Multi-agent Clinical Decision Support System using Case-Based Reasoning

Mykola Korablyov[a], Natalia Axak[a], Oleksandr Fomichov[a] and Volodymyr Hnidenko[a]

[a] *Kharkiv National University of Radio Electronics, av. Nauky. 14, Kharkiv, 61166, Ukraine*

Abstract

Developed multi-agent clinical decision support system (CDSS) for diagnosing diseases, which consists of three high-level modules: interface module, execution module and knowledge module. The system is implemented on the basis of synergy between multi-agent systems (MAS) and Case-Based Reasoning (CBR), which allows increasing the efficiency of implementation of mechanisms of training and adaptation to the characteristics of patients. MAS agents are intelligent and are complex software systems that have the ability to adapt to a specific circumstance that causes a change in their behavior or characteristics, to ensure the ability to adapt and solve the problem before them. Test studies were conducted to determine the diagnosis of heart disease, which showed the ability of the proposed CDSS and the possibility of its use for decision-making in real conditions.
Abstract text.

Keywords [1]

Multi-agent system, clinical decision support system, case-based reasoning, agent, diagnosis of heart disease

1. Introduction

Nowadays, the problem of creating decision support systems (DSS), which are increasingly used in solving complex, difficult formalized problems, which are, in particular, the task of diagnosing various diseases in medicine [1-5]. The task of decision-making is complicated by the need to take into account a large number of different factors. This is primarily due to the difficulty of gathering and analyzing the large amount of information needed to make the right and informed decision.

In determining the diagnosis of the disease is of great importance the need to take into account the mutual influence of different parameters on each other, the reliability of the original data, the completeness of information about the studied patients. Uncertainty arises not only because of the difficulty of creating the model of the object, but also because the developer does not have a clear idea of the relationship of possible solutions, the quality of heuristics used, as well as other difficult factors. The greater the degree of uncertainty about the effectiveness of possible alternative solutions and the importance of the various criteria that evaluate these alternatives, the more important is the expert assessment made by him on the basis of experience and intuition, rather than accurate knowledge. Since it is impossible for experts to diagnose the disease in a large number of people, the application of mathematical methods that have found application in modern clinical decision support systems (CDSS) can help [6, 7].

The CDSS is designed to improve health care delivery by improving medical decisions through focused clinical knowledge, patient information, and other medical information. Traditional CDSS consists of software designed to directly assist in clinical decision-making, in which the characteristics of

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine
EMAIL: mykola.korablyov@nure.ua (M. Korablyov); nataliia.axak@nure.ua (N. Axak); oleksandr.fomichov@nure.ua (O. Fomschov); volodymyr.hnidenko@nure.ua (V. Hnidenko)
ORCID: 0000-0002-8931-4350 (M. Korablyov); 0000-0001-8372-8432 (N. Axak); 0000-0001-9273-9862 (O. Fomschov); 0000-0001-5369-1894 (V. Hnidenko)

an individual patient are compared with a computerized clinical knowledge base, and then clinical assessments or recommendations for a particular patient are transmitted to the physician [8, 9]. The range of functions provided by CDSS is wide, including diagnostics, alarm systems, disease treatment, medication, medication and more. They can take the form of computerized alerts and reminders, computerized manuals, order sets, patient reports, documentation templates and clinical workflow tools, and more.

There are several approaches to building a CDSS [10, 11]:

1. Providing the doctor with relevant information sources that help him to make decisions independently.

2. The use of clinical pathways, so-called technological maps [12], which are prescriptive models of standard health care procedures that must be made for a specific sample of patients;

3, Development of a wide range of private narrow-profile CDSS, in particular Sky-Chain [13], which offers trained neural networks for the analysis of diagnostic images of various diseases;

4. Building a cognitive system capable of self-learning and learning directly from informal text sources, such as the IBM Watson [14] system, which is based on explicit knowledge;

5. Construction of DSS on the basis of a precedent approach [15], which is based on implicit empirical knowledge and according to which it is proposed to form clinical data banks, find cases in them - precedents similar to the current one, and recommend treatment and diagnostic measures based on found precedents.

The main limitations of known methods and technologies currently used in DSS are to solve difficult formalized problems due to insufficient efficiency in solving problems of learning, setting up and adapting to the problem area, processing incomplete and inaccurate source information, interpreting data and accumulating expert knowledge, the same presentation of information coming from different sources, etc. These limitations in DSS can be addressed using the synergies between multi-agent systems (MAS) and case-based reasoning (CBR). Thus, DSS research based on MAS, which uses CBR, to increase the effectiveness of the implementation of mechanisms for learning and adaptation to the specifics of the problem environment are important and relevant.

The paper aims to develop intelligent CDSS using multi-agent technologies, which is to study methods aimed at integrating agents into CDSS, as well as algorithms for using precedent-based reasoning to increase the ability of agents to make decisions. The result of the study is the creation of a software architecture that increases the efficiency of information processing in intelligent CDSS and allows the use of the developed methods and software systems that correspond to specific applications. The proposed methods and algorithms can be used in solving specific decision-making problems in medicine.

2. CDSS based on MAS using CBR

2.1. CDSS based on multi-agent approach

The general structure of multi-agent CDSS for diagnosing diseases is shown in Fig. 1 and consists of an interface module, an execution module and a knowledge module [16, 17]:

The interface module includes interface agents that provide access to decision agents. Access to the execution module and the knowledge module is restricted only to the agents who are there.

The execution module mainly provides services and calculations that may be needed to solve a particular problem and includes: a coordinating agent, a decision-making agent, and a reporting agent.

The knowledge module contains domains and domain-independent problem-solving knowledge. At this level is the database agent, which provides the data that is needed for other agents.

The model of multiagent DSS is represented as follows [18]:

$$\Xi \subset H \times Y, \tag{1}$$

where $H(i)=\{H_D(i)\}$, $(i=\overline{1,M})$ – matrix representation of initial treatment and diagnostic data, $Y=\{Y_D\}$, $(Y \subset \Upsilon)$.

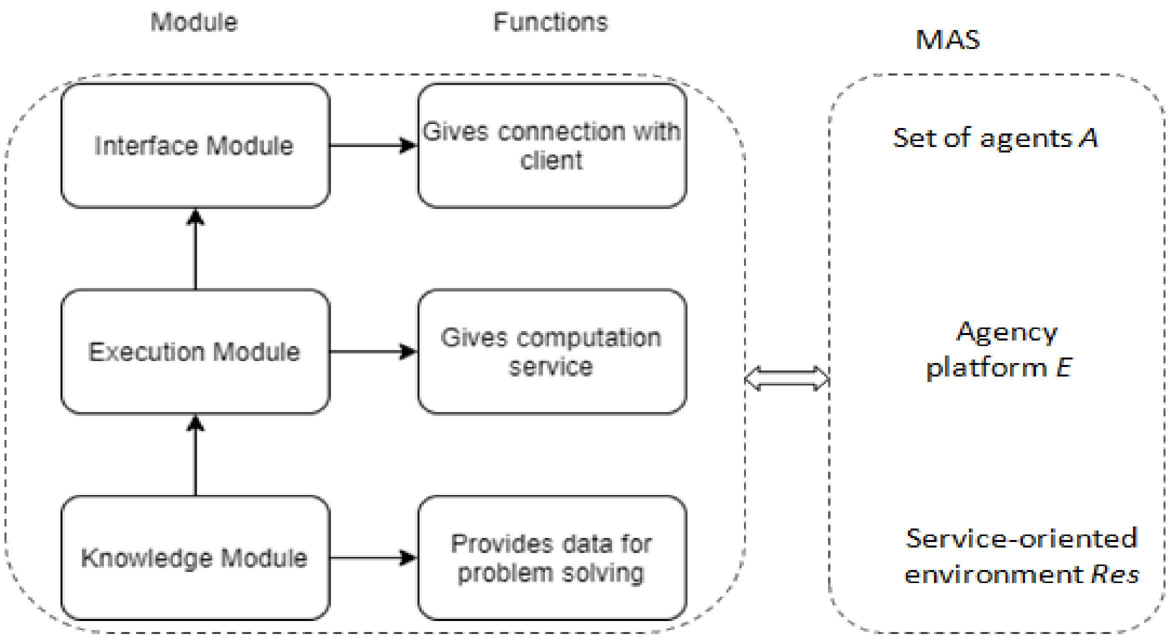

Figure. 1. Structure of DSS based on multi-agent approach

For value Y made a plural of problems, the solution of which pertain to the set D_{Ξ} ={ D_H, D_{MAS} }, where D_H={Par_1, Par_2,..., Par_{14}} – task of processing medical-diagnostic data of measuring devices; D_{MAS} ={D_{Pr}, D_{Plan}, D_{Coord}, D_{Report}}, where D_{Pr}={D_{ident}, D_{match}, D_{select} }, where D_{ident}, – feature identification task; D_{match} – matching task; D_{select} – resolution selection task (selection or ranking); D_{Plan} – the task of building a plan to solve current problems, D_{Coord} – the task of coordinating group actions of agents; D_{Report} – services for generating various reports. The result of the D_{MAS} problem is a set:

$$MAS=\{A,E,Res\}, \tag{2}$$

where agents *A* represented as

$$A=(A_{Search},A_{Adapt},A_{Improve},A_{Exe},A_{Estim},A_{User},A_{Coord},A_{Dec},A_{Report},A_{DB}),$$

E – the environment in which the connections between agents and the environment are implemented; Res is a service-oriented environment that provides a Service to support the identification of possible heart disease Case.

Reflection $T:H_D \to Y_D$ allows for everyone $H_D(i)$ discover $Y_j \in Y_D$ ($j=\overline{1,Q}$, Q – quantity of diseases), which is the resolve to the task D_{Ξ}.

Value $Y_j \in Y_D$ used to formulate decisions about a person's health and to develop further behavioral tactics. As a result of solving the problem D_{Ξ}, a service is provided $Service(H,Y)=Fasaa(T(H_D,Y_D))$

The criterion for the effectiveness of development is the satisfaction of requirements $k:\forall(D_i \in D_{\Xi})\left|(\tau^p<\tau^{max})\&(\tau^r<\tau^{max}\right|\Rightarrow \Xi$, where D_i – is the subproblem of the combined problem D_{Ξ}, τ^p – is the processing period of medical data, τ^r –is the answer time.

2.2. MAS using CBR

Multi-agent technology is used to obtain, reuse and adapt precedents in the CBR system. The structure that combines MAS and CBR technology to support dynamic knowledge sharing is shown in Fig. 2. Each agent is represented using the following set in the CBR MAS (2):

$$A_i=(A_i^a,E_i^e,Fae,A_i^s,Fses,Fse) \tag{3}$$

where A_i^a – agents actions set: i=Search: ***search agent*** A_{Search} – if a new task is introduced, the search agent solves the function of finding the case most similar to the problem. The search ends when the agent finds, indexes of all cases relevant to the problem; i=Adapt: ***adaptation agent*** A_{Adapt} – determines the difference between the selected cases and the problem, and if necessary apply a set of necessary regulations to make the used resolve best suited to the renewed task; i=Improve: ***improvement agent*** $A_{Improve}$ – he criticizes the adapted solution against the previous results. Comparing it with similar solutions in previous cases is the best way. If there is a known error for the derivative solution, the system then decides whether the similarity is sufficient to suspect that the new solution will fail; *i=Exe*: ***execution agent*** A_{Exe} – after the decision is criticized, the executor applies a refined solution to the current problem; *i=Estim*: ***evaluator*** A_{Estim} – if the results are as expected, no further analysis is performed and the cases and their solutions are saved or used to solve future problems. If not, the decision will be reconsidered; *i=User*: ***user interface agent*** A_{User}; *i=Coord*: ***coordinator agent*** A_{Coord}; *i=Dec*: ***decision-making agent*** A_{Dec}; *i=Report*: ***report agent*** A_{Report}; *i=DB*: ***database agent*** A_{DB}; E_i^e – set of environmental conditions; $Fae: A_i^q \times E_i^e \to 2^{A_i^q}$ – functions of environmental behavior; A_i^s – multiple internal states of agents (transmits, receives or waits for messages); $Fses: A_i^s \times E_i^e \to A_i^s$ – agent recovery functions; $Fse: A_i^s \to E_i^e$ – decision-making functions for the agent to act on the current internal state:

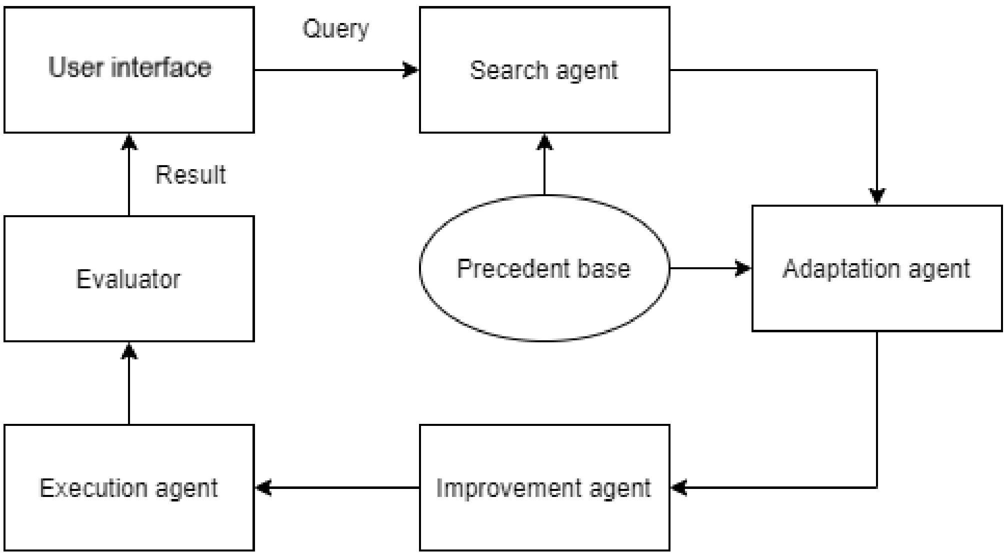

Figure 2: CBR cycle structure

The precedent search agent A_{Search} deals with the process of finding a precedent similar to a new task, which can be focused on different purposes, for example: to find such a precedent so that the resulting diagnosis is the most reliable; find a precedent with a diagnosis that has a minimum execution time; find a precedent for which the time to obtain a new solution will be minimal (minimum modifications); find a precedent that reflects the most modern (late) experience, etc. As a rule, the task of finding a precedent D_{Pr} is divided into three subtasks: 1) D_{ident}, – the choice of properties for comparison (feature identification); 2) D_{match} – comparison; 3) D_{select} – choice of solution (selection or ranking).

The first subtask involves the selection of properties that should be taken into account when finding a better precedent. Precedent-based diagnostic systems typically use a goal, an initial state, and sometimes a description of failures that may arise when solving a problem. The subtask of the comparison is to find one or more diagnoses that (partially) match the selected properties with the current problem. The comparison can be performed in several different ways. The most common are full comparison and use of similarity measures (sometimes, similarity metrics). When fully comparing, a search is made for precedents in which the selected properties exactly match the corresponding properties of the new task. If the system does not find the desired solution, then the generalization of the selected properties of the new task is performed and then again an attempt is made to find a precedent.

A similarity measure is a function that calculates the degree of similarity between a given precedent and a new problem. The simplest measure of similarity can be defined as the number of common sub-goals. More flexible similarity measures calculate the weighted sum of the total sub-goals and the degree of similarity of the initial states. Scales are used to balance the importance and usefulness of tasks.

Weights can be determined by constants for all precedents at once, but in more complex subject areas, each precedent has its own weight vector.

As a result of the operation of the precedent search agent, several relevant precedents can be found. In this case, the planner must select one or more of them. When using similarity measures, this problem is solved automatically. But other methods can be used, based, for example, on the degree of generalization of the precedent (the more specific the precedent, the less effort will be required to adapt).

Consider ways to submit and remove precedents. At the first stage of the CBR-cycle - obtaining precedents - the degree of similarity of the current situation with precedents from the base of precedents (BP) is determined and then removed in order to solve this new problem situation. For the successful implementation of precedent-based reasoning, it is necessary to ensure the correct extraction of precedents from the BP. The choice of the method of obtaining precedents is directly related to the method of setting precedents and, accordingly, to the organization of the BP.

BP can be part of the knowledge base of intelligent CDSS, but can also act as an independent component of the system. The structure of the BP significantly affects the various indicators of the system and, in particular, the time of search and retrieval of precedents. There are different ways of presenting and storing precedents - from simple (linear) to complex hierarchical [19]. It should be noted that simple methods, which are usually based on relational database technology, require a much lower costs of implementation, as well as for the maintenance and support of PSUs than complex ones. However, it may take much longer to find a solution in the simple presentation of precedents than in other more complex ways of presenting and preserving precedents.

The precedent in the general case may include the following components [19, 20]: task description (problem situation); problem solving (diagnosing the problem situation and DM recommendations); the result (or forecast) of the application of the solution. The main ways of presenting precedents can be divided into the following groups: parametric; object-oriented; special (in the form of trees, graphs, logical formulas, etc.). In most cases, for the presentation of precedents is a simple parametric representation, the representation of the precedent in the form of a set of parameters with specific values and solutions (diagnosis and recommendations of DM):

$$CASE = (x_1, ..., x_n, R), \tag{4}$$

where $x_1, ..., x_n$ – parameters of the situation which describes the given precedent; $x_1 \in X_1, ..., x_n \in X_n$; n – parameters number of the precedent; $X_1, ..., X_n$ – valid values range of respective parameters; R – diagnose and recommendations of the DM. Additionally, description of the found resolution use description and comments may be present.

There are a number of methods for extracting precedents and their modifications, the most common of which are: Nearest Neighbor (NN), decision tree method, knowledge-based method, artificial neural networks, etc. [10], which can use different metrics. The method of NN is used in the work, which allows to easily calculate the degree of similarity of the current problem situation and precedents with BP.

Let the given precedent C and the current problem situation T be given in the n-dimensional space of signs (properties, qualities). Then the degree of similarity or proximity $S(C, T)$ of the precedent C and the current situation T can be determined using one of the metrics that determine the distance between two points x_i^C and x_i^T, in particular, the Euclidean distance:

$$d_{CT} = \sqrt{\sum_{i=1}^{n} (x_i^C - x_i^T)^2}. \tag{5}$$

Next, according to the selected metric, the distance from the target point corresponding to the current problem situation to the points representing the precedents from the BP is determined, and the closest point to the target is selected. To determine the value of the degree of similarity $S(C, T)$ it is necessary to find the maximum distance d_{max} in the selected metric, using the boundary of the parameter ranges to describe the initial $x_{i,in}$, in and final $x_{i,fin}$, precedents, $i = 1, ..., n$. Then you can calculate the value of the degree of similarity:

$$S(C,T) = 1 - d_{CT} / d_{\max}, \tag{6}$$

which can take values from 0 to 1.

The work of ***the adaptation agent*** A_{Adapt} will consist of two parts: first, in substituting the purpose and initial conditions of the selected precedent for the purpose and initial conditions of the new task;

secondly, in ensuring the correctness of the diagnosis after replacement. After substituting the goal and the initial conditions of the diagnosis may require changes. Some steps of the diagnosis may be unnecessary, as the goal they served to disappear has disappeared. New goals may emerge, and changing the initial conditions may make a number of steps unsuitable. Correction of the diagnosis can be performed either automatically or by the user.

The result of the search and adaptation phases is a plan to solve the current problem. To close the CBR cycle and replenish the knowledge system, it is necessary to maintain the current planning of the experience. Training is based on observations of the appropriate response to the plan.

After the execution agent applies the solution found to the current problem, the evaluation agent analyzes the results and if the results are as expected, no further analysis is performed and the cases and their solutions are stored or used to solve future problems. If not, the decision will be reconsidered.

The user interface agent A_{User} can be divided into many agents, depending on the needs of the system. The hospital is most successful in using the division into two types of users: 1) engineers (this type of users is engaged in setting up and monitoring the work of the decision-making system); 2) doctors (this type of users analyzes patients' data).

Each of the different users has its own interface agent. Typically, a user interface agent is a web interface that allows users to interact with CDSS, but there can be several different types of such interfaces, for example, it can be: a desktop application or a mobile application. With this interface, users can fill out online forms to interact with CDSS, download documents, and perform data analysis. The physician can provide hospital patient data for analysis using a decision system, and the engineer can download additional data. The user interface agent is responsible for receiving data from users and providing results. It also tracks user settings, can customize the interface depending on the role of the user and his goals, for example, the interface agent can hide some engineering information from doctors.

The user agent interface module contains methods for inter-agent communication as well as receiving input from the user. The process module contains scripts and methods for capturing user input and passing it to the CDSS coordinating agent. The functions of the interface agent provide a web interface for interaction with the user, a web page for user input and description of problems, settings are provided, a web page with information about the status of analysis, providing feedback on the status of various processes, a web page containing final results, dynamic creation of HTML-documents with special formatting depending on the user, etc.

The coordinator agent A_{coord} is responsible for coordination different tasks, which need to be completed while solving cooperative tasks. After getting task from user of interface agent, coordinator agent identifies respected criteria’s, determines alternative, which need to be evaluated, and generates action plan, which is like rating of alternatives. These alternatives can include identification of respectful data sources, query services of other agents and generating reports.

Task of action planning D_{Plan} can be described in the next way:

$$AP = \{ A_i^a, P_i^{ap}, I_i^{ap}, \rho_{ap}, t_{ap,0} \} \tag{7}$$

where P_i^{ap} – the set of the agent's perception of the state of the environment; I_i^{ap} – a subset of the internal states of the agent that is part of the set of internal states $I = I_i^{ap} \times I'$; $\rho_{ap} \subset P_i^{ap} \times I_i^{ap} \times I_i^{ap} \times A_i^a$; – conversion ratio, which determines that the perception P_i^{ap} current perception of environment and current internal state of the plan t_i^{ap}; $t_{ap,0} \in I_i^{ap}$ – starting agents state.

The result of planning is a set of chains of transitions of agents from the initial state to the final, which realizes the goal.

After receiving a request from the coordinating agent for a decision, the process module determines what actions need to be taken to successfully process the request from the user. To do this, it can send requests to the database agent to find additional knowledge about the subject area, and it also sends a request to the decision agent with information about the problem.

The decision-making agent A_{Dec} stores in the knowledge module information about possible agents who are able to solve the problem. Using this information and depending on the type of application, the agent must determine, based on the metadata stored in the knowledge module, the module that best handles it, and send it a request for a solution. In our system, such a module is one, it is a decision-making module using CBR.

The report agent A_{Report} is located in the execution module and provides users with services for generating various reports on the operation of the system. These can be both text reports and various charts: bar and bar charts; graphics; pie charts; histograms; area diagrams; scatter charts; surface diagrams; pie charts; bubble charts, etc. These reports are based on the information that the agent receives from the database agent and may include a variety of information about the parameters of the distribution of data in the knowledge base.

As an example, these can be histograms of heart disease, shown in Fig. 3. Such information can be very important for the analysis of symptoms that can lead to heart disease. For example, if we construct a histogram of the distribution of various attributes of people in whom heart disease was detected, as shown in Fig. 3, we can see that the number of people with the disease increases sharply after 50 years. All this can be very useful for further diagnosis of heart disease.

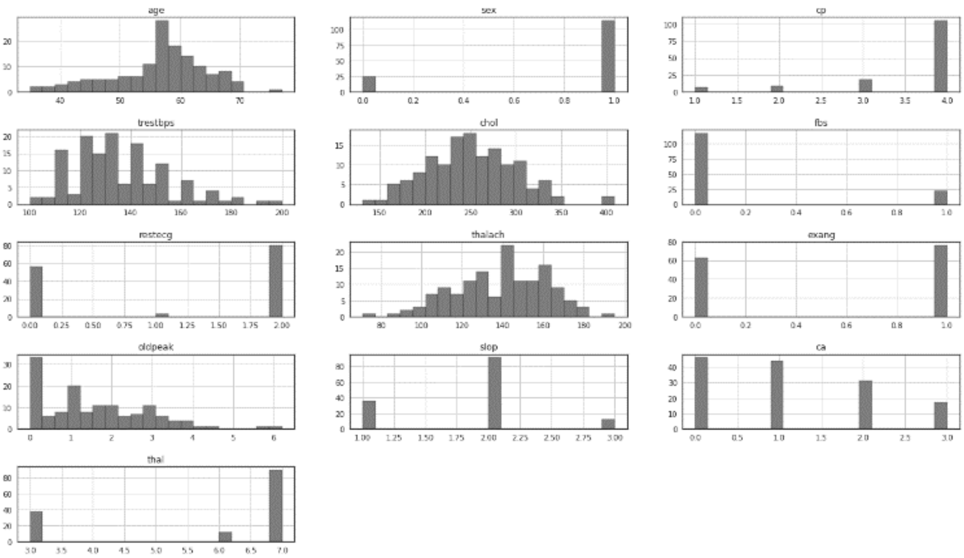

Figure 3: Histograms of heart disease

The database agent A_{DB} is responsible for tracking what data is stored in the database. The database agent interface module provides a public interface to existing databases. This improves the relationship and allows users to access various data sources that might otherwise be unavailable. The process module provides special and predefined data retrieval capabilities. Based on the user's request, the corresponding requests are created and executed in the data warehouse. The results of these queries are passed to the user or other agents.

The database agent can use multiple sources to generate a response to a query. These can be: SQL databases; NoSQL database; text files of various formats (XML, JSON); external data services, etc. The functions of a database agent are interoperability between agents, a database interface, support for local and global schema, and formatting query results according to user needs.

2.3. Integration CBR into CDSS

To combine CDSS agents and CBR agents, you need to connect the search agent to the decision agent. The architecture of the system is shown in Fig. 4, and the sequence diagram is shown in Fig. 5. In the CBR system itself, the application first goes to a search agent, which solves the function of finding cases most similar to the problem. It then passes the result to the adaptation agent, which determines the difference between the selected cases and the problem and, if necessary, applies a set of necessary rules to make the old solution best suited to the new problem. The enhancement agent then criticizes the adapted solution against the previous results, and after the solution is criticized, the execution agent applies the refined solution to the current problem. At the end, the evaluator stores the result in the precedent database for further use and sends the result to the decision agent, who in turn returns the result to the user through the coordination agent.

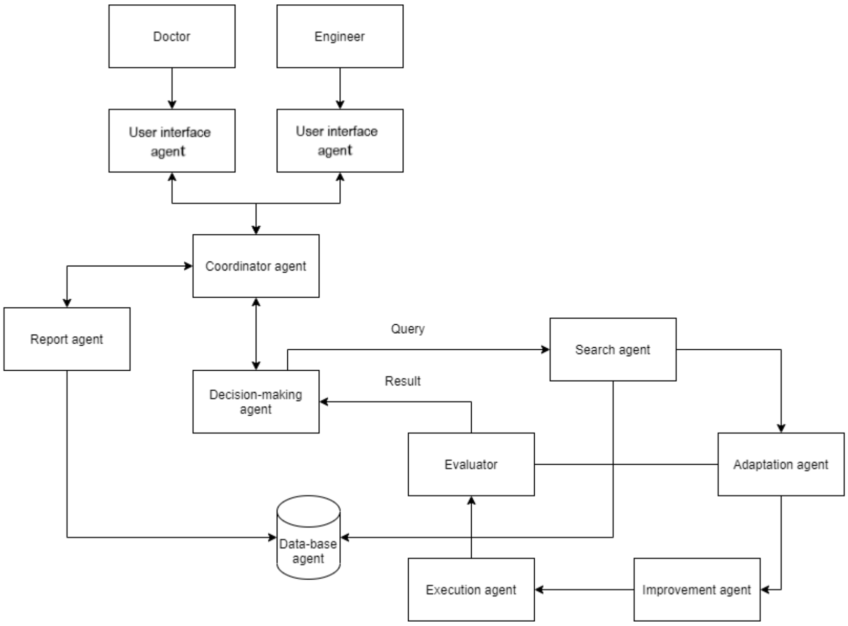

Figure 4: Architecture of multi-agent CDSS using CBR

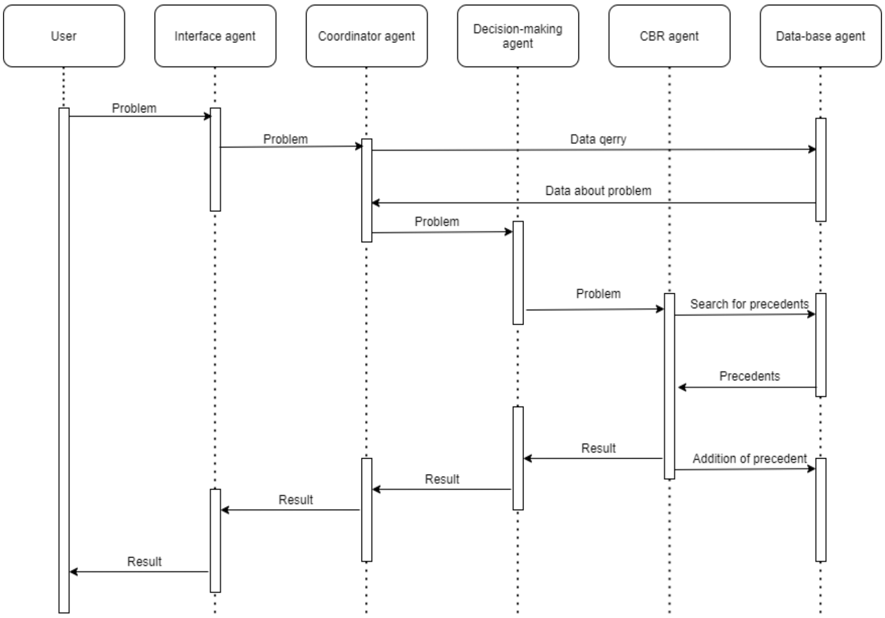

Figure 5: General sequence decision making diagram

3. Experimental studies of the system

The task of supporting the identification of possible heart disease requires effectively organized information that reflects the experience and knowledge of specialists (experts) in knowledge bases, forecasting possible solutions, their analysis and evaluation, which allows the decision maker (DMP) to more reasonably choose one of possible diagnoses [3]. When determining the diagnosis, the specificity of the bases of diagnoses is that they include heterogeneous knowledge of the subject area, knowledge of many different parameters (age, gender, type of pain, blood pressure, blood sugar, electrocardiogram result, maximum heart rate, etc.), which affect the outcome of the decision. Therefore, the feasibility of designing a multi-agent system to ensure the most rational decision-making has been shown. LEADSTO, Leads to Editor is used to model and implement the interactions between agents of a multi-agent system in decision-making for timely response, TTL (Temporal Trace Language) dynamic apparatus to demonstrate the dynamics of agents' behavior.

For the test system for the diagnosis of heart disease, a data set was used, which has more than 300 records and has 14 attributes:
1. Age
2. Sex (1 – man; 0 – woman)
3. SP – type of chest pain:
   - value 1 – typical stenocardia;
   - value 2 – atypical stenocardia;
   - value 3 – non-anginal pain;
   - value 4 – asymptomatic.
4. Trestbps – blood pressure at rest (in mm Hg. On admission to the hospital).
5. Chol – serum cholesterol in mg/dl.
6. Fbs – blood sugar > 120 mg / dl:
   - 1 – true;
   - 0 – false.
7. Restecg – electrocardiographic result:
   - value 0 – normal;
   - value 1 – the presence of an anomaly of ST-T waves (inversion of T-waves and / or increase in ST or depression> 0.05 mV);
   - value 2 – showing probable or definite left ventricular hypertrophy according to Estes criteria.
8. Thalach – the maximum heart rate is reached.
9. Exang – exercise-induced stenocardia:
   - 1 – true;
   - 0 – false.
10. Oldpeak – ST depression caused by exercise relative to rest.
11. Slope – the slope of the peak training segment ST:
   - value 1 – rise;
   - value 2 – direct;
   - value 3 – recession.
12. CA – number of large vessels stained by fluoroscopy.
13. Thal – thallium heart scan:
   - 3 – normal (without cold spots);
   - 6 – fixed defect (cold spots during rest and exercise);
   - 7 – reverse defect (when cold spots appear only during training).
14. Pred_attribute – diagnosis of heart disease (condition of angiographic disease):
   - value of 0: < 50% of the narrowing of the diameter;
   - value of 1: > 50% narrowing of the diameter.

To start the system, you must send a request to the interface agent. Then the system begins its work, which will result in the selection of the four most appropriate precedents, in order of increasing distance from the original. Four possible solutions to the problem have been provided for ATS, including a precedent with complete similarity. Therefore, based on the obtained data, it can be concluded that although there is a precedent, which is similar to the initial one with heart disease, but based on the work of CDSS, the chance of diagnosing the disease in a person with these indicators is low. The distribution of time between agents for decision making is shown in Fig. 6. To make one decision, the system needs about 170 milliseconds, as shown in Fig. 6, which is a good indicator, but can be improved if additional decision-making agents are added to the system, which will allow to horizontally scale the resulting system.

Analyzing the obtained data on the distribution of time for decision-making, we can conclude that the decision-making agent occupies about 40% of all calculations in the system. And if we remember that the disadvantages of the CBR include the increase in the search time of the nearest precedents, it turns out that it is necessary to conduct a comparative analysis of the search time depending on the size of the precedent database. The result of testing the performance of the system depending on the number of precedents in the database is shown in Fig. 7.

◢ dotnet.exe (PID: 16684)	171 (100.00%)
MultiAgentDSS.Agents.Interface...	167 (97.66%)
MultiAgentDSS.Program::Main	167 (97.66%)
MultiAgentDSS.Infrastructure.M...	166 (97.08%)
MultiAgentDSS.Infrastructure.M...	165 (96.49%)
MultiAgentDSS.Agents.Coordin...	163 (95.32%)
MultiAgentDSS.Agents.Desigion...	153 (89.47%)
MultiAgentDSS.Agents.Desigion...	75 (43.86%)
System.Collections.Generic.Has...	2 (1.17%)
MultiAgentDSS.Agents.DataBas...	2 (1.17%)
System.Collections.Generic.Has...	1 (0.58%)

Figure 6: Time distribution between agents for decision making

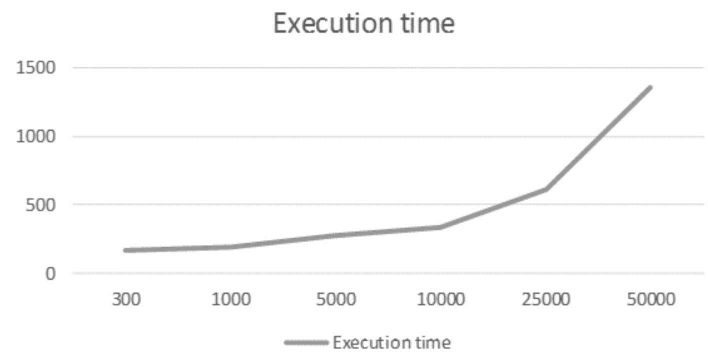

Figure 7: Dependence of decision-making time on the number of precedents graph

As a result of testing, we can say that although with the increase in the number of precedents in the database, the execution time really increases, but it is slow and does not significantly affect the overall running time. Despite these shortcomings, the experiment showed the ability of CDSS, so it can be used to make decisions in real conditions.

4. Conclusion

CDSS is proposed, which is designed to improve health care delivery by improving medical decisions through focused clinical knowledge, patient information, and other medical information. CDSS is based on the use of multi-agent approach and Case-Based Reasoning and consists of modules of interface, execution and knowledge that include various agents. Agents are intelligent and complex software systems that have the ability to adapt to a particular circumstance that causes a change in their behavior or characteristics, to ensure the ability to adapt and ultimately solve the problem before them.

Multi-agent technology is used to obtain, reuse and adapt precedents in the CBR system. The CBR system removes cases related to an existing disease from the database of precedents and solves the issue of deciding the diagnosis of the disease based on the results of previous data on such a disease.

Experimental studies of the proposed CDSS to support the identification of possible heart disease. For the test system for the diagnosis of heart disease, a data set was used, which has more than 300 records and has 14 attributes. The obtained data on the distribution of time for decision-making showed that the decision-making agent occupies about 40% of all calculations in the system. The result of testing the performance of the system, depending on the number of precedents showed that with increasing number of precedents in the database execution time increases, but this is slow and does not significantly affect the overall operating time. Experimental studies show the effectiveness of the proposed solutions: reducing the response time of the system by an average of 15%.

5. References

[1] R.V. Garcia, M.A. Parades-Valverde, M.P. Satas-Zarate, G.A.-Hernandes, Exploring Intelligent Decision Support Systems, Current State and New Trends, Springer International Publishing AG, 2018.

[2] A. Kaklauskas, Biometric and Intelligent Decision Making Support, Springer International Publishing Switzerland, 2015.

[3] S. Borra, N. Dey, S. Bhattacharyya, M.S. Bouhlel, Intelligent Decision Support Systems: Applications in Signal Processing, De Gruyter Frontiers in Computational Intelligence, Volume 4, 2019. doi.org/10.1515/9783110621105.

[4] F. Dargam, P. Delias, I. Linden, B. Mareschal, Decision Support Systems VIII: Sustainable Data-Driven and Evidence-based Decision Support, Springer International Publishing, Proceedings 4th International Conference ICDSST, 2018.

[5] M. Korablyov, N. Axak,. O. Barkovska, D. Rosinskiy, Hybrid Adaptive Decision-Making Model based on Fuzzy Logic and Artificial Immune Systems, Proceedings of the 14th International conference «Computer Sciences and Information Technologies» (CSIT). Volume 1, 2019, pp. 62-66.

[6] R.T. Sutton, D. Pincock, D.C. Baumgart, D.C. Sadowski, R.N. Fedorak, K.I. Kroeker, An Overview of Clinical Decision Support Systems: Benefits, Risks, and Strategies for Success, NPJ Digital Medicine 3, 17 (2020). doi:10.1038/s41746-020-0221-y.

[7] K. Farooq, B.S. Khan, M.A. Niazi, S.J. Leslie, A. Hussain, Clinical Decision Support Systems: A Visual Survey, Informatica, 42 (2018) 485–505. doi.org/10.31449/inf.v42i4.1571.

[8] A.M. Shahsavarani, E.A.M. Abadi, M.H. Kalkhoran, S. Jafari, S. Qaranli, Clinical Decision Support Systems (CDSSs): State of the Art Review of Literature, International Journal of Medical Reviews, Volume 2, Issue 4, 2015, pp. 299-308.

[9] D. Zikos, N. DeLellis, CDSS-RM: A Clinical Decision Support System Reference Model, BMC Medical Research Methodology 18, 137 (2018). doi.org/10.1186/s12874-018-0587-6.

[10] V.L. Malykh, Decision Support Systems in Medicine, Software Systems: Theory and Applications, 2 (41), Volume 10, 2019, pp. 155-184. (in Russian).

[11] A.V. Duvalkina, Decision Support Systems in Medicine, Modern Scientific Research and Innovation. 2 (2017). URL: http://web.snauka.ru/issues/2017/02/78010. (in Russian).

[12] G. Kopanitsa, I. Semenov, Patient Facing Decision Support System for Interpretation of Laboratory Test Results, BMC Medical Informatics and Decision Making, 18 2018.

[13] E. Herrett, A.M Gallagher, K. Bhaskaran, H. Forbes, R. Mathur, T. van Staa, L. Smeeth, Data Resource Profile: Clinical Practice Research Datalink (CPRD), International Journal of Epidemiology, Volume 44, Issue 3 (2015) 827–836. doi.org/10.1093/ije/dyv098.

[14] D. Hernandez, T. Greenwald, IBM Has a Watson Dilemma, The Wall Street Journal, 2018.

[15] S. Srinivasan1, J. Singh, V. Kumar, Multi-agent based Decision Support System using Data Mining and Case Based Reasoning, International Journal of Computer Science Issues, Volume 8, Issue4, No 2 (2011) 340-349. ISSN (Online): 1694-0814.

[16] D. Sharma, F. Shadabi, Multi-agents based Data Mining for Intelligent Decision Support Systems, The 2nd International Conference on Systems and Informatics (ICSAI), 2014, pp. 241-245. doi:10/1109/ICSAI.2014.7009293.

[17] S. Khademolqorania, A.Z. Hamadani, An Adjusted Decision Support System through Data Mining and Multiple Criteria Decision Making, Social and Behavioral Sciences, 73 (2013) 388–395.

[18] N. Axak, Development of Multi-agent System of Neural Network Diagnostics and Remote Monitoring of Patient, Eastern-European Journal of Enterprise Technologies, 4(9(82), (2016), 4–11. doi.org/10.15587/1729-4061.2016.75690.

[19] E. Schake, L. Grumbach, R. Bergmann, A Time-Series Similarity Measure for Case-Based Deviation Management to Support Flexible Workflow Execution, Proceedings of 28th International Conference on Case-Based Reasoning Research (ICCBR), 2020, pp. 33–48.

[20] A.V. Krylov, The Problem of Knowledge Extraction using Case-Based Reasoning, Proceedings of Universities, Volume 61, ’11, 2018, pp. 956—962. (in Russian).