OSTIS-2018 [1] the key property of intelligent systems was considered - their **learnability**, as well as those properties of intelligent systems that provide a high level of learnability (flexibility, stratification, reflexivity).

In this paper, the currently key problem of the development of information technologies in general and of artificial intelligence technologies in particular, the **problem of ensuring information compatibility** of computer systems, including intelligent systems, will be considered.

The urgency of solving this problem is due to the fact that:

- informational compatibility of computer systems will significantly **increase the level of their learnability** due to more effective perception of experience (knowledge and skills) from other computer systems;
- it will be possible to significantly **expand the diversity** of the knowledge and skills used in the computer system without the need to develop special tools for their coordination. It also increases the level of learnability of computer systems and allows you to move to **hybrid, synergistic** computer systems;
- it will be possible to create **collectives of computer systems**, using universal principles of the organization of interaction between computer systems at the meaningful level;
- It will be possible not only to develop compatible computer systems, but also to automate the process of permanent **support of computer systems compatibility**. The need for this support is due to the fact that the compatibility of computer systems during their operation and evolution may be violated. Consequently, there must be tools which will permanently restore the compatibility of computer systems in the conditions of their permanent change;
- it will be possible to automate the process of permanent support (restoration) of information **compatibility** of computer systems not only with other computer systems, but also **with their users**;
- it will be possible to significantly reduce the development time of new computer systems using the permanently expanding **library of reusable computer system components**, which have different levels of complexity (up to typical embedded subsystems) and different types (typical embedded knowledge, for example, ontologies, widely used skills, in particular, programs, interface subsystems, providing messaging with external subjects in a

given external language).

Lets consider the problems of information technology development:

- in the field of traditional computer systems;
- in the field of intelligent systems;
- in the field of informatization of scientific and technical activities.

II. State and problems of traditional information technologies

The current state of traditional information technologies in general can be described as:

- illusion of well-being;
- illusion of omnipotence of financial resources in solving complex technical problems;
- "Babel" of various technical solutions, the compatibility of which no one seriously thinks about;
- lack of an integrated systems approach to automating complex types of project activities;
- lack of awareness that the shortcomings of modern information technologies are of a fundamental, systemic nature.

The shortcomings of modern information technologies include:

1) Diversity of syntactic forms of presentation of the same information, i.e. variety of semantically equivalent forms (languages) of representation (coding) of the processed information (knowledge) in the memory of computer systems. The lack of unification of the representation of various types of knowledge in the memory of modern computer systems leads:
   - to the variety of semantically equivalent models for problems solving (both procedural and non-procedural - functional, logical, etc.), i.e. to duplication of information processing models that differ not in the essence of the methods of problems solving, but in the form of presentation of the processed information and the form of representation of methods (skills) of solving of various problems classes;
   - to the duplication of semantically equivalent information components of computer systems;
   - to the variety of forms for the technical implementation of each model used for problems solving;
   - to the semantic incompatibility of computer systems and, consequently, to the high complexity of their integration into systems of a higher level of hierarchy, which requires additional efforts to translate (convert) information shared by different integrable systems and, therefore, significantly limits the effectiveness of joint problem solving by a team of interacting computer systems. The complexity of the integration process can be significantly reduced by transition of the integrated computer systems to some uniform form, since in this case the integration can be carried out in a universal and automated way;
   - to a significant decrease in the effectiveness of the use of the method of computer systems component design based on libraries of reusable components (especially when it comes to "large" components, in particular, typical subsystems) [2].
2) Insufficiently high degree of learnability of modern computer systems during their operation, resulting in a high complexity of their maintenance and improvement, as well as their insufficiently long life cycle.
3) The lack of opportunity for experts to really influence on the quality of the developed computer systems. The experience of complex computer systems development shows that the mediation of programmers between experts and projected computer systems substantially distorts the contribution of experts. When developing next-generation computer systems, it is not programmers who should dominate, but experts who are able to accurately state their knowledge.
4) The lack of semantic (sense) unification of the interface activity of users of computer systems, which, together with the variety of forms for implementing user interfaces, leads to serious overhead costs for learning of user interfaces of new computer systems.
5) Computer system documentation is not an important component of the computer system itself, determining the quality of operation of this system, resulting in an insufficiently high efficiency of computer system operation due to incomplete and inefficient use of the capabilities of the computer system being operated.

To overcome these shortcomings is possible only through a fundamental rethinking of the architecture and principles of the organization of complex computer systems. The basis of this rethinking is the elimination of the diversity of forms of representation (coding) of information in the memory of computer systems.

The result of this rethinking should be a new stage in the development of information technology.

Overcoming the shortcomings of modern computer systems involves:

- unification of the information processed;
- functional unification (unification of information processing principles).

III. Problems of artificial intelligence technologies development

Expansion of computer systems applications leads to the expansion of the variety of automated activities – management of various types of enterprises, management of organizations, management of complex technical systems, multisensory integration and primary analysis of non-verbal information, recognition, design of artificial objects of various types, design of business process systems aimed at reproduction of the designed artificial objects, communication with users (on natural languages in text and speech form, using the means of cognitive graphics), user learning, comprehensive information services for users.

In turn, the expansion of the variety of automated activities leads to the expansion of the variety of types of problems solved, types of methods and tools for problems solving, types of information used (types of knowledge).

For example, increasing the level of automation of various enterprises leads to a knowledge-oriented organization of their activities, and in the future – to a knowledge-oriented economy. This means that knowledge management tools become the basis of enterprise automation.

From this, in turn, it follows that in perspective enterprise management systems it is necessary to move from databases that provide a presentation of fairly simple (factographic) types of knowledge to knowledge bases, which may include knowledge of the most diverse types.

A. The evolution of computer systems

Thus, the expansion of the field of application of computer systems requires a transition from traditional computer systems to systems focused on processing a wide variety of structured information, as well as on solving more and more complex problems. Consequently, the transition from traditional computer systems to intelligent systems is inevitable. Moreover, this transition has long been happening. This is confirmed by such directions of evolution of computer systems as:

- the transition from the dominance of programs to the dominance of the processed information, i.e., data-driven computer systems;
- from semi-structured data to structured data and data independent of the programs that process this data, i.e., to databases;
- from data to knowledge by expanding semantic types of processed information, and further to computer systems, managed by structured knowledge, and to computer systems, managed by knowledge bases;
- transition from non-context problem solving, the initial data for which are a priori exactly specified, to problem solving with the active use of the context of these problems, i.e. knowledge of the subject domain in which the task is being solved;
- transition from procedural low-level programming languages to high-level procedural programming languages, and to non-procedural programming languages (functional, logical);
- transition from sequential to parallel programs;
- transition from synchronous information processing to asynchronous;
- transition from programs to calculations, to "soft" calculations (fuzzy logic, genetic algorithms, artificial neural networks);
- transition from data-oriented programs, where the data structuring is determined by the corresponding programs, to programs oriented to database processing and further knowledge bases processing;
- transition from address memory to associative memory;
- transition from linear memory to non-linear (reconstructable, reconfigurable, graph-dynamic) memory, in which information processing is reduced not only to a change in the state of the elements in the memory, but also to a change in the configuration of the connections between them;
- transition from traditional computer systems to computer systems capable of solving a wide variety of complex (difficult to formalize) problems and, including intelligent problems, to computer systems with a hybrid well-structured high-quality knowledge base, with a hybrid problem solver, with a hybrid (multimodal) interface (both verbal and non-verbal);
- transition from non-learnable computer systems to learnable.

Consequently, the intellectualization of computer systems is the natural direction of their evolution.

The modern most actively developed areas of development of intelligent systems include:

- knowledge management and ontological engineering [3], Semantic Web [4];
- formal logic (strict, fuzzy, deductive, inductive, abductive, descriptive, temporal, spatial, etc.);
- artificial neural networks, Bayesian networks, genetic algorithms (Machine learning in the narrow sense);
- computer linguistics (NLP), semantic analysis of natural language texts;
- speech processing, semantic analysis of voice messages;
- image processing - technical vision, semantic image analysis;
- multi-agent systems, collectives of intelligent systems [5], [6], [7];
- hybrid intelligent systems, synergistic intelligent systems [8].

B. The current state of artificial intelligence technology

Despite the presence of serious **scientific results** in the field of artificial intelligence, the rate of **evolution of the intelligent systems market** is not so impressive.

There are several reasons for this:

- there is a big gap between scientific research in the field of artificial intelligence and the creation of high-quality technologies for the development of intelligent systems. Scientific research in the field of artificial intelligence is mainly focused on the development of new methods for solving intelligent problems;
- these researches are scattered and not aware of the need for their integration and the creation of a general formal theory of intelligent systems, i.e. there is a "babel" of various models, methods and tools used in artificial intelligence in the absence of awareness of the problem of ensuring their compatibility. Without solving this problem, neither the general theory of intelligent systems nor, therefore, the complex technology of intelligent systems development available to engineers and **experts** can be created;
- the specified integration of models and methods of artificial intelligence is very complex, since it is interdisciplinary in nature;
- intelligent systems as objects of design have a significantly higher level of complexity compared to all the technical systems with which humanity have had a deal;
- as a consequence of the above, there is a big gap between scientific research and engineering practice in this area. This gap can be filled only by creating an evolving technology of intelligent systems development, the creation of which is carried out through active cooperation of scientists and engineers;
- the quality of development of applied intelligent systems depends to a large extent on the mutual understanding of experts and knowledge engineers. Knowledge engineers, not knowing the intricacies of the applied area, can introduce serious errors into the developed knowledge bases. The mediation of knowledge engineers between experts and the knowledge base being developed significantly reduces the quality of the developed intelligent systems. To solve this problem, it is necessary that the knowledge representation language in the knowledge base be convenient not only to the intelligent system and knowledge engineers, **but also to experts**.

The current state of artificial intelligence technology can be described as follows:

- There is a large set of proprietary artificial intelligence technologies with appropriate tools, but there is no general theory of intelligent systems and, as a result, there is no overall integrated technology for intelligent systems design (see Artificial General Intelligence conference [9]);
- Compatibility of particular technologies of artificial intelligence is practically not implemented, and moreover, there is no awareness of such a need.

The development of artificial intelligence technologies is significantly hampered by the following socio-methodological circumstances:

- High social interest in the results of work in the field of artificial intelligence and the great complexity of this science gives rise to superficiality and untidiness in the development and advertising of various applications. Serious science is mixed with irresponsible marketing, conceptual and terminological negligence and illiteracy, throwing in new absolutely unnecessary effective terms that confuse the essence of the matter, but create the illusion of fundamental novelty.
- The interdisciplinary nature of research in the field of artificial intelligence significantly complicates these researches, because work at the junctions of scientific disciplines requires high culture and skills.

C. Directions of development of artificial intelligence technologies

To solve the above problems of the development of artificial intelligence technology:

- Continuing to develop new formal models for intelligent problems solving and to improve existing models (logical, neural network, production), it is necessary to ensure compatibility of these models both among themselves and with traditional models for problems solving that were not included in the number of intelligent problems. In other words, we are talking about the development of principles for the organization of hybrid intelligent systems that provide solutions to **complex problems** that require joint in unpredictable combinations of the most diverse types of knowledge and the most diverse models for problems solving.
- A transition is needed from the eclectic construction of complex intelligent systems using various types of knowledge and various types of problem solving models to their deep integration, when the same representation models and knowledge processing models are implemented in different systems and subsystems in the same way.
- It is necessary to reduce the distance between the modern level of the theory of intelligent systems and the practice of their development.
- It is necessary to significantly increase the level of consistency of actions of persons involved in the process of continuous improvement of knowledge bases.

- It is necessary that the systems themselves, and not just their developers, actively participate in solving this compatibility problem of intelligent systems. Systems themselves must take care of maintaining their compatibility with other systems in the context of the active change of these systems through the mechanism of automated coordination of the concepts used between intelligent systems.

IV. Problems of development of methods and tools of scientific activity informatization

It is obvious that the highest form of information activity is scientific activity and, therefore, the highest level of development of computer systems are the systems that are directly and actively involved in this activity. Scientific activity is aimed at improving the quality of our knowledge about the world around us and, therefore, is associated with the analysis, processing and systematization of this knowledge. It is obvious that if computer systems aimed at automating scientific activities understand the scientific knowledge they process and, therefore, will become not passive performers, but scientific partners who are able to independently analyze, systematize scientific knowledge and use them in various problems solving then the level of automation of scientific activity will be significantly increased.

The most important restraining factors of scientific and technological progress at present are:

- diversity ("babel") of both natural and formal languages used to present the results of scientific and technical research;
- binding scientific and technical texts to natural languages (monographs, reports, articles);
- fundamental contradiction between the principles of the evolution of natural languages as the main means of communication and the requirements for scientific and technical languages.

To solve these problems we need:

- to build a strict formal system of scientific and technical languages;
- to build a clear connection between scientific and technical and natural languages;
- to ensure the design of scientific and technical texts in compatible formal languages that are understandable and convenient for both people and computer systems;
- to provide support for the evolution of this multi-language complex.

The most important direction of increasing the effectiveness of scientific and technical activities (and, in particular, increasing the rate of scientific and technological development) is the transition from the traditional version of the results of this activity (in the form of reports, articles, monographs, reference books) to the presentation of scientific and technical information in the form of an encyclopedic systems of interconnected knowledge bases on various scientific and technical disciplines. The formal result of any scientific discipline should be a knowledge base reflecting the current state of this discipline. For applied scientific disciplines, an additional result should be a computer-aided design system for designing artificial systems of the corresponding class that is accessible to engineers.

The idea of the difficulties of such a transition is greatly exaggerated, since modern tools of knowledge engineering are ready for the implementation of such projects. This is prevented by:

- fear of the new, unusual;
- need to revise the organization of scientific and technical activities.

But the perspective is a transition to a qualitatively new level of culture of scientific and technological progress.

The social significance of this transition is as follows:

- The rate of evolution of scientific knowledge will significantly increase due to the fact that the obtained scientific knowledge is presented in a form convenient for both people and computer systems, as well as by automating their integration, analysis, structuring and coordination of various points of view.
- The efficiency of the use of scientific knowledge in the developed computer systems will significantly increase, due to the fact that there is no need for the step of formalizing this knowledge to be included in the knowledge bases.
- The possibility of direct participation of students in improving the knowledge that corresponds to the academic disciplines they study will significantly improve the quality of such learning, since promotes individual, active and systematic learning of the educational material.

The main problem of the development of scientific and technical activities and, accordingly, of its informatization is the need for deep **convergence** of various scientific disciplines, as discussed in a number of works [10], [11].

An important problem is also the reduction of time and laboriousness in organizing informational interaction between scientists in the **agreement of points of view**, in the joint implementation of any research, in the joint work on articles or monographs, in reviewing.

It should be remembered that any point of view always has shortcomings (incompleteness, fuzziness, etc.). Therefore, it is methodologically necessary to move from the practice of confronting points of view to the practice of integrating points of view (including those that seem to be alternative, contradictory). Only in the development of complex systems can a synergistic effect be achieved, which is based on compensation for the shortcomings of some points of view by the advantages of others.

This is how the organization of a collective creative process should be arranged. Automating such a process involves fixation of a multiplicity of points of view and managing the process of reconciling these points of view.

V. The proposed approach to solving problems that hinder the further evolution of computer systems and technologies - standardization of information representation and processing models

Analysis of the problems of the evolution of computer systems of different levels of complexity, different levels of learnability and intelligence, of different purposes shows that the curse of the "babel" and, as a result, incompatibility, duplication and subjectivity of coordinated information resources and models of processing them haunts us everywhere:

- and in the development of traditional computer systems;
- and in the development of artificial intelligence technologies;
- and in the development of methods and tools of informatization of scientific and engineering activities.

Considering the problem of ensuring the compatibility of information resources and models of their processing, we should talk about various aspects of solving this problem:

- about ensuring compatibility between various components of computer systems, as well as between complete computer systems that are part of computer systems teams;
- about compatibility, i.e. high level of mutual understanding between different computer systems and their users;
- about interdisciplinary compatibility, i.e. convergence of different areas of knowledge;
- about the methods and means of continuous monitoring and restoring compatibility in the conditions of intensive evolution of computer systems and their users, which often violates the achieved compatibility (consistency) and requires additional efforts to restore it.

A. Directions of the evolution of computer systems

In the evolution of computer systems can be distinguished two general directions.

First Direction is

- **expansion of the set and variety of problems** solved by a computer system;
- increase **the complexity of these problems** down to difficultly formalized (difficultly solvable) problems, intelligent problems solved in the conditions of incompleteness, inaccuracy, vagueness, etc .;
- increase **quality of problem solving** either by more efficient use of known models for problems solving (for example, by developing better algorithms), or by using fundamentally new models for problems solving;
- extension the **variety of information (knowledge) used**;
- extension the **variety of used problems solving models**.

Obviously, the expansion of the set of solved problems in the conditions of a large but always finite memory of a computer system makes the transition from particular methods and models for solving problems to their generalizations (or, as D.A. Pospelov noted, from bundle of "keys" to a set of "lockpicks").

It is also obvious that the variety of types of problems solved by computer systems, the variety of models used for problems solving leads:

- to integrated information resources;
- to integrated problem solvers;
- to integrated computer systems;
- to computer system teams.

The problem here is not the integration itself, but its quality. Integration may be **eclectic** if the compatibility of the integrable components is not ensured, and in the case of such compatibility integration may lead to a new quality, to an additional expansion of the set of solved problems. This will mean a transition from eclecticism to hybridity, synergy.

The second general direction of the evolution of computer systems is the increase in their **learnability** and, as a result, the rate of their evolution.

Learnability of computer system is determined by:

- **labor intensity** and the pace of acquisition (expansion) and improvement of actively used knowledge and skills;
- **level of restrictions** imposed on the type of acquired and used knowledge and skills (in fact, these are restrictions on the set of all those problems that can in principle be solved by a given computer system).

In turn, the **labor intensity and rate of expansion and improvement** of the knowledge and skills of a computer system is determined by:

- **flexibility** – the variety and laboriousness of possible changes made to the system in the process of replenishing the system with new knowledge and skills and improving already acquired knowledge and skills;
- **stratification** – a clear separation of the system into hierarchy levels that are rather independent of each other, i.e. the possibility of localizing fragments of a computer system, without going beyond of which it is apriori possible to analyze the effects of certain changes in the system;

- **reflexivity** — the ability to analyze one's own state and one's activity;
- **hybridity** - the ability to acquire and use a wide (and ideally unlimited) variety of knowledge and skills;
- **level of self-learnability** - the level of activity, independence, purposefulness in the process of their learning, i.e. the level of ability to learn without a teacher, the level of automation of the acquisition of new knowledge and skills, as well as the improvement of already acquired knowledge and skills;
- **compatibility** – integration complexity;
- **the ability to continuously monitor and maintain its compatibility** with other computer systems and with its users in the context of the intensive evolution of these computer systems and their users.

Compatibility (integration complexity) of computer systems can be considered in two aspects:

- in the aspect of **deep integration** of computer systems, which involves the transformation of several computer systems into one consistent computer system by combining information and functional resources of integrable computer systems;
- in the aspect of converting several computer systems into **team of interacting computer systems**, capable of jointly corporate solving of complex problems.

Compatibility (complexity of integration) of computer systems is determined by:

- compatibility of various types of information (knowledge) stored in the memory of a computer system;
- compatibility of various problem solving models;
- compatibility of embedded (including typical) subsystems that are part of computer systems;
- compatibility of external information entering the computer system with information stored in the memory of a computer system (the laboriousness of understanding external information - translation, immersion, concepts aligning);
- communication (including semantic) compatibility with users and with other computer systems.

The most important form of computer system learning is the acquisition of new knowledge and skills in the "ready" form, i.e. in the form of some sign structures entered into the memory of a computer system, since the acquisition of knowledge and skills from external reliable sources requires significantly less time compared to their acquisition on its own, based on its own experience and its own mistakes. But in order for this form of learning to be effective, it is necessary to simplify and formalize as much as possible the mechanism (procedure) of immersing new knowledge in the memory of a computer system.

To solve this problem, the creation of a convenient method for coding various types of information in the memory of a computer system is of key importance.

Since the main channel for learning computer systems is the acquisition of knowledge and skills from other subjects – from other computer systems and from users (from developers-teachers and from end users). Consequently, the level of learnability of computer systems is also determined by the level of its compatibility with these external subjects themselves, with the knowledge and skills acquired by it, i.e. the degree of how the computer system, together with the subjects with which it exchanges information, solves the problem of the "babel".

B. The essence of the proposed approach

The essence of our approach to solving the problems of the evolution of computer systems is, firstly, to combine all the above directions of the evolution of computer systems (both general directions and particular ones) and, secondly, to interpret the problem of providing **compatibility** types of knowledge, various models for solving problems, various computer systems as the **key problem** of the evolution of computer systems, whose solution will greatly simplify the solution of many other problems.

For example, without ensuring the compatibility of information resources used in different computer systems, as well as information resources representing knowledge of various semantic types, it is impossible:

- neither to create **computer system teams** capable of coordinating their actions while cooperatively solving complex tasks;
- neither to create **hybrid computer systems** that are capable of using various combinations of different types of knowledge and different models of problems solving when solving complex problems;
- neither to use the **component design methodology** of computer systems **at all levels** of the hierarchy of designed systems.

What kind of informational compatibility and mutual understanding (including between specialists) can we talk about in the presence of terrifying conceptual and terminological messiness, terminological pseudo-creativity, including, in the field of computer science.

Speaking about **compatibility** of computer systems and their components, as well as compatibility of computer systems with users, we should note the ambiguity of the interpretation of the term "compatibility". In this regard, it should be distinguished:

- compatibility as one of the learning factors, like **ability** to quickly increase the level of consistency (integration, mutual understanding). Compare learning as **ability** to rapidly expand knowledge and

skills, but not characterizing the volume and quality of acquired knowledge and skills;
- compatibility as a characteristic of the achieved level of consistency (integration, mutual understanding).

Similarly, the intelligence of a computer system, on the one hand, can be interpreted as **level** (volume and quality) of acquired knowledge and skills, and on the other hand, as **ability** to rapidly expand and improve knowledge and skills, i.e. as **speed** enhance knowledge and skills.

In addition, one should speak not only about the **ability** to rapidly increase the level of consistency and not only about the level of consistency achieved, but also about the **process** of increasing the level of consistency and, above all, about the permanent restoration process (support maintaining the level of consistency achieved, since during the evolution of computer systems and their users (i.e., in the course of expanding and improving the quality of their knowledge and skills), their consistency may decrease.

C. Semantic unification of computer systems

The main factor in ensuring the compatibility of various types of knowledge, various models of problem solving and various computer systems in general is

- unification (standardization) of information representation in the memory of computer systems;
- unification of the principles of organization of information processing in the memory of computer systems.

The unification of the information representation used in computer systems implies:

- syntactic unification of the information used - the unification of the form of representation (coding) of this information. It should be distinguished:
  - •• coding information in the memory of a computer system (internal presentation of information);
  - •• external presentation of information ensuring the unambiguous interpretation (understanding, interpretation) of this information by different users and different computer systems;
- semantic unification of the information used, which is based on the agreement and exact specification of all (!) used concepts using a hierarchical system of formal ontologies.

It is important to note that competent unification (standardization) should not limit the creative freedom of the developer, but guarantee the **compatibility** of its results with the results of other developers. We also emphasize that the current version of any **standard** is not a dogma, but only a basis for its further improvement.

The goal of a quality standard is not only to ensure the compatibility of technical solutions, but also to minimize duplication (repeating) of such solutions. One of the most important quality criteria of a standard is nothing excess.

standart
= *knowledge of the structure and principles of functioning of artificial systems of the corresponding class*
= *ontology of artificial systems of a certain class*
= *theory of artificial systems of a certain class*

Standards, like other knowledge important to humanity, must be formalized and must be constantly improved using special intelligent computer systems that support the process of standards evolution by reconciling different points of view.

VI. The standard of semantic representation of information in the memory of a computer system

A. Unification of the internal presentation of information in computer systems

The objective guideline for **unification of information representation** in the memory of computer systems and the key to solving many problems of the evolution of computer systems and technologies is **formalization of the sense of the information being presented**.

According to V. V. Martynov [12], «virtually every human thought activity (not only scientific), as many scientists believe, uses an internal semantic code, which is translated from a natural language and from which it is translated into a natural language. The amazing ability of a person to identify a huge variety of structurally different phrases with the same meaning and the ability of **remember the meaning outside of these phrases** convinces us of this.»

We also give the words of I.A. Melchuk [13]:

« The idea was the next – the language should be described as follows: one should be able to write down the meanings of the phrases. Not phrases, but their meanings, which is separate. Plus build a system that builds the meaning of the phrase. This is the area or the turn of research in which the intuition of a capable linguist works best: how to express this meaning in a given language. This is what linguists are taught for ..

The linguistic meaning of a scientific text is not at all what you, reading it, extract from it. This, very roughly speaking, is an invariant of synonymous paraphrases. You can express the same meaning by so many. When you say, you can say in different ways: “Now I pour you wine”, or: “Let, I will offer you wine”, or: “Should we drink a glass of wine?”, - all this has the same meaning. And here you can think of how to record this meaning. Exactly it. Not a phrase, but a meaning. And it is necessary to work from this sense to real phrases. The syntax there is also needed by the way, but it is needed only by the way, it can be neither the final goal, nor the starting point. This is an intermediate case. » [14].

The clarification of the principles of **semantic representation of information** is based, firstly, on a clear contrast between **the internal language of a computer system** used to store information in computer memory, and **external languages of a computer system** used for communication (exchange messages) of a computer system with users and other computer systems (sense representation is used exclusively for the **internal representation** of information in the memory of a computer system), and, secondly, to possibility of simplification of the computer system internal language syntax while providing versatility by excluding from such an internal universal language means providing a communication function language (m. e. messaging).

For example, for the internal language of a computer system, such communication tools of the language as conjunctions, prepositions, dividers, limiters, declensions, conjugations, and others are superfluous.

External languages of a computer system can be both close to its internal language, and very far from it (as, for example, natural languages).

Sense is a **abstract** sign construct belonging to the internal language of a computer system, being the **invariant** of the maximum class of semantically equivalent sign constructions (texts) belonging to different languages and satisfying the following requirements:

- **universality** – the ability to present any information;
- **absense of synonymy signs** (multiple occurrence of characters with the same denotates);
- **absense of duplication of information** in the form of semantically equivalent texts (not to be confused with logical equivalence);
- **absence of homonymous signs** (including pronouns);
- **absence of internal structure of signs** (atomic character of signs);
- **absense of declensions, conjugations** (as a result of the absence of the internal structure of signs);
- **absense of fragments** of a sign construct, which are not **signs** (separators, delimiters, etc.);
- **distinguishing of connection signs**, the components of which can be any signs with which connection signs are associated with syntactically defined incidence relations.

The consequence of these principles of the semantic representation of information in the memory of a computer system is that the entity signs included in the semantic presentation of information are **not names** (terms) and, therefore, are not tied to any natural language and do not depend on subjective term addictions of various authors. This means that from the collective development of the semantic representation of any information resources terminological disputes are excluded.

The consequence of these principles of sense representation of information is also the fact that these principles lead to non-linear sign structures (graph structures), which complicates the implementation of computer system memory, but significantly simplifies its logical organization (in particular, associative access).

The nonlinearity of the sense representation of information is due to the fact that:

- each described entity, i.e. an entity that has a corresponding sign can have an unlimited number of connections with other described entities;
- each described entity in the sense representation has a single sign, because synonymy of signs is prohibited here;
- all connections between the described entities are described (reflected, modeled) by the connections between the signs of these described entities.

The essence of the **universal sense representation of information** can be formulated in the form of the following provisions:

- Sense sign construction is interpreted as a set of signs, which are one-to-one designating different entities (denotations of these signs) and a set of connections between these signs;
- Each connection between signs is interpreted, on the one hand, as a set of signs connected by this connection, and, on the other hand, as a description (reflection, model) of the corresponding connection, which connects the denotations of the specified signs or the denotation of some signs directly to other characters, or these signs themselves. An example of the first type of connection between signs is the connection between signs of material entities, one of which is part of the other. An example of the second type of connection between signs is the connection between the sign of the set of signs and one of the signs belonging to this set, as well as the connection between the sign and the file sign, which is an electronic reflection of the structure of the representation of the specified sign in external sign structures. Examples of the third kind of connection between signs are the connection between synonymous signs;
- The denotats of characters can be (1) not only specific (constant, fixed), but also arbitrary (variables, non-fixed) entities that "run through" various sets of signs (possible values), (2) not only real (material), but also abstract entities (for example, numbers, points of various abstract spaces), (3) not only "external", but also "internal" entities, which are sets of signs that are part of the same sign structure.

The key property of the sense representation of information language is the uniqueness of the information representation in the memory of each computer system, i.e., the absence of semantically equivalent sign constructions belonging to the sense language and stored in one sense memory. At the same time, the logical equivalence of such sign constructions is allowed and used, for example, for a compact representation of some knowledge stored in the sense memory.

However, the logical equivalence of the constructions stored in the memory should not be carried away, because **logically equivalent** sign constructions are representations of the same knowledge, but with the help of **different sets of concepts**. In contrast, **semantically equivalent** sign constructions are the representation of the same knowledge with the help of the same concepts. It is obvious that the variety of possible options for the representation of the same knowledge in the memory of a computer system significantly complicates the problems solving. Therefore, by completely eliminating **semantic equivalence** in semantic memory, it is necessary to strive to minimize **logical equivalence**. For this, a competent construction of a system of used concepts in the form of a hierarchical system of formal ontologies [15] is necessary.

An important step in creating a universal formal method of sense coding of knowledge was developed by V.V. Martynov Universal Semantic Code (USC) [12].

As the **standard** of the universal sense representation of information **in the memory of computer systems** we have proposed **SC-code** (Semantic Computer Code). Unlike USC of V.V. Martynov if, firstly, is non-linear in nature and, secondly, is specifically focused on coding information in the memory of computers of a new generation, focused on the development of semantically compatible intelligent systems and called **semantic associative computers**. Thus, the main leitmotif of the proposed sense presentation of information is the orientation to the formal memory model of a non-Von-Neumann computer designed for the implementation of intelligent systems using the sense representation of information. The features of this representation are as follows:

- associativity;
- all information is enclosed in a connections configuration, i.e. processing information is reduced to the reconfiguration of connections (to graph-dynamic processes);
- transparent semantic interpretability and, as a result, semantic compatibility.

Implicit binding to Von Neumann computers is present in all known knowledge representation models. One example of such a dependency is, for example, the obligatory naming of the objects being described.

B. Syntax of SC-code

The universality of the SC-code allows using it to describe any objects. This object can be any language of communication with users (including natural language), as well as the SC-code itself. The syntax of the SC-code is represented as the corresponding formal ontology. The key concepts of the subject domain that are described (specified) by the mentioned ontology are:

sc-element
= *atomic fragment of the sign construction stored in the memory and belonging to the SC-code*
sc-node
sc-connector
sc-edge
= *non-oriented sc-connector*
sc-arc
= *oriented sc-connector*
base sc-arc
incidence of sc-connector*
incidence of incoming sc-arc*

Within the specified domain, the class of all possible *sc-elements* is the maximum class of studied objects study, the concepts *sc-node*, *sc-connector*, *sc-edge*, *sc-arc*, *base sc-arc* are specially syntactically distinguished subclasses of the maximum class of studied objects, and the concepts *incidence of the sc-connector** and *incident of the incoming sc-arc** are treated as relations defined on the set of studied objects.

The family of all entered classes of studied objects (including the maximum class) is interpreted as ***Alphabet of SC-code***. But, unlike other languages, the classes of syntactically distinguished elementary fragments of SC-code texts may overlap. For example, an sc-element can belong to both the *sc-element* class and the *sc-node* class, and can also belong to the *sc-element* class and *sc-connector*, and the *sc-arc* class, and the *basic sc-arc* class.

This feature of the *Alphabet SC-code* makes it possible to build syntactically correct ***sc-texts*** (texts of the SC-code) in the conditions of incompleteness of our initial knowledge about some *sc-elements*.

Lets consider the set-theoretic ontology of the SC-code syntax:

sc-element
<= *subdividing**:
{
• *sc-node*
• *sc-connector*
}

sc-connector
<= *subdividing**:
{
• *sc-edge*

- *sc-arc*

}

sc-arc
⊃ *base sc-arc*

incidence of sc-connector*
=> *first domain**:
sc-connector
=> *second domain**:
sc-element
⊃ *incidence of incoming sc-arc**
∈ *binary relation*
∈ *oriented relation*
∈ *relation, elements of which there are no multisets*
/*for binary relationships, this means no loops*/

incidence of incoming sc-arc*
=> *first domain**:
sc-arc
=> *second domain**:
sc-element

During the text processing, the following rules for clarifying their syntactic markup are executed:

- if it has become known that *sc-element*, having a *sc-element* label, is *sc-node* or *sc-connector*, then it is assigned a label *sc-node* or *sc-connector*, and the label *sc-element* is deleted;
- if it has become known that *sc-element* with the *sc-connector* label is *sc-edge* or *sc-arc*, then the *sc-edge* or *sc-arc* label is assigned, and the label *sc-connector* is deleted;
- if it has become known that *sc-element* with the *sc-arc* label is *basic sc-arc*, then the label *basic sc-arc* is assigned to it, and the label *sc-arc* is deleted.

Note some syntactic features of the SC-code.

- The texts of the SC-code are **abstract** in the sense that they abstract from the specific variant of their encoding in the memory of the computer system. The coding of texts, in particular, depends on the variant of the technical implementation of the memory of a computer system. For example, the actual implementation is the hardware implementation of an associative non-linear memory in which the structural reconfiguration of the stored information is realized, in which information processing is reduced not to a change in the state of memory elements, but to a change in the configuration of the connections between them.
- The texts of the SC-code are structures of a **graph-like type**. All graph structures studied so far can be easily represented in the SC-code (undirected and oriented graphs, multigraphs, pseudographs, hypergraphs, networks, etc.). But, besides this, in the SC-code, there are representable links between connections, connections between whole structures and much more. The SC-code is actually a **graph language**, whose texts are graph-like structures. Thus, the graph theory with its appropriate extension can become the basis for the description of the syntax of the SC-code.

C. Semantic of SC-code

The simplicity of the ***SC-code*** syntax is determined by the following **semantic** properties of *sc-texts* (character constructions belonging to the SC-code).

- **All** (!) *sc-elements*, that is, elementary (atomic) fragments of *sc-texts*, are signs (symbols) of various described entities. At the same time, each entity described in the text *SC-code* must be represented by its sign;
- There are no signs other than *sc-elements*, *sc-texts* (i.e., there are no signs that include other signs);
- Any entity can be described by *sc-text* and, accordingly, is represented in this *sc-text* by its sign;
- All *syntactically distinguished classes of sc-elements* (i.e. all elements of *Alphabet of SC-code*) have a clear semantic interpretation – are classes of *sc-elements*, each of which denotes an entity that shares the same properties with all other entities, denoted by other *sc-elements* of the same class.

From the formal point of view, the denotational semantics of any sign construction (including *sc-text*) is a correspondence (more precisely, morphism) between the set of all signs included in the sign construction and the set of denotates of these signs (i.e. entities, denoted by these signs), as well as between the set of all semantically significant (semantically interpreted) connections between the signs, and the set of corresponding connections connecting either the denotations of all of the specified signs, or the denotations of some of the specified signs directly with the rest of the signs from the mentioned signs themselves.

Consider the denotational semantics of *sc-elements* belonging to different *syntactically distinguished classes of sc-elements*, i.e. having different syntax labels.

If a *sc-element* is labeled as ***sc-element***, then it can denote any described **entity**.

If *sc-element* has a label of ***sc-connector***, which is incident to *sc-element* ***ei*** and to *sc-element* ***ej***, then, on the one hand, it a sign of ***pair*** {***ei***, ***ej***}, and, on the other hand, is a model (reflection, description) of the connection either between the denotate of *sc-element* ***ei*** and the denotate of *sc-element* ***ej***, either between the denotate of *sc-element* ***ei*** and the *sc-element* ***ej*** itself, either between the denotate of *sc-element* ***ej*** and *sc-element* ***ei*** itself.

If the *sc-element* has the label ***sc-node***, then it denotes **an entity that is not a pair**.

If the *sc-element* has a label of ***sc-edge***, which is incident to the sc-element ***ei*** and *sc-element* ***ej***, then it is, on the one hand, the ***undirected pair*** {***ei***, ***ei***}, and on the other hand, is a model (reflection, description) of the connection either between the denotate of *sc-element* ***ei*** and the denotate of *sc-element* ***ej***, either between the denotate of *sc-element* ***ei*** and the *sc-element* ***ej*** itself, either between the denotate of *sc-element* ***ej*** and *sc-element* ***ei*** itself.

If *sc-element* has a label of **sc-arc**, which leaves *sc-element* ***ei*** and enters *sc-element* ***ej***, then, on the one hand, it is a sign ***oriented pair <ei, ej>***, and on the other hand, is a model (reflection, description) of the connection either between the denotate of *sc-element* ***ei*** and the denotate of *sc-element* ***ej***, either between the denotate of *sc-element* ***ei*** and the *sc-element* ***ej*** itself, either between the denotate of *sc-element* ***ej*** and *sc-element* ***ei*** itself.

If *sc-element* has a label of ***base sc-arc***, which outgoes *sc-element* ***ei*** and enters *sc-element* ***ej***, then, on the one hand, is a sign of oriented ***constant positive permanent pair of membership <ei, ej>***, and, on the other hand, is a model (reflection, description) of the connection between the set, which is denoted by *sc-element* ***ei***, and *sc-element* ***ej***, which is one of the elements of the specified set.

We now turn to the consideration of the denotational semantics of the ***incidence of sc-connectors***. Recall that each *sc-connector* is semantically interpreted as a sign of ***pair*** of *sc-elements* incident to this *sc-connector*. Accordingly, each *pair of incidence of the sc-connector*, not being a sc-element, is semantically interpreted as a model (reflection, description) of the connection between the *pair* of sc-elements, denoted by this *sc- connector*, and one of the two elements of this *pair*. At the same time, the membership of the specified *sc-element* within the specified *pair* may have:

- ***constant*** or ***variable*** character depending on the constancy or variability of the specified *sc-connector*;
- ***stationary*** (permanent) or ***non-stationary*** (situational) character depending on the stationarity or non-stationarity of the specified *sc-connector*.

The denotational semantics of the ***incidency of incoming sc-arcs*** is given in a similar way. Each such incidence pair is considered as a connection model between the ***oriented pair***, denoted by *sc-arc* and the second component of this pair (i.e. *sc-element*, in which *sc-arc* ingoes). And similarly to the *incidence of sc-connectors* pairs *incidence of incoming sc-arcs* can have *constant* and *variable* character, as well as *stationary* and *non-stationary* in depending on the nature of the corresponding *sc-arc*.

The formal description of the denotational semantics of *SC-code* by means of the *SC-code* itself is carried out in the form of a hierarchical system of to-level ***formal ontologies***, presented in the form of *SC-code*. In the knowledge base of ***Metasystems IMS.ostis*** all these ontologies are presented [16]. We list some of them.

Consider the ***Ontology of entities***, within which the following concepts are considered:

entity
= *sc-element*
<= *subdividing**:
    {
    • *sc-constant*
    • *sc-variable*
        = *sign of an arbitrary entity from a set of possible values*
    }
<= *subdividing**:
    {
    • *stationary entity*
        = *permanent entity*
    • *temporal entity*
        = *non-stationary entity*
        = *time-varying entity*
        ⊃ *temporary entity*
            = *temporarily existing entity*
    }
<= *subdividing**:
    {
    • *material entity*
    • *terminal abstract entity*
    • *file*
        = *primary (in perception) or final (in display) electronic image of the external information structure*
    • *set*
        = *set of sc-elements*
        <= *subdividing**:
            {
            • *connection*
            • *structure*
            • *class*
                <= *subdividing**:
                    {
                    • *terminal entity class*
                    • *relation*
                        = *class of connections*
                    • *class of classes*
                        ⊃ *parameter*
                    • *class of structures*
                    }
            }
    }

connection
= *tuple*
<= *subdividing**:
    {

• *pair*
= *binary connection*
⊃ *sc-connector*
/*some pairs of sc-elements in some periods of time may not be syntactically designed as connectors, but such a transformation necessarily occurs*/
• *non-binary connection*
}
<= *subdividing**:
{
• *non-oriented connection*
⊃ *non-oriented pair*
• *oriented connection*
⊃ *oriented pair*
}
<= *subdividing**:
{
• *constant connection*
= *(connection ∩ sc-constant)*
• *variable connection*
= *(connection ∩ sc-variable)*
⊃ *sc-variable values of which are constant connections*
⊃ *sc-variable values of which are variable connections*
}

pair
= *designation of a two-power set of sc-elements*
<= *subdividing**:
{
• *non-oriented pair*
⊃ *sc-edge*
• *oriented pair*
⊃ *sc-arc*
⊃ *pair of membership*
}
<= *subdividing**:
{
• *loop pair*
= *looped pair*
= *pair, incident sc-elements of which coincide*
= *couple being a multiset*
• *non-loop pair*

pair of membership
= *connection describing the nature of the membership of some sc-element in some set*
<= *subdividing**:
{
• *pair of constant membership*
= *(pair of membership ∩ sc-constant)*
• *pair of variable membership*
= *(pair of membership ∩ sc-variable)*
}
<= *subdividing**:
{
• *pair of permanent membership*
= *(pair of membership ∩ stationary entity)*
= *pair of stationary membership*
• *pair of temporary membership*
= *(pair of membership ∩ temporary entity)*
= *pair of situational membership*
}
<= *subdividing**:
{
• *pair of positive membership*
= *pair of real membership*
• *pair of fuzzy membership*
• *pair of negative membership*
= *pair of nonexistent membership*
}
⊃ *pair of constant positive permanent membership*
= *(pair of positive membership ∩ sc-constant ∩ stationary entity)*
⊃ *base sc-arc*

The following ontologies clarify (detail) the concepts introduced in ***Ontology of entitites***.

The ***Ontology of sets*** clarifies the concept of *set* of sc-elements, considers various classes of sets (finite, infinite, countable, continual, multisets, sets without multiple elements), different properties (characteristics) and relations, given on sets (the power of sets, inclusion, union, subdividing, intersection, etc.).

The ***Ontology of relations*** deals with such concepts as *binary relation, unary relation, ternary relation, class of connections of equal power, class of connections of different power, arity of a relation, oriented, undirected relation, role relation, relation attributes*, relation domain*, relation domain given attribute*, function*, etc.

For the *Ontology of relations*, a lower level ontology is introduced – ***Ontology of binary relations and correspondences***, which inherits all the properties of relations described in *Ontology of relations*, clarifies the concept of *binary relation* and considers such concepts as *transitive relation, symmetric relation, reflexive relation, equivalence relation, isomorphism, homomorphism*, etc.

Next are introduced

- ***Ontology of parameters and dimensions***
- ***Ontology of structures***
  - ***Ontology of subject domains***
  - ***Ontology of specifications***
  - ***Ontology of knowledge bases***
- ***Ontology of variables and logical formulas***
- ***Ontology of temporal entities***, which deals with such concepts as *non-stationary parameter* (state), *process, action, situation, sequence in time*, temporal decomposition** and other
  - ***Ontology of actions***

- ***Ontology of files and external information structures***

Some of the *ontologies*, presented in *SC-code*, have "general educational" character. This means that for quality of mutual understanding between any subjects (both users and computer systems), i.e. for their qualitative semantic compatibility, all these "general educational" ontologies, and in a coordinated, unified form, should be known by all of subjects (!). Otherwise, there will be no mutual understanding.

The list of *ontologies* can be continued. All *ontologies* are permanently changing (specified, improved). The most important criterion of the quality of the hierarchical system of *ontologies* is the stratification of methods for problems solving corresponding to different *ontologies* – for each problem to be solved, it is desirable to apriori know within which *ontology* it can be solved.

It is obvious that, apart from "general education" *ontologies*, there is a large number of professional, specialized *ontologies*, a consistent presentation and knowledge of which is necessary for mutual understanding (compatibility) of all those who work in the relevant professional field.

Thus, the denotational semantics of *SC-code*, like any other language that claims to be universal, reflects the current state of our knowledge and, therefore, may change. Obviously, these changes are most intense in specialized and new areas of knowledge.

VII. Refinement of the concept of semantic compatibility based on the standard of sense information representation

The most important stage in the evolution of any technology is the transition to the **component design** based on the constantly updated **library of reusable components**.

The main problems for the implementation of component design are

- unification of components by form;
- standards development to ensure compatibility of these components.

To implement component design of *knowledge bases* the next is necessary:

- universal language of knowledge representation;
- universal procedure for the integration of knowledge within the specified language;
- development of a standard that provides **semantic compatibility** of integrable knowledge (such a standard is a consistent system of concepts used).

Even for the semantic representation of knowledge, a kind of semantic coordinates are needed, the role of which is played by the used system of concepts (a kind of key signs), which, in turn, is described (specified, defined) by a hierarchical system of semantically interconnected *ontologies*.

In other words, human knowledge must be brought to a common "semantic denominator" (to a common semantic coordinate system), which is the permanently refined system of concepts specified as a unified ontology. This unified ontology **is stratified** to particular ontologies that are sufficiently evolved **independently** from each other.

One of the criteria for the semantic compatibility of new information with the knowledge base into which this information is immersed can be formulated as follows.

All signs that are new to the perceiving knowledge base (in which these new signs are immersed) must be sufficiently specified (and defined for new concepts) through concepts known to the knowledge base.

The standard of sense representation of information (*SC-code*) makes it possible, on the one hand, to increase the level of compatibility of computer systems, and on the other hand, to formally clarify the concept of integration of computer systems and their components.

Consider:

- Semantic integration of two texts belonging to the language of sense representation of information (SC-code). As a result of this integration, the two original sc-texts are converted into one integrated text;
- Semantic integration of two different models of information processing, presented in the SC-code;
- A model of understanding the text of some external language by translating the source external text into an SC-code and then immersing the constructed sc-text into the knowledge base presented in the SC-code.
- Semantic integration of two computer systems based on the SC-code;
- Semantic compatibility of a computer system built on the basis of an SC-code with its users.

A. Refinement of the understanding process based on the sense presentation of information

It is obvious that the formalization of **sense representation of information** in the memory of a computer system greatly simplifies clarifying how the process of understanding new information takes place, which comes to the input of a computer system or is generated during information processing. This process can be divided into three stages:

- **translation** of information from some external language to an internal semantic language (SC-code). This stage is absent if new information is not entered from the outside, but is directly generated in the memory of the computer system;
- **immersion** of new information presented as *sc-text* into the current state of an information resource stored in the memory of a computer system and also represented as *sc-text*;
- **alignment** (agreement) of the concepts used in the new externally entered or generated information

structure with the concepts used in the current state of the information resource stored in the computer system's memory.

Consider each of these steps in more detail.

Translation of information from some external language into the SC-code is simplified due to the fact that:

- means of SC-code allow to describe **syntax** of external language, because the universality of the SC-code allows, with its help and with any degree of detail, to describe any objects, including such complex systems of the external environment of computer systems as external languages;
- the process of the **semantic analysis** of the source text of an external language can be performed by manipulating the texts of the SC-code and, as a result, obtaining a description of the structure of the source text that has sufficient completeness (detailing) for the subsequent generation of a text that is semantically equivalent to it;
- SC-code can be used to describe **semantics** of an external language, treating it as a description of the properties of morphisms between sc-texts describing the syntactic structure of the source external texts, and sc-texts that are semantically equivalent to these source texts;
- the process of **generating sc-text, semantically equivalent to the original** external text, can also be performed by *sc-texts* manipulating.

Thus, the effectiveness of using of *SC-code* for translating text from some external language into *SC-code* is due to the fact that using the *SC-code* we can describe both the syntax and semantics of an external language. We can parse the external text and the subsequent generation of *sc-text*, semantically equivalent to the original external text, while remaining within the *SC-code*.

Immersion (integration) of a new generated *sc-text* into a given *sc-text* (for example, into the knowledge base presented in *SC-code*) reduces to **merging** (identification) of some *sc-elements* of a new *sc-text* with the *sc-elements* that are part of the given *sc-text*. Thus, the task of immersing a new *sc-text* into a given *sc-text* reduces to the task of constructing a set of pairs of synonymous *sc-elements*, one of which is part of the new submersible *sc-text*, and the second is the part of the given *sc-text*.

The establishment of pairs of synonymous *sc-elements* is carried out:

- by searching for pairs of *sc-elements* that have agreed external names that match (we emphasize that all used concepts must have corresponding matching external names);
- by logical reasoning, using logical formulas of the following types:
  - •• non-existence formulas;
  - •• formulas of existence and uniqueness;
  - •• formulas for the existence of a finite and indicated number of values of the corresponding variables.

To simplify the establishment of pairs of synonymous sc-elements, some statements about non-existence, existence and uniqueness, existence of a given finite number of structures of a given type can be reformulated in a more "constructive" key with the explicit introduction of the ***synonymy of sc-elements*** relation. So, for example, instead of the statement that "For each pair of points, there is a single straight line passing through them", the following wording can be used: "If lines ***pi*** and ***pj*** pass through the points ***ti*** and ***tj***, then either ***pi*** = ***pj***, or ***ti*** = ***tj***, or ***ti*** $\notin$ ***pi***, or ***tj*** $\notin$ ***pi***, or ***ti*** $\notin$ ***pj***, or ***tj*** $\notin$ ***pj***".

A sufficiently detailed description of the example of *sc-text* immersion in the knowledge base, also presented in *SC-code*, is given in Section IX of the article [1] – Example 4.

Alignment of concepts, used in the new integrable (introduced, immersed) sc-text, with the concepts used in the given integrating *sc-text*, is as follows:

- The specified integrating *sc-text* (usually this is the knowledge base presented in *SC-code*) must explicitly contain:
  - •• information about the current status (state, character) of using of each concept known to knowledge base and used either directly in the knowledge base itself or by external actors, information from which can be input to the specified knowledge base;
  - •• information about the current status (state, character) of use of each external sign (most often a term, name) corresponding to each concept used, as well as some well-known entities that are not concepts;
- Integrable (input, immersible) text must:
  - •• use **agreed concepts** and the corresponding **agreed external signs** (terms, names) as much as possible;
  - •• include **definitions** of all concepts that are new, unknown in the integrating text (the definition should use only those concepts that are known to the integrating text);
- To solve the problem of the used concepts **alignment** for the current state of the knowledge base and for the new text (integrated) into this text knowledge base, all concepts used in the knowledge base are divided into:
  - •• currently agreed (recognized) and not changing their status;
  - •• obsolete = concepts used before or rarely used now;
  - •• obsoleting = concepts for which, for a given period of time, their status is replaced from the

status of the agreed concept to the status of the obsolete concept;
  - returning = concepts, the status of which changes from the status of the obsolete concept to the status of the agreed concept;
  - proposed new concepts = new concepts undergoing approval = concepts, the status of which changes from the status of proposed to the status of either approved or obsolete = agreeing concepts;
  - approved concepts = concepts that have been successfully negotiated;
  - rejected concepts = concepts whose agreement results are negative;
  - introduced new concepts = concept, the status of which changes from status of the approved concept to the status of a agreed concept = concepts introduced into use.

Thus, the process of alignment of concepts, the goal of which is to reduce all the concepts used in the integrated *sc-text*, to the agreed concepts of *knowledge base*, is carried out **under the conditions of a permanent change in the status of the used concepts** and constant increase numbers of such concepts.

It should be distinguished:

- family of all concepts known to *knowledge base* at the current moment;
- the current status of all these concepts;
- the set of all transition processes aimed at changing the status of concepts and being implemented at the moment.

Note also that the permanent process of agreement of all the concepts used is a necessary condition for ensuring compatibility (integrability) of *SC-code* texts. But to ensure compatibility of *SC-code* texts, a permanent process is needed to agree not only the concepts used themselves, but also the corresponding *external signs* (names, terms). Moreover, *external signs* (names) and their agreement may be required not only for concepts, but also for entities of other types (for example, for people, settlements, geographical objects, historical events, etc.).

We emphasize at the same time that the principles of organizing the agreement of *external signs* (names) are similar to the principles of organizing the agreement of concepts discussed above in the context of their permanent change. So, for example, each connection of the ***be external sign**** relation. linking the sc-sign of some entity with the *sc-node* denoting the external file of sign of the specified entity, as well as each concept, can be put in compliance with its current status (agreed, obsolete, obsoleting, returning, proposed, approved, rejected, included).

Finishing the consideration of the model of understanding as a model of semantic input of some text, not necessarily belonging to *SC-code*, into the given text *SC-code*, we make several remarks.

Understanding may be distorted (including contradictory) and superficial (incomplete) due to poor-quality immersion of new information in the current state of an information resource stored in the memory of a computer system (error in identifying signs and, as a consequence, incorrectly established synonymy, or incompleteness of identification, not all new signs, synonymous with the knowledge base, are merged with their synonyms).

The problem of understanding, mutual understanding between people, between computer systems, between computer systems and their users is the epicenter of the modern stage of evolution of computer systems and is waiting to be solved. The deeper we penetrate the formalization of the process of understanding (especially the understanding of the texts of natural language), the more and more it is surprising that people still somehow understand each other, although not always. More often it is not an understanding, but an illusion of understanding. Here it is appropriate to recall the well-known phrase: "Happiness is when you are understood."

B. Unification and compatibility of various models of problem solving

Our proposed approach to a significant increase in the level of compatibility (integrability) of various **problem solving models** is as follows:

- All information stored in the memory of each ***problem solver*** (both the actual information processed and the interpreted skills stored in the memory, for example, a different type of program), is presented in the form of a sense representation of this information (in *SC-code*);
- Actually, the solution of each task is carried out by a team of agents working on a common sense (semantic) memory and interpreting the skills stored in the same memory (these agents will be called ***sc-agents***);
- The integration of two different models for problems solving is reduced:
  - to combining the memory of the first model with the memory of the second model;
  - to the integration of all *sc-text* stored in the memory of the first model, with *sc-text* stored in the memory of the second model (this integration is carried out by mutual immersion of these *sc-texts* into each other, i.e., by merging together synonyms, as well as by aligning the concepts they use);
  - to the union of the set of agents included in the first model with the set of agents included in the second model of problem solving.

Thus, the unification of problem solving models by reducing these models to the form of sc-models (i.e., sc-text processing models) improves the compatibility level of these models due to the transparent integration of processed and integrable sc-texts and the trivial union procedure for sets of *sc-agents*. The simplicity of the procedure for union of sets of *sc-agents* corresponding to different models of problem solving is due to the fact that there is no direct interaction between these agents, and the initiation of each of them is determined by sc-agent itself, as well as the current state of information stored in memory.

Thus, as a basis for the unification of information processing principles in computer systems, it is proposed to use the **multi-agent approach**. The focus on a multi-agent approach is due to the following main advantages of this approach [5]:

- autonomy (independence) of agents, which allows to localize changes made to the system during its evolution, and reduce the corresponding labor costs;
- processing decentralization, i.e. the absence of a single monitoring center, which also allows to localize changes made to the system.

But the modern principles of building ***multi-agent systems*** when applied to multi-agent processing of *knowledge bases* have several disadvantages:

- agent knowledge is represented using highly specialized languages, often not intended to represent knowledge in a broad sense and ontologies in particular;
- most modern multi-agent systems assume that agents interact by exchanging messages directly from the agent to the agent;
- the logical level of interaction between agents is rigidly tied to the physical level of the implementation of a multi-agent system;
- the environment with which agents interact, is specified separately by the developer for each multi-agent system, which leads to significant overhead and incompatibility of such multi-agent systems.

It is proposed to eliminate the listed disadvantages by using the following principles:

- agents are proposed to be communicated by specifying (in the common memory of a computer system) actions (processes) performed by agents and aimed at problems solving;
- the external environment for agents is the same common memory;
- the specification of each agent is described by means of a knowledge representation language in the same memory;
- synchronization of agents' activities is proposed at the level of the processes they perform
- each information process at any time has associative access to the necessary fragments of the knowledge base stored in common memory.

C. Semantic compatibility of computer systems

The compatibility level of **computer systems** is determined by the laboriousness of the implementation of integration procedures (integration, connection of knowledge of these systems), as well as the laboriousness and depth of integration of these systems *problem solvers* (skills and interpreters of these skills). We emphasize at the same time that the integration can be different – from eclecticism to hybridity and synergy, the distance is of enormous size.

Compatible *computer systems* are computer systems for which there is an automatically performed integration procedure that turns these systems into a single ***hybrid system***, within the framework of which each original computer system can free to use any necessary knowledge and skills that are part of another source computer system.

The integral *computer system* can be considered as a problem solver, integrating several models of problem solving and having the means of interaction with the external environment (with other computer systems, with users).

Thus, in order to increase the compatibility level of *computer systems*, it is necessary to convert them to the form *multi-agent systems*, working on a common semantic memory, in which the information is represented by texts of *SC-code*. Such unified *computer systems* it is not always advisable to directly integrate (integrate) into larger *computer systems*. Sometimes it is more expedient to combine them into *teams of interacting computer systems*. But when creating such groups of computer systems, the unification and compatibility of such systems are also very important, since significantly simplify the provision of a high level of mutual understanding. For example, contradictions between computer systems belonging to a team can be detected by analyzing the consistency of ***virtual unified knowledge base*** of this team. Moreover, the consistency of the specified virtual knowledge base can be considered one of the criteria for semantic compatibility of the systems included in the relevant team.

D. Advantages of the semantic presentation of information

Why is it appropriate to move to the *semantic representation of information* in the memory of *computer system*:

- *sense representation of information* is an objective, independent of subjectivity and diversity of syntactic decisions, way of information representation;
- within the framework of the semantic presentation, the procedure of integrating knowledge and immersing new knowledge into the *knowledge base* is greatly simplified;

- greatly simplifies the procedure for bringing a different type of knowledge to a general form (to an agreed system of concepts used);
- greatly simplifies the process of integrating various *problem solvers* and whole *computer systems*;
- significantly simplifies the automation of the permanent process of supporting semantic compatibility (consistency of concepts and ontologies) for *computer systems* in the context of their continuous improvement;
- based on the proposed *standard of sense representation of information* significantly simplifies the integration of various disciplines in the field of artificial intelligence, i.e. building a general formal theory of intelligent systems, since building a general formal model of intelligent systems requires a basic language, within which one could easily move from information (from knowledge) to **metainformation** (to metaknowledge, to specifications of initial knowledge). This is confirmed by the fact that:
  - •• the overwhelming number of concepts of artificial intelligence has a metalinguistic character;
  - •• *SC-code* represents the unity of the language and the metalanguage, remaining within the framework of a simple syntax;
  - •• the formal semantic refinement of almost every concept of artificial intelligence requires a prior formal refinement of the corresponding object language. So, for example, how can one speak strictly about the language of ontologies (i.e„ the language of the specification of subject domains) without specifying the language of representation of these subject domains themselves. How can one speak strictly about the language of the description of information processing methods without specifying the language of the representation of this processed information itself.

VIII. Semantic computer systems and technologies

We propose a solution to the problems of modern information technologies by moving to the *sense representation of information* in the memory of computer systems actually transforms modern computer systems (including modern intelligent systems) into ***semantic computer systems***, which, consequently, are not an alternative branch of development of *computer systems*, but a natural stage of their evolution, aimed at ensuring a high level of their learnability and, first of all, **compatibility**.

The architecture of *semantic computer systems* (see fig. 1) almost coincides with the architecture of intelligent systems based on knowledge bases. The difference here is that in the *semantic computer systems*:

- the knowledge base has sense representation;
- the knowledge and skills interpreter is a group of *agents* processing *knowledge base*.

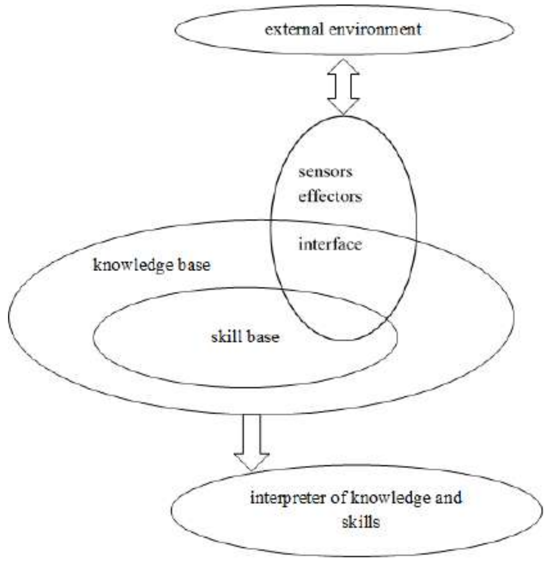

Figure 1. Architecture of ostis-system

As a consequence, *semantic computer systems* have a high level of learnability, i.e. ability to quickly acquire new and improve already acquired knowledge and skills and at the same time not have any restrictions on the type of acquired and improved knowledge and skills, as well as on their sharing.

Moreover, with the agreement of relevant standards, as well as with the permanent improvement of these standards and with their competent support in the conditions of intensive evolution of both the standards themselves and *semantic computer systems* (this is about the permanent support of the correspondence between the current state of computer systems and current state of evolving standards), *semantic computer systems* and their components have a very high degree of compatibility.

This, in turn, virtually eliminates the duplication of engineering solutions and makes it possible to significantly speed up the development of *semantic computer systems* using a constantly expanding library of reusable and compatible components.

The main leitmotif of the transition from modern computer systems (including intelligent) to semantic computer systems, i.e. computer systems based on the sense representation of all the information stored in its memory is the creation of ***general semantic theory of computer systems***, which includes:

- semantic theory of knowledge and knowledge bases;
- semantic theory of problems and models for solving them;

- semantic theory of interaction of information processes;
- semantic theory of user and, including natural language interfaces;
- semantic theory of non-verbal sensory-effector interfaces;
- theory of universal interpreters of semantic models of computer systems and, in particular, the theory of semantic computers.

The epicenter of the next stage of information technology development is the solution to the problem of providing **semantic compatibility** of *computer systems* and their components. To solve this problem is needed

- transition from traditional computer systems and from modern intelligent systems to *semantic computer systems*;
- standard of development of semantic computer systems.

Obviously, *semantic computer systems* are the new generation of computer systems that eliminate many of the shortcomings of modern computer systems. But for the mass development of such systems, an appropriate technology is needed, which should include

- theory of semantic computer systems and a complex of all standards ensuring compatibility of developed systems;
- methods and design tools for semantic computer systems;
- methods and tools of permanent improvement of the technology itself.

Our proposed technology for developing semantic computer systems is named ***OSTIS*** (Open Semantic Technology for Intelligent Systems).

The basis of this technology is ***SC-code*** - the standard of sense representation of information in the memory of computer systems developed by us.

Overall, *OSTIS Technology* is

- **standard** for *semantic computer systems*, ensuring the semantic compatibility of systems conforming to this standard;
- **methods of construction** of such computer systems and their improvement in the course of their operation;
- **tools and means for building** and improving these systems
  - •• language means;
  - •• library of typical technical solutions;
  - •• tools
    - • • • for synthesis and modification;
    - • • • for analyzing, verifying, diagnosing, testing;
    - • • • for eliminating detected errors and flaws.

It is essential to emphasize that *OSTIS Technology* is not just **standard of semantic computer systems**, but a standard that is constantly and intensively improved during the continuous expansion and improvement of the formalization of the types of knowledge used and models for solving problems by reaching a consensus (coordination of points of view) with the participation of all interested individuals and legal entities.

The principal thing is that *OSTIS Technology* allows to create systems that do not necessarily have to solve *intelligent tasks*, but this implementation of *computer systems* provides:

- compatibility;
- high degree of flexibility, which allows unlimited expansion of the functionality of computer systems, including the ability to solve *intelligent tasks*.

We list the principles underlying *OSTIS Technology*:

- orientation to the semantic unambiguous representation of knowledge in the form of semantic networks that have a basic set-theoretic interpretation, which provides a solution to the problem of the diversity of the forms of representation of the same meaning, and the problem of ambiguity of semantic interpretation of information structures;
- use of an associative graph-dynamic memory model;
- application of agent-based knowledge processing model;
- implementation of *OSTIS Technology* in the form of intelligent ***IMS.ostis Metasystem***, which itself is built on *OSTIS Technology* and supports the design of computer systems developed by *OSTIS Technology*;
- ensuring a high level of flexibility, stratification, reflexivity, hybridity, compatibility, and, as a result, learnability of designed systems.

The advantages of OSTIS Technology include:

- *OSTIS Technology* has an open character both for its users (developers of applied intelligent systems) and for those who wish to participate in its improvement;
- *OSTIS Technology* is focused on a constant increase in the pace of its evolution;
- *OSTIS Technology* is the basis for solving the problems of semantic compatibility of various scientific and technical knowledge, since it is focused on the formalization of interdisciplinary connections of the most diverse type.

Perspective directions of *OSTIS Technology* application are:

- Development on the basis of *OSTIS Technology* of a particular technology of designing intelligent reference systems, intelligent semantic textbooks, learning systems and intelligent help-systems in various fields;
- A complete set of compatible semantic electronic textbooks across the entire set of school subjects;

- Intelligent personal assistants (secretaries, referents), providing personalized information services, integration of available services, monitoring and control of users;
- Intelligent control systems of various enterprises, organizations, projects based on ontologies and a formal description of actions performed, events, situations;
- Intelligent automation systems for designing various classes of artificial systems based on ontological models;
- Portals of scientific knowledge and semantic tools of supporting the development of various scientific and technical areas;
- Distributed global semantic knowledge space, which is the result of integrating the knowledge bases of all systems built on *OSTIS Technology* and interconnected by a global network;
- Intelligent systems of excursion service;
- Intelligent systems of complex individual medical monitoring and service;
- Intelligent robotic systems;
- Smart living environment (smart home, smart road, smart city).

IX. Ecosystem OSTIS

Ecosystem OSTIS
= *Sociotechnical ecosystem, which is a group of interacting semantic computer systems and provides permanent support for the evolution and compatibility of all its member systems, throughout their life cycle.*
= *Unlimitedly expandable team of constantly evolving semantic computer systems that interact with each other and with users to solve complex problems in a corporate way and to constantly maintain a high level of compatibility and mutual understanding in interaction both with each other and with users*

Since the above-considered *OSTIS Technology* is focused on the development of *semantic computer systems* with a high level of *learnability* and, in particular, a high level of semantic *compatibility*, and since learnability and compatibility are only ability to learn (i.e., to high rates of expansion and improvement of their knowledge and skills), as well as ability to ensure a high level of mutual understanding (coherence), some kind of environment, social engineering infrastructure, is needed in the framework of which most comfortable conditions have been created for the implementation of the above abilities. This environment is named by us ***Ecosystem OSTIS***, which is a group of interacting (via the Internet):

- *semantic computer systems*, built according to standard of *OSTIS Technology* (such systems will be called ***ostis-systems***);
- users of the specified *ostis-systems* (both end users and developers);
- some computer systems that are not *ostis-systems*, but they are considered as additional information resources or services.

A. Compatibility support between computer systems of Ecosystem OSTIS

The main purpose of ***Ecosystem OSTIS*** is to ensure compatibility of computer systems included in the *Ecosystem OSTIS* both at the stage of their development and during their operation. The problem here is that during the operation of the systems included in the *Ecosystem OSTIS*, they may change due to which compatibility may be violated.

The tasks *Ecosystem OSTIS* are:

- operative implementation of all agreed changes to the *ostis-systems* standard (including changes to the systems of the used concepts and their corresponding terms);
- permanent support of a high level of mutual understanding of all the systems included in the *Ecosystem OSTIS*, and all their users;
- corporate solution of various complex problems requiring coordination of activities of several (most often, a priori unknown) *ostis-systems*, as well as, possibly, some users.

The *Ecosystem OSTIS* is a transition from independent (autonomous, separate, integral) *ostis-systems* to collectives of independent *ostis-system* m, i.e. to distributed *ostis-systems*. The following types of *ostis-systems* can be distinguished by the hierarchy level:

- ***atomic embedded ostis-system***
  = *ostis-system integrated into the independent ostis-system, but not into the composition of another embedded ostis-system*
- ***non-atomic embedded ostis-system***
  = *ostis-system, which is integrated into the independent ostis-system, and includes some other embedded ostis-systems*
  ⊃ *user interface*
- ***independent ostis-system***
  = *consistent ostis-system, which must independently perform the corresponding set of tasks and, in particular, interact with the external environment (verbally – with users and other computer systems, and non-verbally)*
- ***collective of ostis-systems***
  = *a group of communicating ostis-systems, which can include not only independent ostis-systems, but also collectives of ostis-systems*
  = *distributed ostis-sysstem*
- ***Ecosystem OSTIS***
  ∈ *maximum collective of ostis-systems*

∈ *collective of ostis-systems that is not part of another collective of ostis-systems*

We emphasize that the ***independent ostis-systems***, which are part of the *Ecosystem OSTIS*, are met special requirements:

- they must have all the necessary knowledge and skills for messaging and purposeful organization of interaction with other *ostis-systems* belonging to *Ecosystem OSTIS*;
- under the conditions of constant change and evolution of the *ostis-systems* included in the *Ecosystem OSTIS*, each of them should itself monitor its compatibility (consistency) with all the others *ostis- systems* i.e. should independently maintain this compatibility, coordinating with other ostis-systems all changes that need to be coordinated, occurring in themselves and in other systems.
- Each system included in the *Ecosystem OSTIS* must:
  - •• study intensively, actively and purposefully (both with the help of teachers and developers, and independently);
  - •• inform all other systems about proposed or finally approved changes in *ontologies* and, in particular, in the set of *concepts* used;
  - •• accept from other *ostis-systems* proposals for changes in *ontologies* (including the set of concepts used) for agreement or approval of these proposals;
  - •• implement approved changes in *ontologies* stored in its knowledge base;
  - •• help to maintain a high level of semantic compatibility not only with other *ostis-systems* included in *Ecosystem OSTIS*, but also with its *users* (i.e. to train them, inform them about ontology changes).

The Ecosystem OSTIS is a form of realization, improvement and application of *OSTIS Technology* and, therefore, is a form of creation, development, self-organization of the market for semantically compatible computer systems and includes all the necessary resources for this – personnel, organizational, infrastructural.

The *Ecosystem OSTIS* is mapped to its ***integrated knowledge base***, which is **virtual union** of *knowledge bases* of all *ostis-systems* included in *Ecosystem OSTIS*. The quality of this *knowledge base* (completeness, consistency, clearness) is a permanent attention of all the independent *ostis-systems* included in *Ecosystem OSTIS*. Accordingly, each specified *ostis-system* is associated with its own *knowledge base* and its own hierarchical system of *sc-agents*.

By purpose, the *ostis-systems* included in the *Ecosystem OSTIS* can be:

- assistants to specific users or specific user teams;
- standard embedded subsystems of *ostis-systems*;
- information and tool support systems for designing various components and various classes of *ostis-systems*;
- information and tool support systems for designing or producing various classes of technical and other artificially created systems;
- knowledge portals for various scientific disciplines;
- automation systems for managing various complex objects (industrial enterprises, educational institutions, departments of universities, specific students);
- intelligent reference and help-systems;
- intelligent learning systems, semantic electronic tutorials;
- intelligent robotic systems.

B. Compatibility support between computer systems and their users in the Ecosystem OSTIS

There are two aspects to maintaining compatibility and understanding in the *Ecosystem OSTIS*

- compatibility support between *ostis-systems* included in *Ecosystem OSTIS*;
- compatibility and mutual understanding between the *ostis-systems* included in the *Ecosystem OSTIS* and their users, with active encouragement from the *Ecosystem OSTIS*, so that each user of *Ecosystem OSTIS* at the same time is not only its active end user, but also its active developer.

Thus, to ensure high operational efficiency and high rates of evolution of *Ecosystem OSTIS*, it is necessary to constantly increase the level of information compatibility (level of mutual understanding) not only between the computer systems that make up the *Ecosystem OSTIS*, but also between these systems and their users. One of the ways to ensure such compatibility is the desire to ensure that each user's *knowledge base* (picture of the world) becomes a part (fragment) ***Joint Knowledge Base of Ecosystem OSTIS***. This means that each user should know how the structure of each scientific and technical discipline is arranged (objects of research, subjects of research, definitions, statements, etc.), and how different disciplines can be interconnected.

The formation of such system building skills of the picture of the World should be started from the secondary school. For this purpose, it is necessary to create a set of compatible intelligent learning systems for all secondary education disciplines with clearly described interdisciplinary connections [17], [18]. Thanks to this, it is possible to prevent the users from forming the "mosaic" picture of the World as a multitude of poorly related disciplines. And this, in turn, means a significant improvement in the quality of education, which is absolutely necessary for high-quality operation of next-generation computer systems – *semantic computer systems*.

Users and, first of all, the developers of *Ecosystem OSTIS* should have a high level of:

- mathematical culture (formalization culture) when building a formal model of the environment in which an intelligent system functions, formal models of the problems it solves and formal models of various methods of problems solving it uses;
- system culture, which allows to adequately assess the quality of the developed systems from the point of view of the general theory of systems and, in particular, assess the overall level of automation implemented with the help of these systems. System culture involves the desire and ability to avoid eclecticism, the desire and ability to provide high-quality stratification, flexibility, reflexivity, as well as high-quality maintenance, a high level of learnability and a comfortable user interface of the systems being developed;
- technological culture, ensuring compatibility of the developed systems and their components, as well as the continuous expansion of the library of reusable components of the created systems and assuming a high level of design discipline;
- ability to work in a team of developers of high-tech systems, which implies a high level of ability to work at interdisciplinary junctions, a high level of communication skills and agreeability, i.e. the ability not only to defend one's point of view, but to coordinate it with the views of other developers in the interests of development *Ecosystem OSTIS*;
- activity and responsibility for the overall result – high rates of evolution *Ecosystem OSTIS* in general.

Thus, the high evolution rates of *Ecosystem OSTIS* are provided not only by the professional qualifications of users (knowledge of *OSTIS Technology*, current status and problems of *Ecosystem OSTIS* and skills of using *OSTIS Technology* and intelligent systems included in the *Ecosystem OSTIS*), but also the relevant human qualities. Obviously, the modern level of agreeability, activity and responsibility cannot be the basis for the evolution of such systems as *Ecosystem OSTIS*.

Support compatibility of *Ecosystem OSTIS* with its users carried out as follows:

- each *ostis-system* includes embedded ostis-systems oriented on
  - permanent monitoring of the activities of end users and developers of this *ostis-system*,
  - analysis of the quality and, above all, the correctness of this activity,
  - permanent unobtrusive personalized training aimed at improving the quality of user activity, i.e. to improve their skills;
- within the *Ecosystem OSTIS* there are *ostis-systems*, specifically designed to train users of *Ecosystem OSTIS* to the basic recognized knowledge and skills to perform the corresponding classes of tasks. This includes the knowledge corresponding to the level of secondary education, and knowledge corresponding to the basic disciplines of higher education in the field of informatics (and, in particular, in the field of artificial intelligence), and basic knowledge of *OSTIS Technology* and about *Ecosystem OSTIS*.

The problem of creating a market for compatible computer systems is the **challenge to modern science and technology**. Scientists working in the field of artificial intelligence require the ability to work collectively on solving interdisciplinary problems and bring these solutions to a general integrated theory of intelligent systems, involving the integration of all areas of artificial intelligence, and to technologies available to a wide range of engineers. Intelligent systems engineers are required to actively participate in the development of relevant technologies and to significantly increase the level of mathematical, systemic, technological, and organizational-psychological culture.

But the main task here is to reduce the barrier between scientific research in the field of artificial intelligence and engineering in the development of intelligent systems. For this, science should be constructive and focused on the integration of its results in the form of an integrated technology for developing intelligent systems, and engineering, having realized the knowledge-intensiveness of its activities, should actively participate in the development of technologies.

Particular emphasis in the *Ecosystem OSTIS* is placed on the ongoing process of agreement of *ontologies* (and, first of all, on the harmonization of the family of all used concepts and terms corresponding to these concepts) between all (!) active subjects of *Ecosystem OSTIS* – between all *ostis-systems* and all users.

In the presence of *ostis-systems*, which are personal assistants of users in cooperation with the *Ecosystem OSTIS*, this whole Ecosystem will be perceived by users as a single intelligent system uniting all information resources and services available in the *Ecosystem OSTIS*.

The principles of organization of *Ecosystem OSTIS* create all the necessary conditions for attracting scientific, organizational and financial resources to the development and improvement of *OSTIS Technology*, which will be aimed at developing methods and means of artificial intelligence and forming a market for semantically compatible intelligent systems.

X. IMS.ostis Metasystem

The effectiveness of any technology, including ***OSTIS Technology*** [16] is determined not only by the time terms for the creation of artificial systems of the corresponding class, but also by the rates of improvement of the technology itself (rates of improvement of automation tools underlying technology).

To fixate the current state of *OSTIS Technology*, as well as to organize its effective use and its permanent improvement with the participation of scientists working in the field of artificial intelligence, and engineers who develop semantic computer systems for various purposes into the *OSTIS Ecosystem* the *IMS.ostis* [16] is introduced, the purpose of which makes it key *ostis-system* within the *OSTIS Ecosystem*.

IMS.ostis Metasystem
= *Intelligent metasystem of integrated informational and instrumental support for the design of compatible semantic computer systems, which is a form of realization of the general theory and technology of designing semantic computer systems and which maintains a high rate of evolution of this theory and technology*
= *Intelligent MetaSystem for intelligent systems design*
= *IMS.ostis*
= *Intelligent System Framework*
= *Intelligent metasystem of complex support for the design of compatible semantic computer systems using OSTIS Technology*
= *Framework of ostis-systems*
= *Framework IMS.ostis*

The IMS.ostis Metasystem is in the *Ecosystem OSTIS* a key intelligent system that supports not only the design of new intelligent systems and not only the replacement of obsolete components in the intelligent systems included in the *Ecosystem OSTIS*, but also inclusion (integration) in the *Ecosystem OSTIS* of newly created intelligent systems.

IMS.ostis Metasystem is focused on the development and practical implementation of methods and tools **component design** and semantically compatible intelligent systems, which provides the ability to quickly create intelligent applications for various purposes.

The areas of practical application of the component design technology of semantically compatible intelligent systems are not limited by anything.

A. Structure of developed ostis-systems

The architecture of computer systems developed by *OSTIS Technology* is clearly stratified into two subsystems:

- *knowledge base*, which is a complete semantic model of an intelligent system (which will be called the sc-model of the intelligent system or the sc-model of the knowledge base of the intelligent system, as it is formed as a coherent sign construct belonging to *SC-code* – the base language of the internal sense representation of knowledge in memory of *ostis-systems*);
- basic universal interpreter of the semantic model of an intelligent system stored in its memory (the interpreter of the sc-model of the knowledge base of an intelligent system).

These subsystems of *ostis-systems* can be developed completely independently of each other with the observance of clear requirements imposed by *OSTIS Technology* which consist in the interpretation of the syntax and semantics of *SC-code* that are identical for these subsystems which is the universal language of the internal semantic representation of knowledge in the memory *ostis-systems*, as well as the syntax and semantics of ***SCP language*** (Semantic Code Programming), which is the sublanguage of *SC-code* and is a basic language of agent-oriented programming which is focused on processing of sign structures, belonging to *SC-code*.

The considered stratification of *ostis-systems* to compatible with each other *knowledge base* and the knowledge base interpreter, firstly, provides ample opportunities for a wide variety of implementation options for the interpreter of sc-models of knowledge bases (including various implementations of semantic computers with associative graph-dynamic, reconstructable memory) and, secondly, makes it possible to easily transfer (reload) the knowledge base of an intelligent system into the memory of another knowledge base interpreter. The second possibility means the platform independence of the intelligent systems developed by *OSTIS Technology*, since the various implementations of interpreters of sc-models of knowledge bases are nothing but different platform options for *ostis-systems* implementing.

Thus, if there is a sufficiently effective version of the implementation of the interpreter of sc-models of knowledge bases, the development of *ostis-system* comes down to designing ***sc-model of its knowledge base*** [15], which includes itself:

- *sc-model of the integrated problem solver of this ostis-system* [19], which, in turn, includes:
  - sc-models of classes of problems to be solved (in particular, stored programs of high-level languages);
  - scp-programs of knowledge processing agents;
- *sc-model of the integrated interface of the ostis-system*, which is a built-in ostis-system, focused on solving interface problems related to ensuring the direct interaction of the ostis-system with the external environment (both non-verbal receptor-effector interaction, and verbal interaction with users, with other ostis-systems, with other computer systems).

B. Technical implementation of the IMS.ostis Metasystem

The purpose of *IMS.ostis Metasystem* is the implementation of the design technology of semantically compatible computer systems in the form of a metasystem built using the same technology and providing comprehensive information and tool support for designing semantically compatible computer systems. The composition of the specified metasystem includes:

- full description of the technology itself;
- history of the evolution of technology;
- description of technology usage rules;
- description of the organizational infrastructure aimed at the development of technology;
- library of reusable compatible components of intelligent systems;
- methods and tools for designing various types of intelligent system components;
- technical tools of coordinating the activities of project participants, aimed at the continuous improvement of technology.

Tasks of ***IMS.ostis Project*** are:

- To develop *IMS.ostis Metasystem*, which provides fast component design of semantically compatible computer systems for various purposes.
- To develop methods and tools to ensure the intensive development of the market for semantically compatible applied intelligent systems created on the basis of *IMS.ostis Metasystem*.
- To develop methods and means to stimulate the intensive development of the *IMS.ostis Metasystem*.

The scientific novelty of *IMS.ostis Metasystem* is the unification of the representation of various types of information in the memory of computer systems based on the sense (semantic) presentation of this information, which ensures:

- avoiding duplication of the same information in different intelligent systems and in different components of the same system;
- semantic compatibility of various components of intelligent systems and various intelligent systems in general;
- is a significant expansion of libraries of compatible reusable components of computer systems due to "large" components and, in particular, typical subsystems.

The principles of the technical implementation of *IMS.ostis Metasystem* completely coincide with the principles of the technical implementation of applied intelligent systems developed with the help of this metasystem. Thus, the *IMS.ostis Metasystem* is an intelligent system designed for comprehensive information and tool support for designing semantically compatible computer systems, the purpose of which is not imposed any restrictions.

The knowledge base of *IMS.ostis Metasystem* includes:

- current state of models and methods used in the development of intelligent systems using *IMS.ostis Metasystem*;
- systematic library of reusable and compatible components of intelligent systems;
- description of design tools for various types of intelligent systems components (fragments of knowledge bases, problem solvers, user interfaces);
- description of the tools of coordinating collective activities aimed at the continuous development of *IMS.ostis Metasystem*;
- description of the evolution history of *IMS.ostis Metasystem*;
- description of design tools for various classes of intelligent systems.

The problem solver and the user interface of *IMS.ostis Metasystem* provide support for the entire complex of design tasks solved by the developers of applied intelligent systems, as well as by the developers of the *IMS.ostis Metasystem*.

The *IMS.ostis Project* is implemented in the form of interaction of *IMS.ostis Metasystem* with its users and is based on the following principles:

- In order to stimulate the development of the market of compatible application intelligence systems developed with the help of *IMS.ostis Metasystem* and the development of this metasystem itself, technical tools are used to analyze and evaluate the object and significance of the personal contribution of each developer in special arbitrary units.
- In order to stimulate the development of a market for compatible application-based intelligent systems developed using *IMS.ostis Metasystem*, for each such intelligent system registered and specified in the framework of *IMS.ostis Metasystem*, developers are given remuneration in the used conventional units after this application has been tested for semantic compatibility with other systems developed using the *IMS.ostis Metasystem*. At the same time *IMS.ostis Metasystem* becomes a platform for advertising and distribution of intelligent systems developed with its help.
- Stimulating the development of the *IMS.ostis Metasystem* is as follows. Participation in the development of the *IMS.ostis Metasystem* is open, for which it is sufficient to register accordingly. The copyright of each developer of *IMS.ostis Metasystem* is protected and each of his contributions, depending on his value, is automatically measured and recorded in the conventional units used.
- Participation in the development of *IMS.ostis Metasystem* can take a variety of forms (in the simplest case, it can be an indication of specific errors, specific difficulties that the user has encountered, the formulation of specific wishes; a more complicated contribution is to add to knowledge base of new metasystem knowledge, new components in the library of reusable components). At the same time, the author of a new reusable component included in the *IMS.ostis Metasystem* library can choose any

license for its distribution and, in particular, assign it any price.

- The use of *IMS.ostis Metasystem* by registered users is free to use with them. In the commercial development of applied intelligent systems, the cost of each access to the *IMS.ostis Metasystem* is quite affordable, but significantly reduced, depending on the level of user activity in the development of *IMS.ostis Metasystem*. This is another mechanism to stimulate participation in the development of *IMS.ostis Metasystem*.

Thus, the specified principles of the *IMS.ostis Metasystem* provide on an ongoing basis the involvement of unlimited scientific, technical and financial resources and, in particular, unlimited scientific, technical and financial resources to develop the market for semantically compatible applied intelligent systems attracting any professionals who want to participate in this open project.

XI. A FAMILY OF VARIOUS OPTIONS FOR IMPLEMENTING A UNIVERSAL INTERPRETER OF SEMANTIC MODELS OF COMPUTER SYSTEMS

universal interpreter of sc-models of computer systems
= *typical built-in basic ostis-system*
= *built-in empty ostis-system*
= *universal interpreter of sc-models of ostis-systems*
= *universal basic ostis-system, providing simulation of any ostis-system by interpreting the sc-model of the simulated ostis-system*
/*the relationship between the simulated and universal ostis-system is to a certain extent similar to the relationship between the Turing machine and the universal Turing machine*/
= *SCP language program interpreter*
/*Semantic Code programming*/
= *scp-machine*

The implementation of the *universal interpreter of sc-models of computer systems* may have a large number of options, both software and hardware implemented. The logical architecture of *universal interpreter of sc-models of computer systems* ensures the independence of the designed computer systems from the variety of options for the implementation of the interpreter of their models and includes:

- *semantic graph associative memory* (sc-memory, sc-storage of sign structures represented in the SC-code);
- *interpreter of the SCP language* which is a basic procedural programming language oriented to the processing of texts of the SC-code stored in a semantic graph associative memory.

A. Hardware implementation of a universal interpreter of semantic models of computer systems

Semantic associative computer
= *Hardware-implemented interpreter of semantic models (sc-models) of computer systems*
= *Semantic associative knowledge-driven computer*
= *A computer with a non-linear structurally reconstructable (graph-dynamic) associative memory, processing of information in which is reduced not to a change in the state of the memory elements, but to a change in the configuration of the connections between them*
= *sc-computer*
= *scp-computer*
= *Computer driven by knowledge presented in the SC-code*
= *Computer oriented on SC-code texts processing*

The basic principles underlying the *semantic associative computer*:

- non-linear memory – each elementary fragment of text stored in memory may be incident to an unlimited number of other elementary fragments of this text;
- reconstructable (reconfigurable) memory – the processing of the information stored in memory is reduced not only to changing the state of the elements, but also to reconfiguring the connections between them;
- as an internal method of coding knowledge stored in the memory of a semantic associative computer, we use a universal (!) method of nonlinear (graph-like) semantic representation of knowledge, which we called the SC-code (semantic, semantic computer code);
- information processing is carried out by a team of agents working over common memory. Each of them responds to the corresponding situation or event in memory (a computer is controlled by stored knowledge);
- there are software-implemented agents whose behavior is described by in-memory agent-oriented programs, which are interpreted by the relevant groups of agents;
- there are basic agents that cannot be implemented programmatically (in particular, they are agents of agent program interpretation, basic receptor agent-sensors, basic effector agents);
- all agents work over common memory at the same time. Moreover, if for some agent at some point in time there are several conditions for its use in different parts of memory, different acts of the specified agent in different parts of memory can be executed simultaneously (an agent act is an indivisible, consistent process of agent activity);
- to ensure that agent acts that are executed in parallel in the shared memory do not "interfere" with each other, for each act, its current state is fixed and constantly updated in memory. That is, each

act informs everyone else about its intentions and wishes that other agents should not interfere with (for example, these are various types of locks of the used elements of semantic memory);
- besides, agents (more precisely, acts performed by them) must comply with "ethics" trying not to harm themselves to create the most favorable conditions for other agents (acts), for example, not to be greedy, to return faster, not to lock extra memory elements, as soon as possible to release (unlock) locked memory elements;
- the processor and the memory of the semantic associative computer are deeply integrated and constitute a single processor memory. The processor of the semantic associative computer is uniformly "distributed" in its memory so that the processor elements are simultaneously the elements of the computer's memory. Information processing in the semantic associative computer is reduced to the reconfiguration of communication channels between the processor elements, therefore the memory of such a computer is nothing more than a switchboard (!) of the specified communication channels. Thus, the current state of the configuration of these communication channels is the current state of the information being processed.

XII. Embedded intelligent system for collective development of semantic knowledge bases

It is known that the development of a knowledge base of intelligent systems is a very laborious process, in many ways determining the quality of an intelligent system. It is also obvious that shortening the development time of the knowledge base is possible through the organization of collective development, but it leads to the number of problems, for example:

- How within the team of developers of the same knowledge base to prevent the syndrome of "swan, crayfish and pike" , or the syndrome of "seven nannies" and how to reduce the overhead costs of coordinating their activities to create a quality knowledge base.
- How to ensure the possibility of including any already formalized knowledge into the knowledge base of any intelligent system (if they are needed there) without any "manual" adjustments of this knowledge and thereby completely eliminate the redevelopment and adaptation of this knowledge.

The quality of the knowledge base is determined by its following characteristics:

- fullness = integrity = no information holes
- consistency = correctness = no errors
- relevance = compliance with the current state of the environment and the current state of human knowledge about the environment
- structuring.

The design of intelligent systems consists in building a semantic model of this intelligent system, which includes the model of the knowledge being processed, various models for solving various classes of problems, and various models for the interaction of intelligent systems with its external environment. In this case, the knowledge being processed can be both problem-solving models in the knowledge base, and models for solving interface problems, which, respectively, should also be part of the knowledge base of intelligent systems.

A set of tools for designing intelligent systems can be divided into

- tools for knowledge base design;
- tools for intelligent system solvers design;
- tools for intelligent systems interfaces design.

At the same time, it is essential to emphasize that the design of the problem solver of the intelligent system consists in the design of knowledge of a special type – the skills and specifications of the agents who interpret these skills when solving specific tasks. The design of interfaces of intelligent systems is reduced to the design of knowledge, which is a semantic model of an embedded intelligent system, focused on solving of interface problems.

Embedded typical intelligent system of complex support for knowledge bases design
= *Embedded typical intelligent system for complex automation of design, as well as managing the process of collective design and improving knowledge bases of intelligent systems at all stages of their life cycle*
= *Intelligent computer-aided knowledge base design system*
= *Embedded intelligent system, supporting the design and improvement of knowledge bases of intelligent systems at all stages of their life cycle*
= *Intelligent computer framework of knowledge bases of intelligent systems developed by OSTIS Technology*
= *System for the collective knowledge bases development support based on OSTIS Technology*

This embedded intelligent system performs:

- monitoring of the activities of each participant in the process of designing knowledge bases, which is necessary to protect his copyright, to assess the scope and significance of his contribution to the project activity, to assess his professional qualifications, to qualitatively assign new design works, taking into account his current qualifications and planned directions of his qualification enhancement, for the implementation of rollbacks, that is, the

cancellation of erroneous decisions made by administrators or managers of the projected knowledge base;
- version control of the designed knowledge base, the implementation of the necessary rollbacks to previous versions;
- control of performing discipline;
- analysis of the current state and dynamics of the design process, identification of critical situations;
- semantic analysis of the correctness of the results of the design work of all participants;
- assessment of the scope and significance of the activities of each project participant;
- assessment of the current status and dynamics of the development of the qualification portrait of each project participant;
- formation of recommendations for improving the skills of each project participant;
- quality control (consistency, integrity, completeness, clearness) of the current state of the designed and improved knowledge base.

Each participant in the knowledge base design process can perform various types of design work:

- propose a new fragment in the agreed part of the knowledge base or some adjustment (deletion, modification) in this part of the knowledge base;
- agree or disagree with the proposed correction or addition to the agreed part of the knowledge base;
- verify, test, review the correction proposed by someone or add to the agreed part of the knowledge base and write comments on the finalization of this proposal;
- propose the wording of a new project task, for example, to eliminate the indicated contradiction (errors), to fill in the indicated information hole;
- make constructive criticisms to the wording of the new project task;
- suggest a performer or a group of performers to perform a project task that is not yet being performed;
- make constructive criticisms to the proposed performers of some free project task.

XIII. Scientific knowledge portals that formalize interdisciplinary communication

The objectives of the intelligent portal of scientific knowledge are:

- Acceleration of immersion of each person in new scientific areas with constant preservation of the overall consistent picture of the World (educational goal);
- Fixation in a systematized form of new scientific results so that all the main connections of new results with known ones are clearly marked;
- Automation of coordination of work on the review of new results;
- Automate the analysis of the current state of the knowledge base.

The creation of intelligent portals of scientific knowledge, providing an increase in the pace of integration and the reconciliation of various points of view, is a way to substantially increase the pace of evolution of scientific and technical activity.

Compatible portals of scientific knowledge, implemented in the form of *ostis-systems*, included in *Ecosystem OSTIS*, are the basis of the new principles of organization of scientific activity, in which

- the results are not articles, monographs, reports and other scientific and technical documents, but fragments of a global knowledge base, the developers of which are freely formed scientific teams consisting of specialists in relevant scientific disciplines,
- use the portal of scientific knowledge is carried out
  - •• to coordinate the process of reviewing new scientific and technical information from scientists to the knowledge bases of these portals,
  - •• the process of coordinating the different points of view of scientists (in particular, the introduction and semantic correction of concepts, as well as the introduction and correction of terms corresponding to different entities).

The implementation of a family of semantically compatible scientific knowledge portals in the form of compatible *ostis-systems*, included in *Ecosystems OSTIS*, involves the development of a hierarchical system of semantically consistent formal ontologies corresponding to various scientific and technical disciplines, with a clearly defined inheritance of the described entities properties with well-defined interdisciplinary connections that are described by the connections between the corresponding formal ontologies and the subject domains they specify.

Implementing scientific knowledge portals as a family of semantically compatible *ostis-systems* also means trying to overcome the "babel" diversity of scientific and technical languages, not changing the essence of scientific and technical knowledge, but reducing this knowledge to a single universal form of semantic knowledge in the memory of scientific knowledge portals, i.e. to a form that is sufficiently clear to both *ostis-systems*, and any potential users.

An example of a scientific knowledge portal built in the form of *ostis-system* is the *IMS.ostis Metasystem*, which contains all the currently known knowledge and skills that are part of the *OSTIS Technology*.

XIV. Conclusion

The main directions of solving the problem of information compatibility of computer systems are:

- *semantic information technology*, which is based on the sense representation of information in the memory of computer systems;

- *self-organizing ecosystem* supporting the evolution and compatibility of computer systems built on semantic information technology during the operation of these systems.

Thus, the current stage of development of traditional and intelligent information technologies marks the transition from modern information technologies to ***semantic information technologies*** and to the corresponding self-organizing ecosystem consisting of ***semantic computer systems***. The epicenter of the current stage of development of information technology is to ensure and self-ensure the information compatibility of computer systems and the consistency of their functioning.

Obviously, the pace of development of semantic information technologies, as well as the market for applied semantic computer systems, depends primarily on the number of professionals involved in the development of these technologies and in expanding the diversity of their applications. The most effective form of achieving these goals is **open projects** and, above all, an open development project of *IMS.ostis Metasystem*, providing an opportunity for everyone to contribute to the development of semantic information technologies.

The website of the Belarusian Association of Specialists in the Field of Artificial Intelligence (http://baai.org.by [20]) provides information on a number of such open-source projects developed and supported by this association of specialists.

Acknowledgment

This work was supported by the BRFFR-RFFR (No F18R-220).

References

[1] V. Golenkov, N. Guliakina, N. Grakova, I. Davydenko, V. Nikulenka, A. Eremeev, and V. Tarassov, "From training intelligent systems to training their development tools," *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 81–98, 2018.

[2] A. N. Borisov, "Postroenie intellektual'nyh sistem, osnovannyh na znanijah, s povtornym ispol'zovaniem komponentov [Building of intelligent knowledge-based systems with reusable components]," *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 97–102, 2014.

[3] T. A. Gavrilova, D. V. Kudryavcev, and D. I. Muromcev, *Inzheneriya znanij: modeli i metody : uchebnik [Knowledge Engineering: Models and Methods: Textbook]*. Spb. : Lan', 2016.

[4] World Wide Web Consortium [Electronic resource]. Mode of access: http://www.w3.org. — Date of access: 24.11.2016.

[5] M. Wooldridge, *An introduction to multiagent systems*. Chichester : J. Wiley, 2009.

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

[7] I. Z. Batyrshin, A. O. Nedosekin, A. A. Stetsko, V. B. Tarasov, A. V. Yazenin, and N. G. Yarushkina, *Nechetkie gibridnye sistemy. Teoriya i praktika [Fuzzy hybrid systems. Theory and practice]*. M. : FIZMATLIT [M.: PHYSMATLIT], 2007.

[8] A. Kolesnikov, *Gibridnye intellektual'nye sistemy: Teoriya i tekhnologiya razrabotki [Hybrid intelligent systems: theory and technology of development]*, A. M. Yashin, Ed. SPb.: Izd-vo SPbGTU, 2001.

[9] (2018, dec) Artificial general intelligence conference (agi). [Online]. Available: http://agi-conference.org

[10] A. Palagin, "Problemy transdisciplinarnosti i rol' informatiki [transdisciplinarity problems and the role of informatics]"," *Kibernetika i sistemnyj analiz [Cybernetics and systems analysis]*, vol. 5, pp. 3–13, 2013.

[11] A. Yankovskaya, "Analiz dannykh i znanii na osnove konvergentsii neskol'kikh nauk i nauchnykh napravlenii [data and knowledge analysis based on the convergence of several sciences and scientific fields]," in *Mezhdunarodnaya konferentsiya "Intellektualizatsiya obrabotki informatsii" (IOI-8) [International Conference "Intellectualization of Information Processing" (IIP-8)]*. Kipr, 2010, pp. 196–199.

[12] V. Martynov, *Semiologicheskie osnovy informatiki [Semiological foundations of computer science]*. Minsk: Nauka i tekhnika [Science and technics], 1974.

[13] I. Melchuk, *Opyt teorii lingvisticheskih modelej Smysl-Tekst*. Shkola «Yazyki russkoj kul'tury», 1999.

[14] I. A. Melchuk, "Kak nachinalas' matematicheskaya lingvistika [how did mathematical linguistics begin]," *Ocherki istorii informatiki Rossii [Essays on the history of computer science in Russia]*, pp. 358–370, 1998.

[15] I. Davydenko, "Semantic models, method and tools of knowledge bases coordinated development based on reusable components," *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 99–118, 2018.

[16] (2018, Dec) Ims.ostis metasystem. [Online]. Available: https://ims.ostis.net

[17] A. Bashmakov and I. Bashmakov, *Razrabotka komp'yuternykh uchebnikov i obuchayushchikh sistem [Development of computer textbooks and training systems]*. M: Informatsionno-izdatel'skii dom Filin" [Information and Publishing House Filin]], 2003.

[18] V. Taranchuk, "Vozmozhnosti i sredstva wolfram mathematica dlya razrabotki intellektual'nykh obuchayushchikh sistem [opportunities and means of wolfram mathematica for developing intelligent tutoring systems]," *«Nauchnye vedomosti Belgorodskogo gosudarstvennogo universiteta. Seriya: Ekonomika. Informatika» [«Scientific statements of Belgorod State University. Series: Economy. Computer science»]*, vol. 33, no. 1 (198), pp. 102–110, 2015.

[19] D. Shunkevich, "Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems," *Otkrytye semanticheskie tehnologii proektirovanija intellektual'nyh sistem [Open semantic technologies for intelligent systems]*, pp. 119–132, 2018.

[20] (2018, Dec) Belarusian public association of experts in the field of artificial intelligence. [Online]. Available: https://baai.org.by

МЕТОДЫ И СРЕДСТВА ОБЕСПЕЧЕНИЯ СОВМЕСТИМОСТИ КОМПЬЮТЕРНЫХ СИСТЕМ

Голенков В.В., Гулякина Н.А., Давыденко И.Т., Еремеев А.П.

В работе рассмотрены основные актуальные проблемы в области разработки современных компьютерных систем, в частности – проблема обеспечения информационной совместимости компьютерных систем. Предложен подход к них решению, основанный на использовании Открытой семантической технологии проектирования интеллектуальных систем (OSTIS).

Received 11.12.18