Methods and tools for designing and analyzing the quality of knowledge bases of next-generation intelligent computer systems

Stanislau Butrin
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: stas.butrin1331@gmail.com

***Abstract*—In the article, an ontological approach to the design of knowledge bases of next-generation intelligent computer systems is considered. It is based on the usage of a multi-agent approach to ensure the consistency of tools for knowledge base quality analysis. The results will improve the efficiency of designing knowledge bases of intelligent computer systems.**

***Keywords*—knowledge base, ontology, subject domain, knowledge base verification, knowledge base quality analysis**

I. Introduction

An important phase in the development of any system is quality control, since the degree of liveness and effectiveness of the system are determined at this stage.

It is believed that the quality of intelligent computer systems is largely determined by the quality of their knowledge bases [1].

Currently, tools for creating knowledge bases of intelligent computer systems are rapidly developing. However, the development of knowledge bases is a collaborative process, which is characterized by the occurence of contradictions and misunderstandings. Therefore, special attention should be paid to the means of checking and verification of knowledge operated by intelligent systems.

In the real world, contradictions and errors are inevitable, since humans, like the systems created by them, will be limited to a certain picture of the world, within which some knowledge will be considered as true and others as false. Nevertheless, the system must be able to adapt, adjust its picture of the world and its concept of certain information, in order to effectively perform the tasks for which it was created.

II. Analysis of existing approaches to solving the problem

Verification is a type of quality analysis and part of the development process. It consists in checking the information for correctness and accuracy. Its purpose is to identify errors, various defects, and shortcomings in order to eliminate them in time.

Currently existing verification methods are well-developed, as well as a large number of different verification models using extended decision tables, Petri nets [2], various logics, such as logics with vector semantics [3], [4], and other models. Moreover, specialized ontologies are formed to describe a variety of means and models of knowledge base verification [5]. However, there is no mechanism for interaction of the tools using these methods.

Most of the work in the field of verification focus on a particular approach or model, although the most effective approach to verification is a combination of different methods.

Therefore, knowledge base verification tools that currently exist have a number of problems such as [6]:

- dependence on the format of information representation, because of what it is necessary to spend additional time on converting information;
- the problem of impossibility to be reused, since the tools are usually created taking into account the specifics of a particular system;
- problem of lack of mechanism for interaction between means of verification and knowledge analysis;
- high role of the human in the verification process, because the most common way of verifying databases is a manual check of the database by an expert; a human acts as an administrator, making an unanimous decision, imposing their opinion on the system;
- modern tools do not take into account and do not consider the process of verification within the interaction of systems with each other.

These problems could be solved if:

- developers used a unified and convenient knowledge representation format;
- systems were created using a common methodology and were compatible with each other;
- experts thought over and implemented a mechanism that allows the system to try to make a decision

about its state and presence of problems and errors; the system may not always make the right decisions, but those should be its mistakes, not those of experts and developers.

Having analyzed the works carried out in this area, it is possible to notice a decline of interest in the verification of knowledge bases. A possible reason for this is that in the absence of a unified methodology for the development of intelligent computer systems and their knowledge bases, it is inappropriate to develop a methodology for the design of tools for knowledge base quality analysis.

Thus, at the moment, the problem of complex analysis and verification is relevant because of the lack of methodology for the design of analysis and verification tools that can effectively interact with each other to solve the problem.

III. Proposed approach

Within this article, an OSTIS Technology is used as the proposed approach. This technology is a complex of models, tools, and methods designed for the development of next-generation intelligent computer systems.

The advantages of the OSTIS Technology within the verification problem are:

- availability of a common methodology for the design of intelligent systems, which allows solving the problem of compatibility of systems during their interaction;
- all knowledge is represented in a unified form, which allows effectively processing them, reducing the cost of converting to a minimum;
- means by which contradictions are detected, analyzed, and resolved are described in the knowledge base, and their specification is represented in the system knowledge base, thereby making it easy to expand them and let the system know what tools it contains;
- absence of semantic equivalent fragments, which ensures that corrections are made locally and eliminates the need to make corrections repeatedly in different places;
- multi-agent approach, which allows considering means of analysis and verification of knowledge bases as a collective of agents, capable of interacting with each other and then making a joint decision about the state of the knowledge base.

Within the OSTIS Technology, works related to verification have already been conducted [7] but did not touch on the subject of verification in sufficient detail, in particular, there is no description of the approach to the design of verification means and a mechanism that would ensure their compatibility, while other works considered more special cases, such as verification during knowledge integration [8]. However, the verification of the knowledge base and intelligent system is not limited to this.

The OSTIS Technology uses subject domains to formalize knowledge, allowing allocating only a certain class of entities under study from the diversity of the World, focusing attention only on something specific. Ontologies are used to specify subject domains. By ontology, the semantic specification of any knowledge, which has a rather complex structure, is meant.

The *OSTIS* Technology is based on the usage of unified semantic networks with a basic set-theoretic interpretation of their elements as a method of knowledge representation. This way of knowledge representation is called an *SC-code*, and the semantic networks, represented in the *SC-code*, are called *sc-graphs* (*sc-texts*, or *texts of the SC-code*). The elements of such semantic networks are called *sc-elements* (*sc-nodes* and *sc-connectors*, which, in turn, can be *sc-arcs* or *sc-edges* depending on their orientation). The *Alphabet of the SC-code* consists of five basic elements, on the basis of which SC-code constructions of any complexity are built, including the introduction of more particular kinds of sc-elements (e.g., new concepts). The memory storing SC-code constructions is called semantic memory, or *sc-memory*.

The technology also offers several universal options for visualizing *SC-code* constructions, such as *SCg-code* (graphical variant), *SCn-code* (nonlinear hypertext variant), *SCs-code* (linear string variant).

The OSTIS Technology uses a multi-agent approach, which allows conveniently solving the problem of interaction of verification means, since in this case, the verification means should be considered as a collective of agents.

Thus, the proposed approach comes down to the development of:

- specialized *subject domain and ontology*, which would contain all the necessary knowledge about the possible types of problem fragments of the knowledge base and ways to fix them;
- an algorithm that would allow the system to identify problem fragments in itself and eliminate them, while ensuring the consistency of the means of the system;
- a specialized problem solver, containing the necessary agents to identify and eliminate the problem fragments.

IV. Analysis of knowledge base quality

The quality of the knowledge base is largely determined by the level of presence/absence of non-factors [9] in the knowledge base.

non-factor
:= [group of semantic properties that determine the quality of information stored in the memory of a cybernetic system]
= {

- *correctness/incorrectness of the information stored in the memory of a cybernetic system*
- *uniqueness/uniqueness of the information stored in the memory of a cybernetic system*
- *integrity/unintegrity of information stored in the memory of a cybernetic system*
- *compliance/incompliance of information stored in the memory of a cybernetic system*
- *reliability/unreliability of information stored in the memory of a cybernetic system*
- *accuracy/inaccuracy of information stored in the memory of a cybernetic system*
- *certainty/uncertainty of information stored in the memory of a cybernetic system*
- *determinacy/undeterminacy of information stored in the memory of a cybernetic system*

}

In this article, the focus is on such non-factors as:consistency, incompleteness, incompliance.

consistency/inconsistency of the information stored in the memory of a cybernetic system
:= [level of presence of various kinds of contradictions and, in particular, errors in the stored information]

contradiction*
:= [pair of contradictory fragments of information stored in the memory of a cybernetic system*]
⇒ *note**:
[The most common contradictory fragments of information are:
□ some regularity (rule), explicitly represented in memory
□ information fragment that does not correspond (contradict) to this regularity
]

completeness/incompleteness of information stored in the memory of a cybernetic system
:= [extent to which the information stored in the memory of a cybernetic system describes that system environment of existence and the problem-solving methods it uses (in sufficient detail) for the cybernetic system to actually be able to solve all the set of problems corresponding to it]

compliance/incompliance of information stored in the memory of a cybernetic system
:= [variety of forms and total amount of information garbage included in the information stored in the memory of a cybernetic system]

The process of creating and editing the ostis-system knowledge base is reduced to the formation of proposals by developers to edit a particular section of the knowledge base. Subsequently, these proposals are considered by the administrators of the knowledge base. The scheme of the knowledge base with proposals is shown in Figure 1.

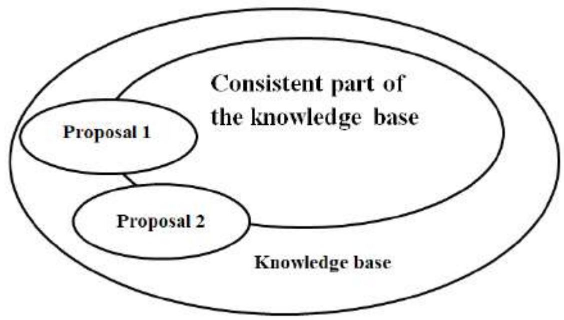

Figure 1. A knowledge base scheme with proposals made

The main reason of increasing the level of non-factors in the knowledge base is the occurence of new information or changes in existing one.

Examples of such changes are cases where:

- the user creates or modifies a fragment of the knowledge base;
- the system obtains new information by merging knowledge bases or by using tools to automatically provide the knowledge base with data from various sources;
- changes occur in the system during the work of the agents.

In the proposed approach, the structures in the knowledge base, which increase the level of non-factors in the system, must be localized and described, so that they can be fixed in the future. It is important that the mechanism of localization and description itself should be universal. This implies that the processing of such structures should not depend on their type.

V. Subject domain of problem structures

Therefore, there is a need to develop a specialized Subject domain, which would describe this kind of structures. For this purpose, the concept of a problem structure is introduced.

problem structure
:= [structure describing a unsatisfactory knowledge base fragment]
⇐ *combination**:
{• *incorrect structure*

:= [structure containing fragments that contradict any rules or patterns described in the knowledge base]
- *structure describing incompleteness in the knowledge base*
  := [structure in which there is incompleteness (i.e., there are a number of information holes)]
  ⇒ *note**:
  [A structure describing an incompleteness in the knowledge base is a structure containing a fragment of the knowledge base in which some information is missing that is necessary (or at least desirable) for an unambiguous and complete understanding of the meaning of the fragment.]
- *information garbage*
  := [structure whose removal would not significantly complicate the system]
  := [structure containing a fragment of the knowledge base that, for whatever reason, has become unnecessary and requires deletion]

}

In addition to the problem structure itself, the Subject domain addresses its more special cases and related concepts.

incorrect structure
⇐ *inclusion**:
{• *duplication of system identifiers*
• *mismatch of connective elements to relation domains*
• *cycle in relation order*
• *structure that contradicts the singularity property*
}

structure describing incompleteness in the knowledge base
⇐ *inclusion**:
{• *no maximum class of research subjects of the subject domain specified*
• *for the entity, the system identifier is specified, but no main identifiers are specified for all external languages*
• *no relation domains specified*
• *concept is not associated with any subject domain*
}

VI. Algorithm for knowledge base fragment verification

To ensure the compatibility of verification tools it is required to develop an algorithm that allows eliminating problem structures in the knowledge base of an intelligent system in a unified way.

The process of verification and correction of the structure in this algorithm should be considered as an iterative process, in which, after proposing any changes, the following should be checked:

- if it has ceased to be problematic;
- whether changes have created new problem structures.

If it is not possible to propose changes to fix it, the structure must be reverted to its original state.

Taking into account the features mentioned above, the general algorithm for working with problem structures in the knowledge base should include the following steps:

- identifying problem structures;
- fixing the state of the problem structure;
- proposing changes to correct the problem structure;
- applying the proposed changes;
- checking the changed structure;
- rolling back in case of impossibility to correct the structure;
- fixing the non-corrected structure.

VII. Problem solver of knowledge base verification

The tools for quality analysis within the OSTIS Technology are agents. At a minimum, agent interactions should include the ability for an agent to initiate other agents and to access the results of their work.

The organization of the mechanism for interaction of the corresponding means of the verification process can be carried out by the corresponding agents. An example of such a system is a system in which an agent monitors the state of a knowledge base and, in the case of new information arrival, initiates the appropriate verification means. The initiated means will analyze the received information and record its state in the base. If necessary, the means of correction will propose the appropriate changes and apply them.

The system of such verification tools will be a problem solver, an example of the possible structure of which is represented below.

Problem solver of means for identifying and eliminating contradictions
⇐ *decomposition of an abstract sc-agent**:
{• *non-atomic agent for contradiction detection*
  := [Set of agents providing contradiction retrieval and fixation in the structure]

- *Non-atomic agent for contradiction elimination*
  := [Set of agents creating proposals to fix contradictions]
- *Agent applying proposals to fix contradictions*
- *Non-atomic agent for structure verification*

}

VIII. Version control model

For the coordinated interaction of agents of an intelligent system, only their specifications are not enough, but also a format for describing the structures with which they will work is required. Such a format is a version control model, which must take into account the changes made to the structure in the process of fixing it, while being convenient for the work of the agents.

It is supposed that correction agents do not apply the changes immediately but only make proposals for changes.

Thus, the corrected structure is only a modified version of the original structure.

This is done, among other things, to avoid providing the knowledge base with copies of the initial structure.

An example of a version control model [10] that can be used is shown in Figure 2.

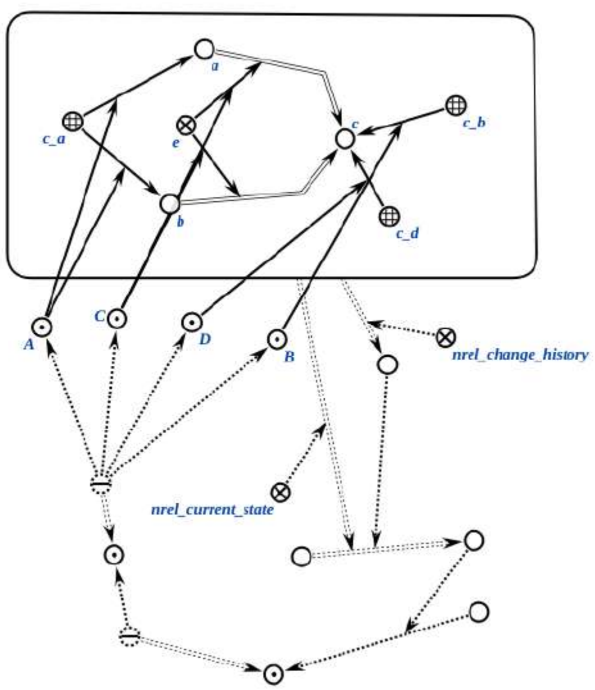

Figure 2. An example of versioning model

IX. Description of problem solver agents

Verification agents can be divided into:

- agents detecting problem structures;
- agents correcting/removing problem structures.

The stage of problem structure detection implies the initiation of agents for problem structure detection. This can be carried out by some agent which has knowledge of what search agents are in the system, or agents can be initiated, for example, when new information is added to the knowledge base. The problem of agents for problem structure detection is to find in the knowledge base the fragments causing problems, describe, and record information about them, so that later the correction agents could make appropriate changes to them. The result of the work of such agents in the general case is the immersion of the problem fragment in the structure belonging to the corresponding classes of problem structures. An example of the result of the agent for problem structure detection is shown in Figure 3.

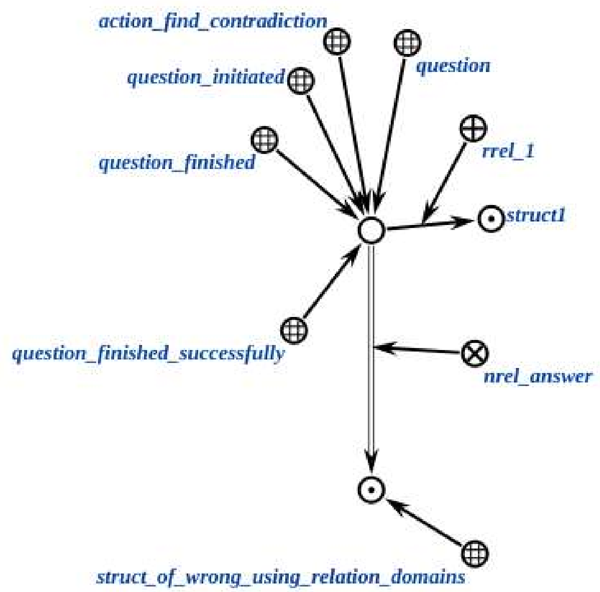

Figure 3. An example of the result of the agent for problem structure detection

The stage of fixing the state of the problem structure implies the usage of the version control model to fix the state of the structure. For the initial state, the elements belonging to the structure must be marked.

The stage of proposing changes to fix the structure involves the work of the agents forming proposals to change the structure. These agents can be called either by the supra-agent-coordinator on the basis of what kinds of problem structures they can fix, or they can respond to an event themselves, for example, on adding the belonging of the structure to the appropriate class of problem structures. The result of the work of such agents is sets of elements that should be removed or added to the structure so that it ceases to be problematic.

An example of the result of the agent for structure change proposal is shown in Figure 4

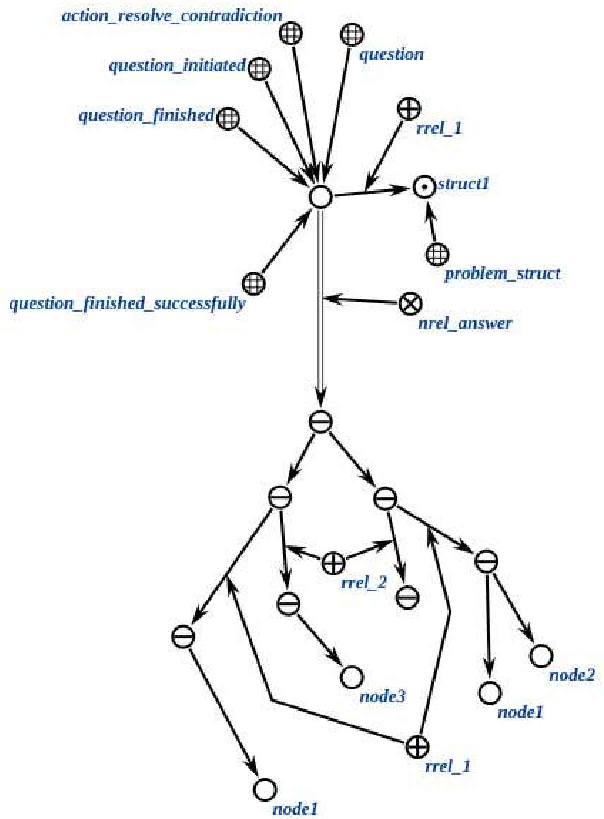

Figure 4. An example of the result of the agent for structure change proposal

Further, the stage of applying the changes proposed by the agents takes place, after which the state of the knowledge base is checked. This is necessary to make sure that:

- structure is fixed and is no longer problematic;
- fixes over the problem structure have not generated new problem structures that the system cannot fix.

In the case where the system is unable to propose changes capable of fixing the problem fragment, the structure should be returned to its original state. It is important to fix the fact that the system at the moment is not able to fix the problem structure on its own. This is necessary to avoid further unnecessary attempts to fix this structure, as well as to indicate that the solution to this problem may require the help of an expert or developer.

X. Conclusion

In the article, an approach to the design of tools for analyzing the quality of knowledge bases of next-generation intelligent computer systems is proposed. It is based on the usage of a multi-agent approach to ensure the consistency of tools for knowledge base quality analysis.

The subject domain of problem structures is allocated, as well as the algorithm of interaction between agents for verification of knowledge bases, which allows describing the problem fragments of the knowledge base and designing tools that can consistently analyze and improve the quality of the knowledge base.

The obtained results allow increasing the efficiency of the development of tools for analyzing the quality of knowledge bases, in particular, the means of verification, which ultimately allows improving the quality of the knowledge bases themselves.

References

[1] C. Gavrilova, *Gavrilova T.A. Knowledge Bases of Intelligent Systems / T.A. Gavrilova, V.F. Khoroshevsky. - SPb: Peter, 2000*, 2000.

[2] L. Martin and A. Romanovsky, “Stochastic activity networks for the verification of knowledge bases,” 08 2017, pp. 37–44.

[3] L. Arshinskiy, A. Ermakov, and M. Nitezhuk, “Logic with vector semantic as a means of knowledge bases verification,” *Ontology of designing*, vol. 9, pp. 111–122, 12 2019.

[4] ——, “Complex verification of rule-based knowledge bases using VTF-logic,” *Ontology of designing*, vol. 10, pp. 112–120, 04 2020.

[5] G. Rybina and V. Smirnov, “Methods and algorithms of knowledge base verification in integrated expert systems / G.V. Rybina, V.V. Smirnov // Izvestia RAN. Theory and Control Systems 2007,” vol. 4, pp. 91–102, 11 2005.

[6] D. Zhang, “Knowledge base verification: issues and approaches,” 10 2022.

[7] I. Davydenko, “Semantic models, method and tools of knowledge bases coordinated development based on reusable components,” in *Open semantic technologies for intelligent systems*, V. Golenkov, Ed., BSUIR. Minsk , BSUIR, 2018, pp. 99–118.

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

[9] A. Narin’jani, “Ne-faktory: kratkoe vvedenie [Non-factors: a brief introduction],” *Novosti iskusstvennogo intellekta [Artificial intelligence news]*, no. 2, pp. 52–63, 2004.

[10] N. Zotov and K. Bantsevich, “Principi obespechenya versionnosti fragmentov baz znaniy intellectalnich sistem [Principles of providing versioning of fragments of knowledge bases of intelligent systems],” *Information technology and management : proceedings of the 58th scientific conference of graduate students, undergraduates and students*, p. 69, 2022.

Методика и средства проектирования и анализа качества баз знаний интеллектуальных компьютерных систем нового поколения

Бутрин С.В.

В работе рассмотрен подход к проектированию средств анализа качества баз знаний интеллектуальных компьютерных систем нового поколения. Он основан на использовании многоагентного подхода для обеспечения согласованности средств анализа качества баз знаний.

Полученные результаты позволяют повысить эффективность разработки средств анализа качества баз знаний, в частности, средств верификации, что в конечном итоге позволяет повысить качество самих баз знаний.

Received 01.11.2022