https://www.proconference.org/index.php/usc/article/view/usc11-01-001
DOI: 10.30888/2709-2267.2022-11-01-001

UDC 004.318

CHATBOT @RIBS_KARKAS_BOT IMPLEMENTS THE FUNCTIONS OF THE EXPERT SYSTEM

Burdaev V.P.
c.t.s., as.prof.
ORCID: 0000-0001-9848-9059
National Technical University "Kharkiv Polytechnic Institute",
Kharkiv, Kyrpychova 2, 61002

Abstract. *The paper presents the results of integrating the chatbot @ribs_karkas_bot with an expert system for organizing online consultation. A description of the architecture and implementation of the chatbot messenger Telegram in an expert system based on the system "KARKAS" – a tool for building models of knowledge bases. The structure of the interaction of chatbot and agents of the expert system in the online mode is considered. An example of an online consultation of an expert system in the medical subject area is given.*

Key words: *chatbots, agents, messages, knowledge base, expert system.*

Introduction.

In the business environment, the free Telegram messenger has become the corporate communication standard. This is due to the following reasons: a high degree of data encryption in it, stability of work, the ability to transfer large amounts of information, openness of the protocol, cross-platform.

On the other hand, what is very important for integrating the Telegram messenger with other applications is that the developers provide an API-based library for working with chatbots.

A bot (chatbot, interlocutor) is a program that simulates human communication based on elements of artificial intelligence. Today bots can communicate with each other to achieve their goals in other words, they can be used as agents in multi-agent systems [1].

Chatbots are usually integrated into dialog systems, such as virtual assistants, giving them the ability to communicate naturally or engage in casual conversations unrelated to areas of their core expert systems.

This paper [2] presents experience in implementing a chatbot for expert recommendation tasks. The chatbot was developed as an expert recommendation a system to help developers find the right person to contact in open source projects. Chatbot targets the Pharo software ecosystem and developer community and is

integrated with the chat service Discord, which the Pharo community uses as one of its main communication channels. The article [3] discusses the need to design interactions in networks of people and intelligent machines. The article [4] discusses the issues of integrating bots into chat platforms for developers. This article [5] explores the concept of a "repair bot" and introduces Repairnator. The Repairnator bot is a standalone agent that continuously monitors test failures, reproduces bugs, and runs program repair tools for every bug that is reproduced. If a patch is found, the Repairnator bot informs the developers about it.

Main text.

The paradigm of integrating chatbots to work with expert systems is now becoming increasingly important [2].

Using the API Telegram library, the bot @ribs_karkas_bot was created for online user consultation with a tool for creating knowledge bases with the "KARKAS" system [6 – 9].

The @ribs_karkas_bot allows for online consultation with the following prototypes of expert system: the "RIBS" system [3 – 6] is designed to determine the risk of coronary heart disease (RCHD) in a practically healthy person. The relevance of the development of the system lies in the fact that at present in medicine there is a clearly expressed process of transition to the concept of RCHD prevention, then there are concepts of risk factors associated with the lifestyle of a particular patient. The purpose of the system is to recognize the presence of risk factors for coronary heart disease with an emphasis on the patient's individual lifestyle, using the knowledge of experts. behavior, the degree of social and psychological support, the level of physical activity, the degree of adequacy of rest.

The content of the @ribs_karkas_bot command /help is shown in (Figure 1).

To build a knowledge base filtering it is enough to specify a chain of rules of a hierarchical functional system to achieve the main goal. Then, with the help of a recursive algorithm, other rules for local functional system of knowledge bases are built by generating rules during consultation with an expert.

The knowledge base of the subject area is considered as a hierarchical functional system, in which the result has an organizing effect on all stages of ontology formation. Classes and connections between them can be considered as a logical construction of a hierarchical functional system. The hierarchical structure of the database allows the inference machine to achieve a local goal at each level of the hierarchy and, accordingly, a global goal.

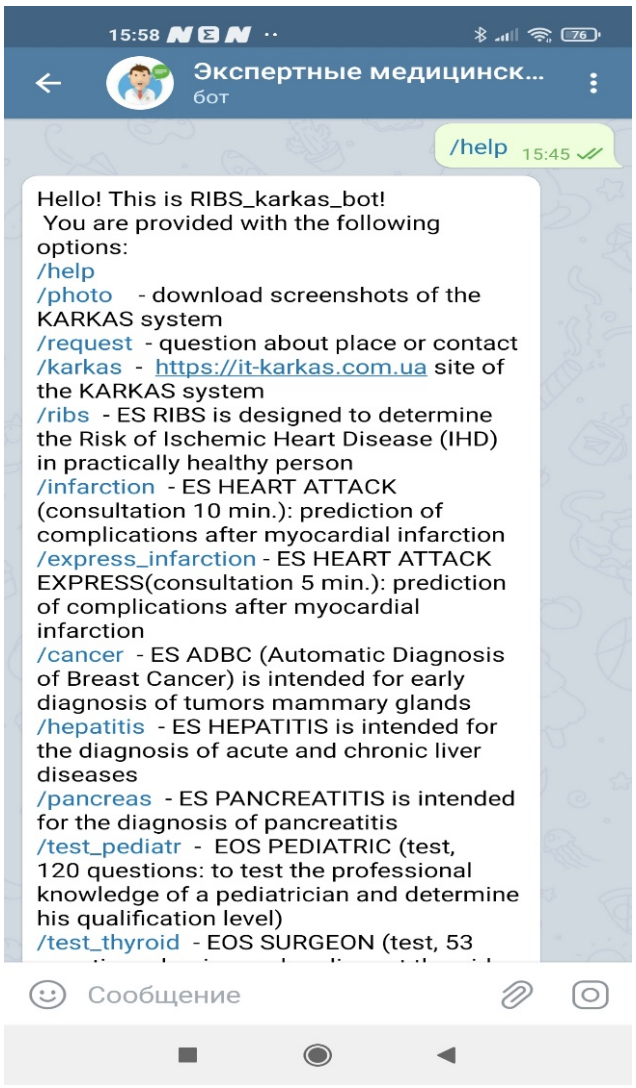

Figure 1 - Type of command /help chatbot @ribs_karkas_bot

Authoring

The inference engine of the "KARKAS" system uses a hierarchical functional system during the consultation with the user. The user can choose different modes of operation of the inference machine: the use of direct inference, inverse inference, indirect inference, Bayesian formula, table of criteria, when the consequent product is a list of parameters.

Consultation and dialogue agents exchange messages with each other to perform the following operations:

1. Pressing: buttons, check boxes, radio buttons.

2. Transmission and reception of messages between visual objects on the form.

Thus, the above modules perform the functions of agents and in this sense, the implemented chatbot @ribs_karkas_bot in the system "KARKAS" can be considered as a multi-agent system.

The algorithm of the chatbot @ribs_karkas_bot consists of the following steps:

Step 1. Activate the chatbot @ribs_karkas_bot in the Telegram messenger.

Step 2. Select the commands: /help or /start, then, the /ribs command calls the expert of system prototype to select the risk of coronary heart disease.

Step 3. The bot launches the consulting agent of the "KARKAS" system.

Step 4. The inference engine of the "KARKAS" system is activated.

Step 5. The hierarchical functional system is formed for dialogue with the user.

Step 6. The dialog agent is activated, which sends the bot a message with the text of the question and answers. The bot receives the message as a JSON object, performs its parsing, displays the message in the chat and waits for the user's response.

Step 7. The user in the chatbot selects or enters the answer. The bot sends the response to inference engine of the expert system.

Step 8. The expert system consulting agent receives the message and transmits it to the inference engine, which transmits the message to the dialogue agent. The purpose of the consultation is specified, based on a hierarchical functional system, during the dialogue with the user.

Step 9. The iterative consultation process continues until the inference engine receives the result from the expert system. The user can terminate the consultation with the /quit command at any time.

Summary and conclusions.

Have been considered the @ribs_karkas_bot chatbot is integrated into the "KARKAS" system.

Were received the @ribs_karkas_bot which allows online consultation with the "RIBS" expert system to determine the risk factor for coronary heart disease.

References:

1. S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 4nd. ed., 2020.

2. Jhonny Cerezo;Juraj Kubelka;Romain Robbes;Alexandre Bergel, Building an Expert Recommender Chatbot, 2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE), pp. 59 – 63.

3. A. Folstad and P. B. Brandtzæg. Chatbots and the new world of hci. Interactions, 24(4),2017, pp. 38–42.

4. C. Lebeuf, M.-A. Storey, and A. Zagalsky. Software bots. IEEE Software, 35(1), 2018, pp. 18–23.

5. M.-A. Storey and A. Zagalsky. Disrupting developer productivity one bot at a time. In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 928–931. ACM, 2016.

6. G.S. Osipov, Intelligent dynamic systems, Scientific and Technical Information Processing volume 37, 2010, pp. 259–264.

7. V.P. Burdaev, Complexity of dynamical systems, Saarbrucken: Lap Lambert Academic Publishing, 2015.

8. J.C. Giarratano, G.D. Riley, Expert Systems: Principles and Programming, 4th. ed., 2007.

9. Computer system "KARKAS", 2012,URL: https://it-karkas.com.ua.

sent: 07.06.2022