taking into account the previously mentioned entities and understanding how they relate to the current topic of conversation. This makes the dialog with the system more productive and natural, similar to talking to a person who remembers all the details to solve a problem.

III . Example of application of the results obtained

An example of realization of the proposed approach is an intelligent geometry learning system consisting of three components: a KB, a problem solver and a UI.

Below is a decomposition of the *Geometry Intelligent System Problem Solver*.It consists of a search module — agents that search for constructs in the KB and a computational problem solver, i.e., agents that implement algorithms for solving geometry problems.

Solver of intelligent system problems in geometry
=> decomposition of an abstract sc-agent:*
{ •Abstract non-atomic sc-agent search agent
•Computational Problem Solver
=> decomposition of an abstract sc-agent:*
•Abstract sc-agent for interpreting arithmetic expressions
•Abstract non-atomic sc-agent for interpreting logical rules interpreting logical rules
{•Abstract sc-agent of constructing a strategy for finding a solution to a problem in width
}
}

Next, a decomposition of Abstract non-atomic sc-agent searchis presented. It consists of a necessary set of scagents that can be used in solving specific problems. For example, if a theorem proving problem needs to be solved, it is appropriate to use Abstract sc-agent for searching axioms of a given ontology or Abstract sc-agent for searching theorems of a given ontology.

Abstract non-atomic sc-agent search agent
=> decomposition of an abstract sc-agent:*
{•Abstract sc-agent for finding an annotation for a given section
• Abstract sc-agent for searching axioms of a given ontology
• Abstract sc-agent for searching theorems of a given ontology
• Abstract sc-agent for finding direct links between two objects
• Abstract sc-agent for searching concepts through which a given concept is defined
• Abstract sc-agent for searching the scope of a relation definition
• Abstract sc-agent to find a definition or explanation for a given object
• *Abstract sc-agent for finding examples for a given concept*
• *Abstract sc-agent for finding a formal statement record for a given statement sign*
• *Abstract sc-agent for finding illustrations for a given object*
• *Abstract sc-agent of finding key sc-elements for a given subject area*
• *An abstract sc-agent searches for concepts that are defined on the basis of a given*
• *Abstract sc-agent search for all constructs isomorphic to a given pattern*
• *Abstract sc-agent for finding the sc-text of a proof for a given assertion*
• *Abstract sc-agent for searching relations defined on a concept*
• *Abstract sc-agent for searching sc-text of condition and problem solution*
• *Abstract sc-agent for searching statements about an object*
}

The following is a decomposition of the Abstract nonatomic sc-agent problem solver.

Abstract non-atomic sc-agent problem solving agent=> *decomposition of an abstract sc-agent*:*
• *Abstract sc-agent for searching the value of an unknown quantity*
• *Abstract sc-agent for verifying the truth of an assertion*
• *Abstract sc-agent application of problem-solving strategies*
• *Abstract sc-agent of performing logical inference*
• *Abstract non-atomic sc-agent for calculating mathematical expressions=>decomposition of an abstract sc-agent*:*
• *Abstract sc-agent for coordinating the calculation of mathematical expressions*
• *Abstract sc-agent for degree expansion, root extraction and finding the natural logarithm*
• *Abstract sc-agent for addition and subtraction of quantities and numbers*
• *Abstract sc-agent of product and division of quantities and numbers*
• *Abstract sc-agent for comparing quantities and numbers*
• *Abstract sc-agent for calculating trigonometric expressions*

Abstract sc-agent for coordinating the calculation of mathematical expressions takes a formula as a parameter. An example is shown below in Fig. 4.

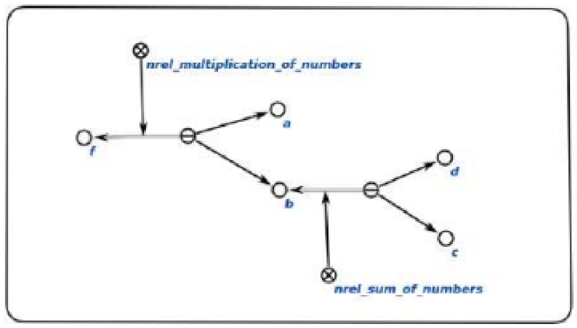

Рис. 1: Example formula as input parameter of sc-agent

A formula is represented as an sc-structure, which contains sc-bindings of mathematical operation relations and sc-nodes, which are signs of numbers or variables whose value is known or to be calculated. In this example, the formula consists of:

- variables:
  -a,
  -b,
  -c,
  -d,
  -f,
- relations:
  – nrel_sum_of_numbers,
  – nrel_multiplication_of_numbers,
- arcs and edges.
  The abstract sc-agent of coordination of calculation of mathematical expressions of formula processing after initiation searches for all sc-edges of relations of arithmetic operations and forms a structure for calling the operation calculation agent with the parameter of the scedges connecting the node of the relation and the binary arc of the basic kind. In turn, each of the sc-agents for operation computation checks whether it can compute an operation of the given type. If it can, it computes the operation and creates an sc-node with the answer, otherwise it does not continue its work. Thus, the abstract sc-agent coordinating the computation of mathematical expressions processing formula does not know in advance which agent to call specifically. All agents react to the initiated action by checking the input parameters as the initial condition of the problem. The values of the variables in the formula can be specified in advance, or they can be found in the course of solving the problem. Below are the steps of calculating the values of the variables in the formula, if the values of a,d,c are known in advance (otherwise the formula would not have a specific value).Since the values of a and b must be known to compute f , the sc-agent checks if their values are known. Since the value is known only for a , the agent will generate a

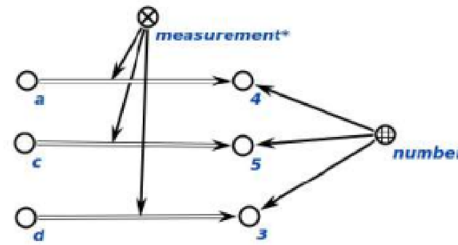

Рис. 1: Example formula as input parameter for sc-agent

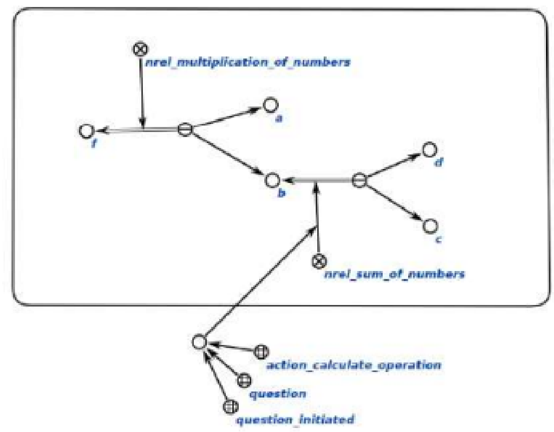

Рис. 2: Example of operation calculation agent initiation

structure to initiate the agent to compute an arithmetic operation, after which the value of b will be known. An example of such a structure for initiating the scagent is given below in 2
Once completed, the agent will create the following construct in the KB. 7
Thus, if all initiated agents are successfully executed, the value of the value of f
This agent can be used to calculate the values of area, perimeter, etc. using predetermined formulas.

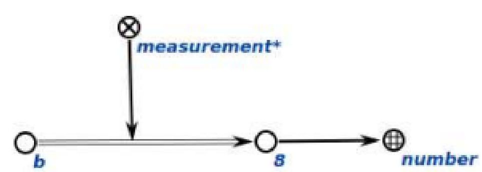

Рис. 3: Example of the result of the operation calculation agent execution

IV. Conclusion

The paper proposes an approach to the development of methods and means of constructing plans for problem solving in ISs, which allows us to improve the accuracy of answers, as well as to overcome the shortcomings of modern ISs.
The proposed model allows us to consider the developed problem solver at different levels of detail, which provides the possibility of step-by-step design of solvers, as well as their modifiability. Classification and specification of actions, problems are specified.
The architecture is considered and the IS itself, realizing the proposed approach, is described. The obtained results will allow to increase the efficiency of designing ISs and means of automating the development of such systems, as well as to provide an opportunity not only for the developer, but also for the IS to automatically supplement the system with new knowledge and skills.

Acknowledgment

The authors would like to thnk the research groups of the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics.

[1] I. Z. Batyrshin,Fuzzy hybrid systems. Theory and practice,H. G. Y. M., Ed. Physmatlit, 2007.
[2] T. Pratt, Programming languages: development and realization : transl. from English / T. Pratt, M. Zelkovits, A. Matrosov, Ed. SPb. Peter Print, 2002.
[3] L. A. Gladkov, Genetic algorithms : textbook, V. M. K. n. r. L. A. Gladkov, V. V. Kureichik and supplementary ed. M., Eds. Physmatlit, 2006.
[4] V. V. Emelyanov, Theory and practice of evolutionary modeling, V. M. K. V. V. Emelyanov, V. V. Kureichik, Ed. Physmatlit, 2003.
[5] M. B. Berkinblit, Neural networks : an experimental textbook, M. B. Berkinblit, Ed. MIROS : All-Russian extramural multidisciplinary school of the Russian Academy of Sciences., 1993.
[6] V. A. Golovko, Neural networks: training, organization, and applications : a textbook, V. A. Golovko, Ed. Journal. "Radiotekhnika 2001.
[7] A. N. Gorban, Neural networks on a personal computer, D. A. R. A. N. Gorban, Ed. Novosibirsk : Nauka, 1996.
[8] V. N. Vagin, Reliable and plausible inference in intelligent systems, 2nd ed., D. A. P. B. N. Vagina, Ed. Fizmatlit, 2008.
[9] B. A. Kulik, The logic of natural reasoning, B. A. Kulik, Ed. St. Petersburg. : Nev. dialect, 2001.
[10] D. Poya, Programming languages: design and implementation : transl. from English, M. Z. e. b. A. M. T. Pratt, Ed. Peter Print, 2002.
[11] I. Z. Batyrshin, Basic operations of fuzzy logic and their generalizations, I. Z. Batyrshin., Ed. Kazan : Fatherland, 2001.
[12] N. P. Demenkov, Fuzzy control in technical systems : textbook, N. P. Demenkov, Ed. Moscow State Technical University Publishing House, 2005.
[13] D. A. Pospelov, Modeling reasoning: experience in analyzing thought acts. Radio and communications, 1989.
[14] R. A. Reiter, A logic for default reasoning, R. Reiter, Ed. Artificial Intelligence., 1980, vol. 13, no. 13.
[15] A. P. Eremeev, Construction of ternary logic-based decision functions in decision-making systems under uncertainty, A. P. Eremeev, Ed. Famous Russian Academy of Sciences. Theory and systems of management., 1997.
[16] A. Tamkin, M. Brundage, J. Clark, and D. Ganguli, "Understanding the capabilities, limitations, and societal impact of large language models," CoRR, vol. abs/2102.02503, 2021. [Online]. Available: https://arxiv.org/abs/2102.02503
[17] K. Bantsevich, M. Kovalev, and N. Malinovskaya, "Integration of large language models with knowledge bases of intelligent systems," Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], pp. pp 213–219, 2023.
[18] V. V. Golenkov and N. A. Gulyakina, "Next-generation intelligent computer systems and technology of complex support of their life cycle," Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], pp. 27–40, 2022.
[19] M. V. Kovalev, "Convergence and integration of artificial neural networks with knowledge bases in next-generation intelligent computer systems," Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems], pp. 173–186, 2022.
[20] (2023, March) Opinion | Noam Chomsky: The False Promise of ChatGPT. [Online]. Available: https://www.nytimes.com/2023/03/08/opinion/noam-chomsky-chatgpt-ai.html
[21] V. Golenkov, N. Gulyakina, and D. Shunkevich, Open technology for ontological design, production and operation of semantically compatible hybrid intelligent computer systems, G. V.V., Ed. Minsk: Bestprint, 2023.
[22] A. N. Averkin, A Comprehensive Dictionary of Artificial Intelligence, H. G. Yarushina, Ed. Radio and communications, 1992.
[23] "Github:scl-machine [electronic resource]." [Online]. Available: https://github.com/ostis-ai/scl-machine

МЕТОДЫ И СРЕДСТВА ПОСТРОЕНИЯ ПЛАНОВ РЕШЕНИЯ ЗАДАЧ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ НА ПРИМЕРЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПО ГЕОМЕТРИИ

Малиновская Н. В., Макаренко А. И.

В данной работе предлагается подход к разработке методов и средств построения планов решения задач в интеллектуальных системах на примере интеллектуальной системы по геометрии. Описанный подход направлен на повышение точности ответов засчет возможности декомпозиции задач на более простые, а также направлен на преодоление недостатков современных интеллектуальных систем. Описана интеллектуальная система, реализующая предлагаемый подход. Received 01.04.2024