V. P. Burdaev

INTEGRATION CHAT-BOTA @RIBS_karkas_bot WITH EXPERT SYSTEM

Анотація. *В роботі представлені результати інтегрування чат-бота @ribs_karkas_bot з експертною системою для організації консультування в режимі онлайн. Дано опис архітектури і реалізація імплементації чат-бота месенджера TELEGRAM в експертну систему на базі системи "КАРКАС" - інструментальний засіб для побудови моделей баз знань. Розглянуто структуру взаємодії чат-бота і агентів експертної системи в онлайн режимі. Наведено приклад онлайн консультації експертної системи в медичній предметної області.*

Ключові слова: *чат-боти, агенти, повідомлення, база знань, експертна система.*

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 chat bot 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.*

Keywords: *chatbots, agents, messages, knowledge base, expert system.*

Introduction and formulation of the problem. In the business environment, the TELEGRAM free manager has become the corporate communications standard. This is due to the following reasons: a high degree of data encryption in it, work stability, the ability to transfer large amounts of information, open protocol, cross-platform.

On the other hand, what is very important for integrating the TELEGRAM manager with other applications is that developers provide a library based on an API for working with chat bots.

A bot (chat bot, interlocutor) is an application 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 to as agents in multi-agent systems [1].

One of the first programs implementing the chat bot concept was the ELIZA program, which imitated the behavior of a psychotherapist during the initial interview of a patient [1]. The idea of implementing this program was to find key words or combinations in the text of communication, in order to ask a question to maintain a dialogue with the interlocutor. If the keyword is found in the database, then the question to the interlocutor was asked in accordance with a pre-prepared question template or the approval of the interlocutor was transformed into a question. If the word combination was not found, the program asked the interlocutor with general questions, for example, "Why do you think so?"

There are several strategies for implementing this dialogue:

1. The question is select from the list of questions, according to the higher frequency of use of the question in the subject area.

2. The bot collects questions and phrases used by the interlocutors, thus learning and increasing their content of the subject area.

3. A syntactic approach based on the grammatical analysis of an interlocutor's phrase and provided with rules of the "if-that" kind.

Naturally, the lack of such a logical dialogue between the bot and the interlocutor became the disadvantage of such communication. The interlocutor appeared illusion that the bot understands it, although in fact it is not. In other words, the chat bot lacks the implementation of interaction with the output machine, as is widely used in expert and expert training systems [1].

In recent decades, with the advent of the smartphone, the popularity of the concept of artificial intelligence in relation to messaging applications has increased. The global chat bots market is expected to grow in the coming years.

One of the main advantages of chat bots in customer service is that interlocutors can freely ask questions that they would not ask a customer service representative or company manager. In addition, the bot is able to instantly answer questions.

Chat bots are usually integrated into interactive systems, for example, virtual assistants, which enables them to naturally communicate or participate in casual conversations that are not related to the areas of their main expert systems.

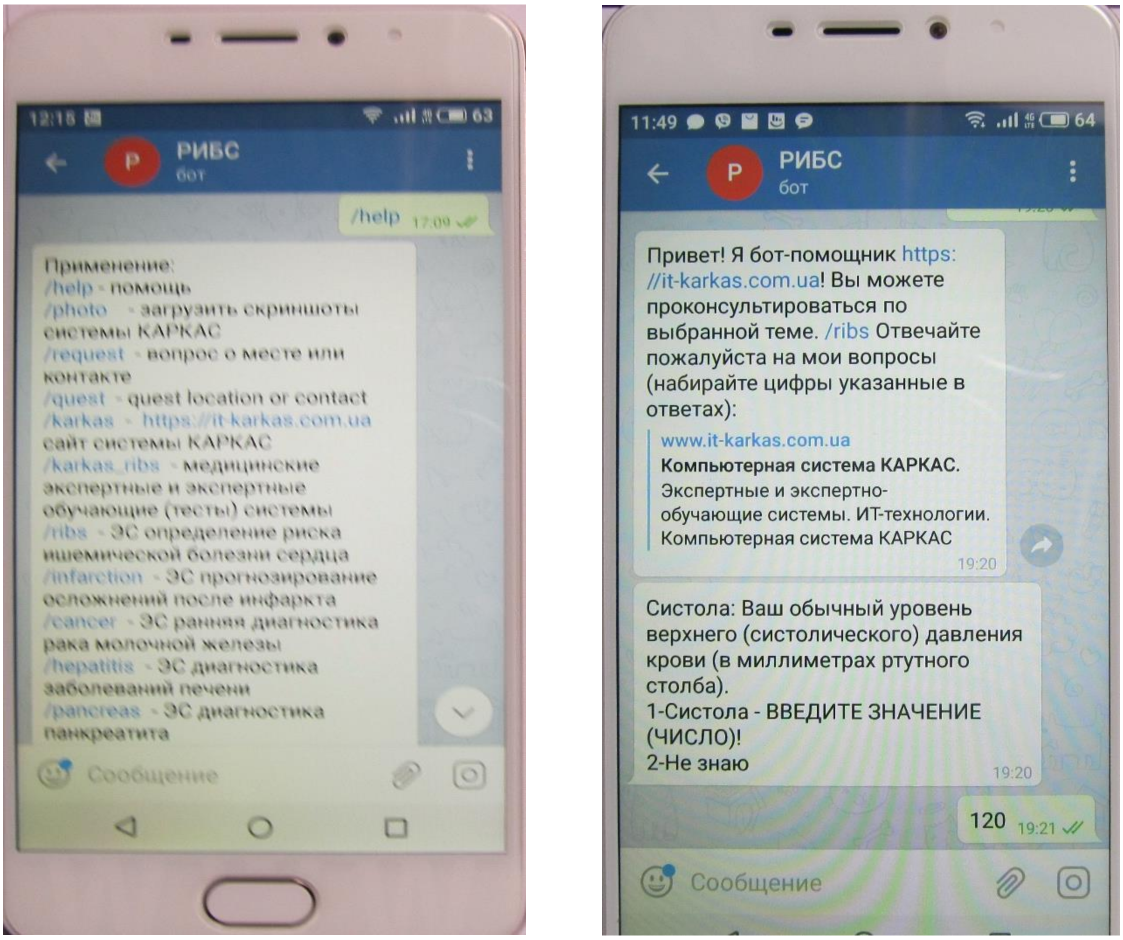
In most cases, chat bots use messaging applications to communicate with customers. A person can type or ask a question, and the chat bot will respond with the correct information. Depending on the situation, many chat bots can learn from what the client says to personalize the interaction and build up the previous interaction.

A chat bot can be viewed as a question-answer system (QA-system) with machine learning elements, namely, natural language parsing functions, an inference machine, and a communication module with external applications. The actual problem for chat bots of QA-systems is the creation of an inference machine that determines the relevance of knowledge to a given question.

The paradigm of integrating chat bots to work with expert systems is now becoming increasingly relevant [2].

Using the Telegram API library, the @Ribs_karkas_bot bot was created for online consultation of the user with a tool for creating knowledge bases with the ″KARKAS″ system [3 – 5].

Chat bots of the TELEGRAM messenger, as interlocutors, when working with the "KARKAS" system, provide more opportunities for mobile consultation with the expert system via a smartphone, which, for example, is important for making effective decisions in different subject areas such as medicine, ecology, business. In other words, you can now send a text message to the @Ribs_karkas_bot bot (a bot to determine the risk of coronary heart disease) and get instantly necessary information, that is, carry out a real-time consultation. The contents of the chatbot / help command are shown in fig. 1.



/help 17:09

Применение:
/help - помощь
/photo - загрузить скриншоты системы КАРКАС
/request - вопрос о месте или контакте
/quest - quest location or contact
/karkas - https://it-karkas.com.ua сайт системы КАРКАС
/karkas_ribs - медицинские экспертные и экспертные обучающие (тесты) системы
/ribs - ЭС определение риска ишемической болезни сердца
/infarction - ЭС прогнозирование осложнений после инфаркта
/cancer - ЭС ранняя диагностика рака молочной железы
/hepatitis - ЭС диагностика заболеваний печени
/pancreas - ЭС диагностика панкреатита

Привет! Я бот-помощник https://it-karkas.com.ua! Вы можете проконсультироваться по выбранной теме. /ribs Отвечайте пожалуйста на мои вопросы (набирайте цифры указанные в ответах):

www.it-karkas.com.ua
Компьютерная система КАРКАС.
Экспертные и экспертно-обучающие системы. ИТ-технологии. Компьютерная система КАРКАС
19:20

Систола: Ваш обычный уровень верхнего (систолического) давления крови (в миллиметрах ртутного столба).
1-Систола - ВВЕДИТЕ ЗНАЧЕНИЕ (ЧИСЛО)!
2-Не знаю
19:20

120 19:21

Fig. 1. Type of command / help chatbot @Ribs_karkas_bot

The @Ribs_karkas_bot bot allows you to conduct an online consultation with the following prototypes of expert systems:

– the "RIBS" expert system [3 – 5] – designed to determine the risk of coronary heart disease (CHD) in a practically healthy person. The urgency of developing a system is that currently in medicine there is a clearly expressed process of transition to the concept of prevention of coronary artery disease, that is, to the concept 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 CHD with an emphasis on the individual lifestyle of the patient, using the knowledge of experts. The features of the system include the fact that with its help the patient is diagnosed: type of coronary behavior, degree of socio-psychological support, level of physical activity, degree of adequacy of rest;

– the "INFARKT" expert system [3 – 5] – helps doctors diagnose infarction patients, evaluate their condition and forecast the development of the following complications of myocardial infarction: fibrillation, acute left ventricular failure, chronic heart failure, arrhythmia, thromboembolism, myocardial rupture, recurrent infarction. The inference engine implements the Bayesian decision method. Consultation with the "INFARKT" expert system in case of 28 symptoms takes 10 minutes, and the option of express consultation (7 symptoms) takes 3 minutes;

– the "GEPATIT" expert system [3 – 5] – is designed to diagnose acute and chronic liver diseases. The system makes it possible: to recognize the cause of liver disease and, if possible, by eliminating it, obtain a therapeutic effect, specifically include drugs for the treatment of liver diseases, conduct a statistical evaluation of therapeutic interventions in patients;

– the "ADBC" expert system [3 – 5] (automatic diagnosis of breast cancer) is intended for the early diagnosis of mammary tumors. Diagnostics relies on the knowledge of an oncologist expert, who are grouped in the following sections: thermography, anamnesis, physical examinations, echotomography. The system allows to classify the following tumors: lipoma, fibroadenoma, fibrocystic mastopathy diffuse; fibrocystic mastopathy localized, DFA (diffuse), DFA (localized), mastitis.

The "KARKAS" system is a toolkit for developing prototypes of knowledge bases for expert and expert training systems. The knowledge representation is based on a hierarchical functional system that is generated by the "KARKAS" system based on production rules and frames. The output machine uses a hierarchical functional system during user consultation. The user can select various modes of operation of the output machine: use direct output, reverse output, indirect output, Bayes formula, criteria tables, when the product's consequent is a list of parameters.

The online consultation module (interlocutor) allows you to communicate with the TELEGRAM messenger with the "KARKAS" system knowledge bases via the Internet, in other words, to provide real-time consultation.

Main part. Messenger TELEGRAM has millions of active users and is the fastest messaging application.It works on all devices on mobile and desktop platforms.

The "KARKAS" system is a tool for building knowledge base models. The structure of the subject area can be diverse, for example, the choice of solutions among a specific set of options, the use of unreliable knowledge. The "KARKAS" system allows both to develop models of knowledge bases and can be used for testing and teaching students on a local network. The "KARKAS" system using chat bots: @Ribs_karkas_bot, @test_karkas_bot, @it_karkas_bot allows online consultation with users and testing students' knowledge in various subject areas: computer graphics, database technology, web analytics, business intelligence systems.

The "KARKAS" system is built according to the modular principle and for this reason it has the ability to connect other additional modules. The following main modules can be distinguished in the system architecture: loader; visual editor for developing knowledge base models; offline and online consultation agent; dialogue agent; data clustering module.

The loader performs system startup and coordinates the interaction of all modules. The visual editor of knowledge base models allows you to create for expert and expert-training systems: questions with answers, products, frames, filtering the knowledge base, the hierarchy of knowledge base classes, a hierarchical functional system. There is an opportunity to place the knowledge base on the site https://it-karkas.com.ua.

The consultation agent contains the following components: inference engine, an explanation block, a notice board, a fact base monitor.

The data clustering module allows for interactive and intelligent data classification.

To test the user's knowledge, a test generator is used, and a teacher’s monitor is used to monitor the tests. The modules of the system are presented in the form of software passive agents (their interaction is carried out using parameter arrays and through the general computer memory). The environment of their interaction is the system "KARKAS" fig. 2.

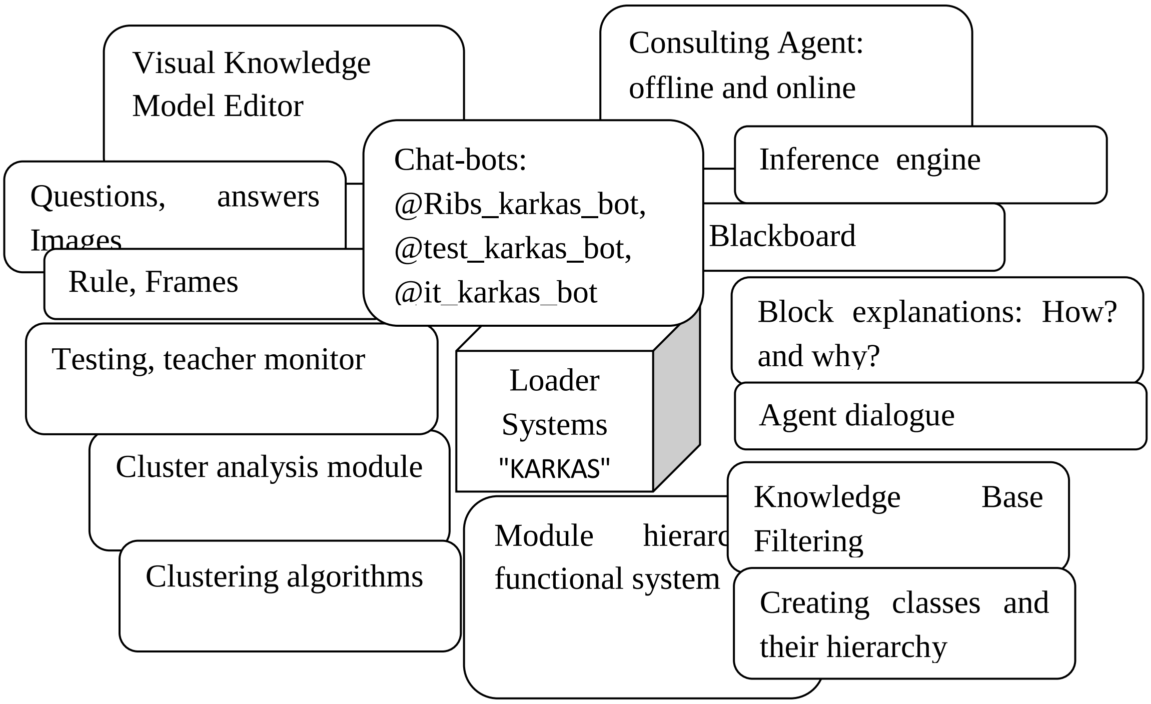

Visual Knowledge Model Editor
Consulting Agent: offline and online
Chat-bots: @Ribs_karkas_bot, @test_karkas_bot, @it_karkas_bot
Inference engine
Questions, answers Images
Blackboard
Rule, Frames
Block explanations: How? and why?
Testing, teacher monitor
Loader Systems "KARKAS"
Agent dialogue
Cluster analysis module
Knowledge Base Filtering
Module hierarc functional system
Clustering algorithms
Creating classes and their hierarchy

Fig. 2. Tag cloud system "KARKAS"

The system has two modes of use:

– the cognitive mode – a specialist who models the domain ontology and constructs a knowledge base model;

– the problem specialist's mode (user qualification is not high enough, and therefore he needs help and support from the system).

The system combines various approaches in building knowledge-based systems (table 1).

Table 1

Components of the system "KARKAS"

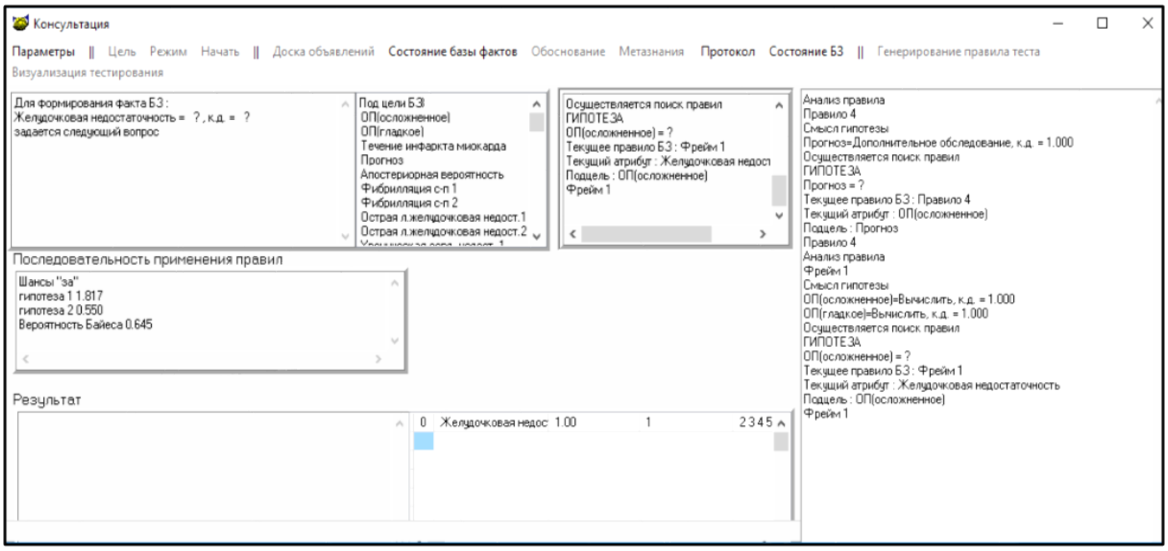
Expert Systems	Expert training systems	Multi-agent systems
Visual Knowledge Base Editor	Training	Help agents
Hierarchical functional system	Testing, intellectual tests	Chat bots: @Ribs_karkas_bot
Inference engine, Unit explanation	Teacher's monitor	@test_karkas_bot @it_karkas_bot

The components of the system are interconnected, exchange information with each other and allow to adequately display the ontology of the subject area during its operation.

The system is implemented using the Embarcadero® RAD Studio 10.2 Tokyo programming environment and contains the following files: karkasW.exe (system loader), karkased.exe (visual editor of knowledge bases), karkascon.exe (consultation agent), karkas.hlp (help), pattern.knb (knowledge base pattern).

For the MS Windows XP operating system, the following agents are used: msagent.exe (MS agent server), merlin, peddy (agent characters), and the speech recognition engine (Microsoft Speech API).

To integrate the KARKAS system with the @Ribs_karkas_bot chat bot, a consultation agent is used (Fig. 3) and a dialogue agent (Fig. 4).


Консультация
Параметры || Цель Режим Начать || Доска объявлений Состояние базы фактов Обоснование Метазнания Протокол Состояние БЗ || Генерирование правила теста
Визуализация тестирования

Для формирования факта БЗ :
Желудочковая недостаточность = ? , к.д. = ?
задается следующий вопрос

Под цели БЗ
ОП(осложненное)
ОП(гладкое)
Течение инфаркта миокарда
Прогноз
Апостериорная вероятность
Фибрилляция с-п 1
Фибрилляция с-п 2
Острая л.желудочковая недост.1
Острая л.желудочковая недост.2

Осуществляется поиск правил
ГИПОТЕЗА
ОП(осложненное) = ?
Текущее правило БЗ : Фрейм 1
Текущий атрибут : Желудочковая недост
Подцель : ОП(осложненное)
Фрейм 1

Анализ правила
Правило 4
Смысл гипотезы
Прогноз=Дополнительное обследование, к.д. = 1.000
Осуществляется поиск правил
ГИПОТЕЗА
Прогноз = ?
Текущее правило БЗ : Правило 4
Текущий атрибут : ОП(осложненное)
Подцель : Прогноз
Правило 4
Анализ правила
Фрейм 1
Смысл гипотезы
ОП(осложненное)=Вычислить, к.д. = 1.000
ОП(гладкое)=Вычислить, к.д. = 1.000
Осуществляется поиск правил
ГИПОТЕЗА
ОП(осложненное) = ?
Текущее правило БЗ : Фрейм 1
Текущий атрибут : Желудочковая недостаточность
Подцель : ОП(осложненное)
Фрейм 1

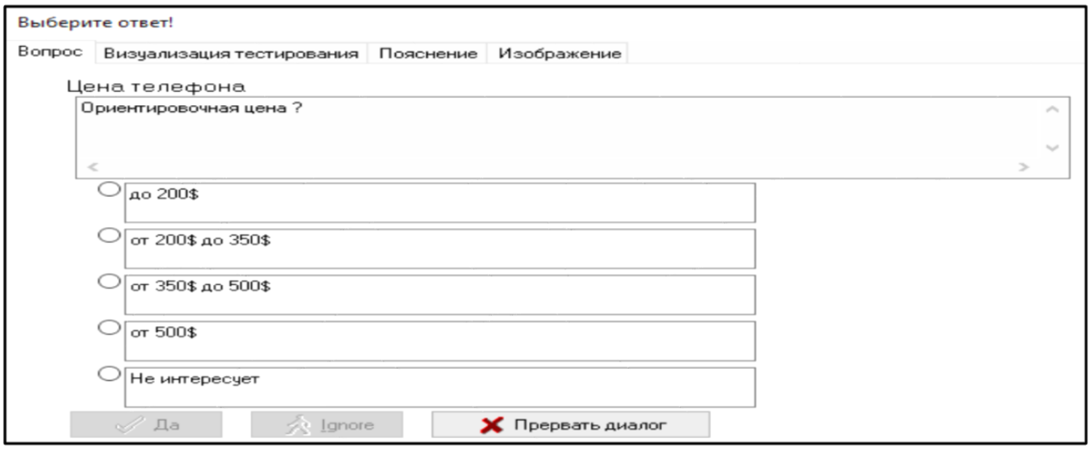
Последовательность применения правил

Шансы "за"
гипотеза 1 1.817
гипотеза 2 0.550
Вероятность Байеса 0.645

Результат

0	Желудочковая недос	1.00	1	2 3 4 5

Fig. 4. Type of form agent consultation system "KARKAS"



Выберите ответ!

Цена телефона

Ориентировочная цена ?

○ до 200$

○ от 200$ до 350$

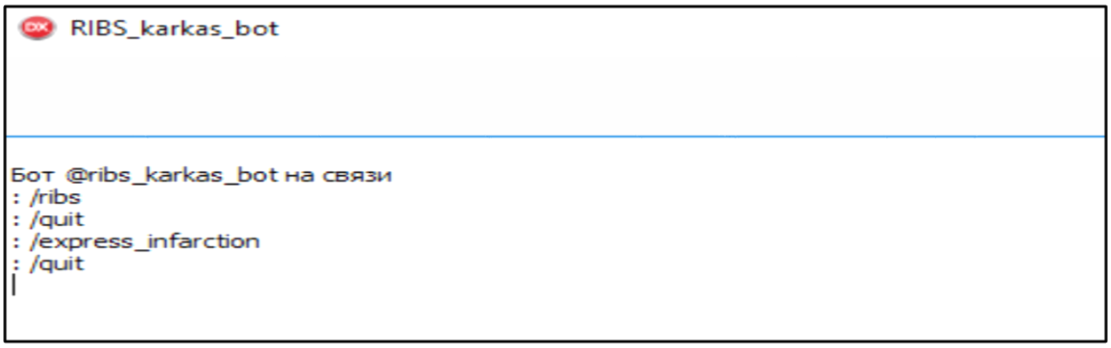
○ от 350$ до 500$

○ от 500$

○ Не интересует

Fig. 5. View of the form of the agent of the dialogue system "KARKAS"

The main form of the chat bot is as follows fig. 6.



RIBS_karkas_bot

```
Бот @ribs_karkas_bot на связи
: /ribs
: /quit
: /express_infarction
: /quit
```

Fig. 6. The main form of the chatbot loader @Ribs_karkas_bot

Integration of the chatbot with the consultation module of the KARKAS system consists in exchanging messages between them, that is, sending and receiving requests for working with TELEGRAM servers fig. 7

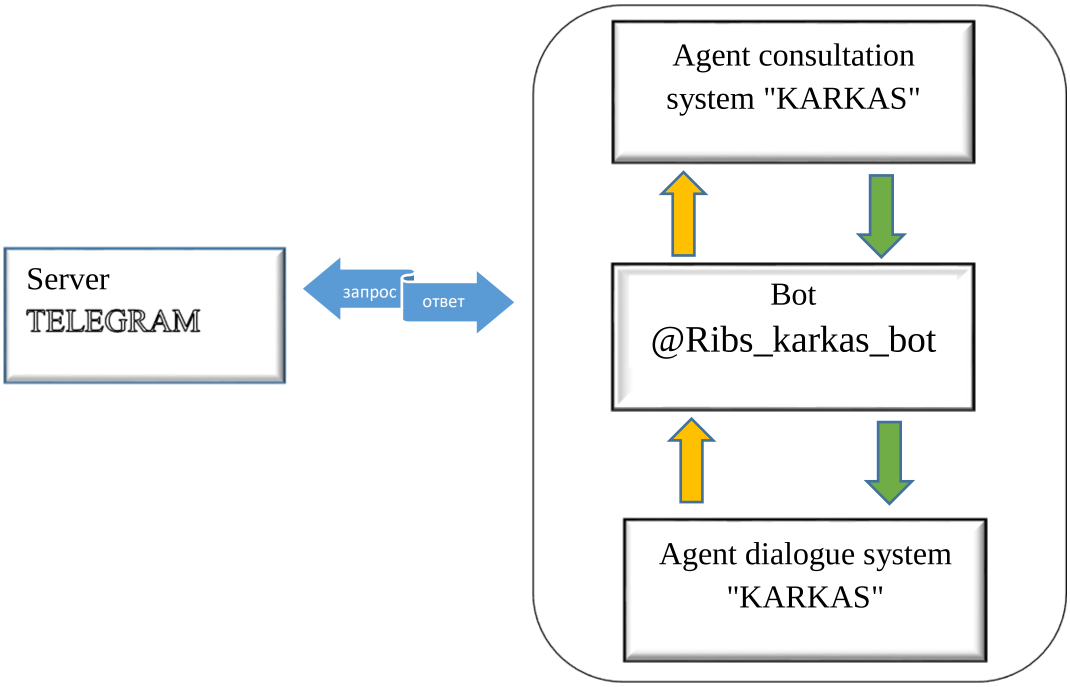

Server
TELEGRAM
запрос
ответ
Agent consultation system "KARKAS"
Bot @Ribs_karkas_bot
Agent dialogue system "KARKAS"

Fig. 7. Architecture of chatbot integration with KARKAS system

The following components are used to work with queries with TELEGRAM servers:

1. for parsing JSON objects the superobject library is used;

2. for implementation through the https protocol, the OpenSSL libraries were used: libeay32.dll and ssleay32.dll;

3. Indy 10 library and the following procedure are used to send http requests and download knowledge bases via ftp protocol from https://it-karkas.com.ua:

```
        procedure TFrmKarkasBot.BitBtnFTPloadES_KNBClick(Sender: TObject);
        begin
        FtpClient1.HostName:='it-karkas.com.ua';
        FtpClient1.UserName:='it-karkas';
        FtpClient1.PassWord:='******************';
        FtpClient1.Port:='21';
        FtpClient1.LocalFileName:=DirKarkas+'\'+SNameFileSiteKnb;
        FtpClient1.HostDirName:='/httpdocs/uploads/files/public-
folder/manual/knb/es_knb';
        FtpClient1.HostFileName := extractfilename(ftpClient1.LocalFileName);
        FtpClient1.Passive := True;
        try
        FtpClient1.Connect;
        FtpClient1.Cwd;
        FTPClient1.Binary := True;
        FtpClient1.Get;
        FtpClient1.Quit;
        except
        ShowMessage ('Ошибка загрузки файлов');
        end;
```

4. TelegAPI library is used to work with TELEGRAM servers.

Consider a number of functions and procedures of the agent consultation and dialogue system "KARKAS", which allows the bot to exchange messages with the channels available in TELEGRAM, using the TELEGRAM API and JSON via the secure HTTPS protocol.

Since the user communicates only with the chat bot, the problem arises in transmitting and receiving messages between the bot, consultation and dialogue modules in an automatic mode, i.e. without his participation. To realize this possibility, the WM_SENDBOT constant was introduced, with the help of which messages are transmitted and received for the modules to function without user intervention.

Agents communicate with each other to perform the following operations:

1. push buttons, check boxes, radio buttons;

2. sending and receiving messages between visual objects on the form.

Thus, the above modules perform the functions of agents and in this sense, the implemented chat bot @Ribs_karkas_bot in the "KARKAS" system can be considered as a multiagent system.

For the interaction of the main form of the chatbot module with the consultation and dialogue forms of the "KARKAS" system, the following procedures are required:

1. To transmit a message from the chatbot to the agents of consultation and dialogue, a variable was introduced FSendNumberMessage:

```
private
   FSendNumberMessage: Cardinal;
```

A virtual constructor has been declared and the WndProc procedure has been redefined, which receives a message from the agents and sends them a message:

```
public
constructor Create(AOwner:TComponent);ovrride;
constructor  TFrmKarkasBot.Create(AOwner:TComponent);
begin
SendNumberMessage:=RegisterWindowMessage('WM_SENDBOT');
inherited;
end;
procedure TFrmKarkasBot.WndProc(var Msg: TMessage);
begin
if Msg.Msg=FSendNumberMessage then begin
BtnSendMsgClick(Self);
end else inherited;end;
```

2. The user can interrupt the dialogue with the bot with the command: /quit, which is implemented in the following code:

```
procedure TCaseAnswer.BitBtnQuitClick(Sender: TObject);
var
  TempWnd,Wnd: THandle;
begin
   if F_Consult.Visible then
   begin
   Wnd := FindWindow(NIL, 'Выберите ответ!');
      if Wnd <> 0 then  begin
      Wnd:=FindWindowEx(Wnd, 0, 'TTabbedNotebook', nil);
      Wnd:=FindWindowEx(Wnd, 0, 'TTabPage', nil);
      Wnd:=FindWindowEx(Wnd, 0, 'TPanel', nil);
      TempWnd:=Wnd;
      Wnd:=FindWindowEx(TempWnd, 0, 'TBitBtn', 'Прервать диалог');
                     if Wnd <> 0 then  begin
                             SendMessage(Wnd, BM_CLICK, 0, 0);
                             end;
end;  end  else Bot:=0;
end;
```

3. Bot launches consultation module procedure:

```
procedure TFrmKarkasBot.BitBtnRunESClick(Sender: TObject);
begin
   BotBusy:=1;
   F_Consult.Show;
   if Bot=1 then begin F_Consult.Close;
              BotBusy:=0;//снятие блокировки для запуска бота
              end;   //если консультация загружена очистка указателей
   TrimWorkingSet;
   Bot:=0;
end;
```

4. Due to the fact that the answers to the questions of the expert system can be very long, therefore, to display them, the messenger buttons are not used, but the text is used. Therefore, chat bot offers numbered answers to the question of the expert system. If the question involves the choice of multiple answers, the answers are entered in numbers separated by commas. If, as an answer, you need to dial a number, for example, an age value, then the user in the chat dials the number that the bot sends to the inference engine to the expert system in fig. 1.

Bot interaction script with agent consultation.

1. You can activate the application (ribs_karkas_bot.exe) that starts the bot on a resource that has access to the Internet (hosting, home computer). Then in the messenger TELEGRAM start it: @Ribs_karkas_bot. Type the commands /help or /start the bot and the bot will offer to choose the commands to run expert systems, tests (fig. 1).

2. For example, when you select the /ribs command, the following operations are performed:

– knowledge base ribs.knb is loaded from https://it-karkas.com.ua;

– the consultation agent is started and the expert system inference engine is activated;

– dialogue agent is activated.

3. The result of the expert system consultation is transmitted to the bot by the broadcast protocol by the following procedure:

```
procedure ResultMainBot;
var i2 : Integer;
begin
    if (Variant_ES=1)OR (Variant_ES=2) then
            begin
            StrTemp:='';
            for i2:=0 to MemoResult.Lines.Count-1 do
            begin
            StrTemp:=StrTemp+MemoResult.Lines.Strings[i2] +' ';
            end;
            EdtMessage.Text:=StrTemp;
            case Bot of
            0 : begin end;//no bot
            1,2: begin // bot main
            NumberAns:=999;
            SendQA:=EdtMessage.Text;
            Recipients:=BSM_APPLICATIONS;
BroadcastSystemMessage(BSF_POSTMESSAGE,@Recipients,FSendNumberMessage,0,0);
                  end;
             end;//case
             end;
end;
```

Message handling by the dialog agent includes the following procedures:

1. The dialog agent sends the bot a message with the following procedure:

```
        procedure TCaseAnswer.BitBtnMainClick(Sender: TObject);
        var
          Recipients:DWORD;
          StrTemp, usage: String;
        begin
            StrTemp:='';
            for i := 0 to MemoQus.Lines.Count-1 do
            begin
             StrTemp:=StrTemp+MemoQus.Lines.Strings[i] +' ';
            end;
            //EdtMessage.Text:=AnsCurrAtribut^.Name+':                '+StrTemp+'
'+ESquestion;
            EdtMessage.Text:=AnsCurrAtribut^.Name+': '+StrTemp+ #13#10;
            usage:='';
            for  i  :=  1  to  NumberAns  do  usage:=usage+IntToStr(i)+'-'+
ESAnswer[i]+#13#10;
            usage:=EdtMessage.Text+usage;
            SendQA:=usage;
            Recipients:=BSM_APPLICATIONS;
        BroadcastSystemMessage(BSF_POSTMESSAGE,@Recipients,FSendNum
berMessage,0,0); //послать сообщение объектам управление
        end;
```

2. In order to receive messages to the dialogue agent from the bot, it is required to constantly send a request to the TELEGRAM server for GetUpdate updates. This is implemented in the following procedure, in which the parameter Timeout = 30 seconds sets how long to wait for a response from the server and the parameter Offset sets the last processed message plus the following message:

```
procedure TCaseAnswer.BitBtnParsingClick(Sender: TObject);
var
JsObject1, JsObject2: ISuperObject;
JsonArray: TSuperArray; s,h: string; i: integer;
RHttp: TIdHTTP;
RSSL: TIdSSLIOHandlerSocketOpenSSL;
Index: integer;
begin
try RHttp:=TIdHTTP.Create; except end;
try RSSL:=TIdSSLIOHandlerSocketOpenSSL.Create; except end;
RHttp.IOHandler:=RSSL;
while True do
begin
JsObject1:=SO(RHTTP.Get(BaseUrl+S_Token_bot+'/getUpdates?offset='+ IntToStr(Offset)+'&timeout=30'));
JsObject2:= JsObject1.A['result'].N[0];
Offset:=JsObject2.i['update_id'] + 1; // In case there were no messages, Offset will become equal to 1.
if Offset<>1 then
begin
if JsObject2.s['message.text']<>'' then begin
MemoBot.lines.add(JsObject2.s['message.text']);   // Message text
Index := MemoBot.Lines.Count - 1;
if Index < 0 then
raise Exception.Create('Мемо пуст')
else
begin
EdtFromBot.Text := MemoBot.Lines[Index];
Try RHttp.Disconnect; except end;
RHttp.Free; RSSL.Free; Exit; end; end; end; end; RHttp.Free; RSSL.Free;
end;
```

Thus, the @Ribs_karkas_bot chat bot algorithm consists of the following steps:

Step 1. Run the application on the computer ribs_karkas_bot.exe (chat bot).

Step 2. Select the commands: / help or / start, then, for example, / ribs.

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

Step 4. The "KARKAS" inference engine is activated.

Step 5. Forms a hierarchical functional system for dialogue with the user "KARKAS".

Step 6. The agent of dialogue is activated and the message with the text of a question and answers a bot is sent. It accepts the message as a JSON object, parses it, 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 and the bot sends it to the output machine to the expert system.

Step8. An expert advice expert system accepts the message and sends the inference engine. The dialogue agent performs the next step according to the constructed hierarchical functional system for the dynamic purpose of the consultation. The purpose of the consultation is formed during the dialogue with the user. In other words, it dynamically depends on the user's answers to the questions of the expert system.

Step 9. The iterative consultation process continues until the inference engine receives the result. The user can stop the consultation at any time with the command / quit fig. 9.

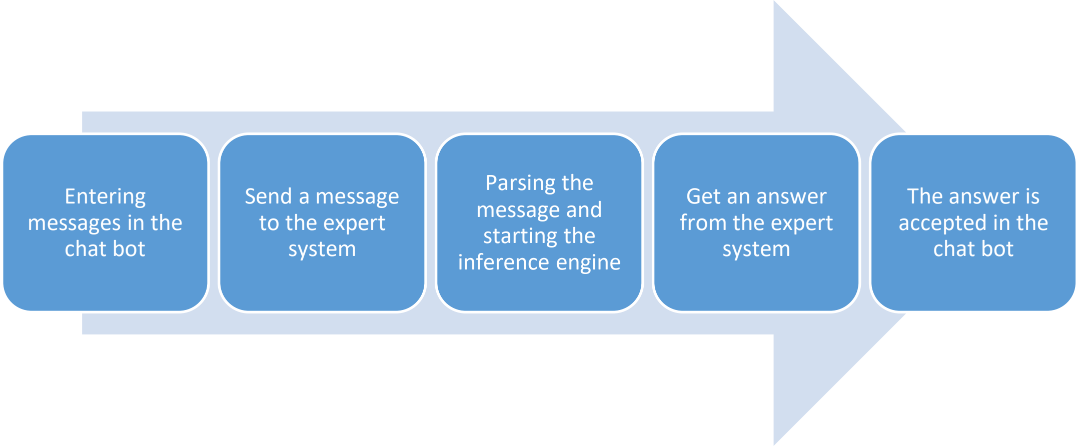

Entering messages in the chat bot
Send a message to the expert system
Parsing the message and starting the inference engine
Get an answer from the expert system
The answer is accepted in the chat bot

Fig. 9. Chat bot interaction @Ribs_karkas_bot step with expert system

Conclusions. As a result, a fully functioning chat bot @RIBS_karkas_bot was created, which is integrated into the "KARKAS" system, which allows for online consultation with the RIBS expert system to determine the risk factor for coronary heart disease. After the application is deployed, it is planned to significantly expand the functionality of the bot.

The "KARKAS" system is a toolkit for developing prototypes of knowledge bases for expert and expert-training systems both offline and online on smartphones. The knowledge representation is based on a hierarchical functional system that is generated by the "KARKAS" system based on production rules and frames. The inference engine uses a hierarchical functional system during user consultation. The user can select various output machine modes: use direct output, reverse output, Bayes formula, criteria tables.

Using the "KARKAS" system, a number of expert system prototypes have been developed in the following subject areas: medicine, economics, mobile communications, and cluster analysis of multidimensional data.

The KARKAS system using chat bots: @Ribs_karkas_bot, @test_karkas_bot, @it_karkas_bot allows online consultation with users and testing students' knowledge in various subject areas: computer graphics, database technology, web analytics, business intelligence systems.

Literature

1. Рассел С. Искусственный интеллект: современный поход / С. Рассел, П. Норвиг. – 2-е изд.; [Пер. с англ. – М.: Изд. дом "Вильямс", 2006. – 1408 с.

2. Бурдаєв В.П. Інтегрування месенджерів з системою "КАРКАС". // Тези доповідей: міжнарод. наук.-практ. конф. Проблеми і перспективи розвитку IT-індустрії, Харків, 2018, с.7.

3. Бурдаєв В. П., Системи навчання з елементами штучного інтелекту. Харків, Україна: ХНЕУ, 2009.

4. Бурдаев В. П., Сложность динамических систем: монография. Saarbrucken, LAP LAMBERT Academic Publishing, 2015.

5. Burdaev V. P. ″About one concept of constructing a temporal knowledge base″, in Proc. of the 1st International Congress Fundamental and Applied Studies in the Pacific and Atlantic Oceans Countries, Tokyo University Press, 2014, pp. 272–276.