Int. J. Nonlinear Anal. Appl. 13 (2022) No. 1, 1827-1839
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.5801

Implementing the EBDI model in an E-health system

Abir Hussein Jabber[a,*], Ali Obied[b]

[a]Department of Computer Science, University of Al-Qadisiyah, Al-Qadisiyah, Iraq
[b]Computer Science Department, College of Computer Science and Information Technology, University of Al-Qadisiyah, Iraq

(Communicated by Madjid Eshaghi Gordji)

Abstract

Health-care systems (patient monitoring and diagnosis systems) have recently attracted the interest of researchers. Intelligent software has been used as an agent in this field where research addresses the areas of electronic health (E-health) to develop and improve health care systems and to reduce the effort and time on health users (physicians, nurses) and to follow up on the condition of patients, especially the elderly, and also Intelligent software agents are using a result when most hospitals are out of capacity. The major goal of this study is to learn how we may use intelligent software for E-Health to assist physicians, nurses, and other health practitioners in collecting and tracking patient data on a daily basis in order to enhance treatment decisions. How can we achieve this in real-time in a dynamic and non-deterministic environment? As a result, we have built a model-based system such as the Extensible Beliefs Desires and Intentions (EBDI) model as architecture for located autonomic software agents in run-time under dynamic and non-deterministic environments circumstances in this study. We have formalized the EBDI agent in E-Health as an online observer to monitor the state of a patient linked to sensors (thermometer, glucose meter, oxygen meter, etc.). This agent filters the data (sensor readings) and determines which readings are normal or abnormal before sending the patient's information to the administrative agent, which is a different type of EBDI agent. In addition, the system proposes a treatment plan based on the sensor readings and necessitates the participation of a human doctor in emergency situations. This study proposes developing a monitoring system to follow up (monitor) the health status of patients based on a multi-agent-system (MAS) with the Clustering process (using the K-mean Algorithm), where the system is designed to support warnings and alerts in abnormal health conditions and to call for a doctor's intervention in emergencies (when it is necessary).

Keywords: Situated Agent, Electronic-Health (E-health), (EBDI), K-mean algorithm, multi-agent-system (MAS)

*Corresponding author
Email addresses: com.post01@qu.edu.iq (Abir Hussein Jabber), ali.obied@qu.edu.iq (Ali Obied)

Received: October 2021 *Accepted:* October 2021

1. Introduction

In recent years, agent technology has experienced great growth in a number of disciplines, notably in the field of E-health, where medicine is defined by its complexity, dynamism, and diversity. A variety of difficulties arise (increasing cost of care, population growth, and lack of caregivers) in the healthcare industry filed. Therefore the characteristics of Intelligent Agents (proactivity, intelligence, sociability, and autonomy) are a good choice to address many challenges in this sector, because the key success criteria were the natural translation of real-world medical problems into electronic life, These qualities also result in decreased operating costs and the automated execution of activities in various Agent-based systems, As a result, Agent Technology-based intelligent applications may be able to provide better health care than the traditional medical system. In truth, one of the most fascinating features of modern health care is the possibility of automating fundamental tasks like monitoring patients' status and providing rapid aid in an emergency [8, 5, 6].

The Multi-agent systems (mas) have cleared the door for new applications in the healthcare industry, such as socialized and personalized healthcare models. As a result, this article aims to develop both a multi-agent and an agent-based system for the hospital. This article illustrates how EBDI-based software agents with K-mean algorithm and the Jade framework may be used to automate tasks while simultaneously enhancing patient service [5, 6].

2. Motivation

Situated autonomic software can be envisaged as a system which acts and/or reacts autonomously to external stimuli, generated from sensing its environment, which is achieved independently of external human intervention.

The motivation for this work is to understand the theoretical and practical foundation of automated generation and refinement of system's observer models.

Because of the health care field dynamic, various challenges face in healthcare (population growth increasing cost of care, and lack of caregivers). Therefore, intelligent deliberative software agent that acting rationally can deliver the best health care than the traditional medical system. To: enable surgeons, nurses, and health professionals to gather and track patient data regularly, optimize treatment choices, maximize hospital quality, and help them control procedures.

3. Background

3.1. Situated Agent

In recent years, both artificial intelligence (AI) and computer science have turned their attention to the idea of software agents. Small systems for customized email filters [8] to big and sophisticated systems like air traffic control [12] are among the uses. Agents, on the surface, appear to be systems that can reason and choose their course of action, such as satisfying their (design) objectives and/or goals [15, 13]. Given a set of beliefs about the universe, an agent is considered to be rational if it chooses to behave in its own best interests [16]

One of the AI community's goals is to create computer programs that can operate as autonomous, rational agents, making logical judgments about what actions to do on their own. However, it is not enough to create programs that conceive of appropriate behaviours to do independent of the condition of their environment — agents' behaviour must be located and matched with the state of their environment [16]

Definition 3.1. *Situated Autonomic software can be envisaged as a system which acts and/or reacts autonomously to external stimuli, generated from sensing its environment.*

As shown in the (Figure 1) the agent receives sensory information from its surroundings and reacts if required according on a given action groundings or policies .An agent's environment which it occupied it can be physical (in the case of robots) or a software environment (in the case of a software agent).actions will be physical such as moving objects around. The sensor input received by an agent can include video feeds In the case of physical embodied agent, .but actions will be software commands such as UNIX command, which removes a file, and sensor input will be obtained by performing command such as which obtains a directory listing ,at the case of software agent [11]

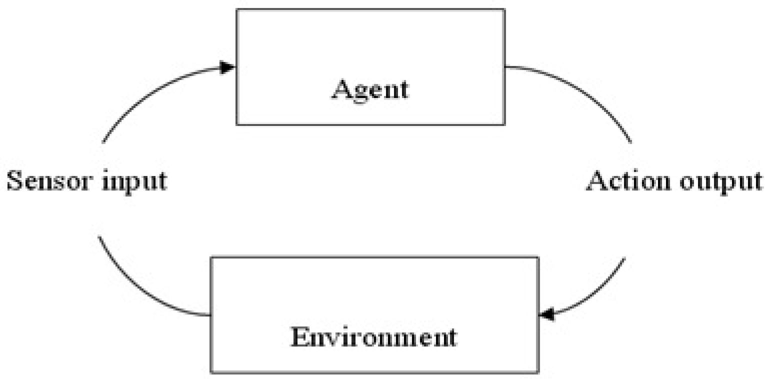

Figure 1: The agent takes sensory from the environment and produces as output actions that affect it

In virtually all realistic applications, agents have at most partial control over their surroundings. In other meaning, while they are achieving actions that change their surroundings, they cannot control completely the behaviour of their environment. As result for that It makes logical, sense to split the environment, at least conceptually, into those portions that the agent over which it has no control controls, those parts controls, and, those parts it partially controls.

3.2. Extensible Beliefs Desires and Intentions (EBDI) model

That the proposed (Extensible Beliefs, Desires, and Intentions) offer a highly appropriate framework to the construction of embedded intentional software that continually observes or/and monitors its surroundings and acts in accordance with its situated BDI, which is based in normative contexts. [2,3,4,8 and 9].Methods are provided by EBDI for improving an agent's decision-making processes such that the action has the maximum reward). Beliefs based on information acquired or retrieved from a variety of sources, such as the environment, domain, or other services' Beliefs; reflect the state of events in an ideal world that this often growing the services has targeted. In contrast, it is desirable that a system's beliefs group (monitored system states) oppose it, which the system discovers is not similar and triggers a collection of intents [13, 9, 2]. Additionally, EBDI has three levels of intentions:

- Situated intentions: an action set to a system specifies an action that the system will do to satisfy its defined wishes in a particular circumstance, and/or to reconcile a mismatch between the models aims (desires) and the model's surroundings (beliefs).

- Normative intentions: indicate a set of activities that must be taken to ensure that a set of criteria, including duties (deontic), is followed and rule representations are observed before a set of standards is observed. The grant of intent is also used to preserve the legitimacy of new rules.
- Utility intents are a set of activities designed to maximize the utility of target-oriented intentions [5, 6] and [14].

These three elements complement a library with plans. The strategies describe the procedural understanding of low-grade activities that are meant to contribute to attaining a target in certain conditions, i.e. they provide strategic steps that determine how to achieve it. Balk and Gilbert (Balk and Gilbert) Goals are defined in terms of what an agent aims to accomplish purposefully [14].The BDI agent paradigm recognizes the supremacy of logical action principles in terms of wants, intents, and beliefs. Intentions directed and limited constrain an agent's strategy processes and are a useful concept for directing the reasoning of the agent. This intentional approach has been successfully utilized to generate single-agent templates to create single-agent models [5, 6] and [14].

Embedded intents serve as a task group for the system, allowing it to fulfil its chosen wishes in a given scenario, or to handle the mismatch between the system's aims (desires) and the system's environment (beliefs). Normative intents are a collection of activities done to ensure a certain group of rules, including duties (deontic), and the rule's operation is checked before a specific intention is carried out. Maintain the integrity of evolving regulations as well. Utility intents are a set of activities that work together to improve goal-oriented intentions [12].This means that an agent may have many competing purposes at any same time. In the first case, there may be a disagreement over whether the conduct was intentional or not. As the agent seeks order to conform to its norm group (ontology), the general norm group (shared ontology), and its duties to itself and another agent, the more deliberate conflict will arise.

Several of these competing motives may even be incompatible. Methods for improving an agent's decision-making such that the action with the largest benefit for the system is taken have been researched on occasion. Defining and using functions that add a sense of action benefit, on the other hand, is extremely difficult. A very high (even infinite) number of perception-action pairings may be included in the full specification, which may change from one work to the next [9]. Furthermore, because the terms "agent" and "environment" are intimately linked, none can be explained without the other. In fact, noticing the difference between an agent and its surroundings is rare, and drawing a line between them can be challenging at times [14].

3.3. E-health (Electronic Health)

Electronic health (which also known as e-health) is a type of healthcare. It is uses the ICT (information and communication technology) information and communication systems to sustain it.

The precise meaning of the term E-Health varies depending on the situation. Despite the lack of a generally accepted clear or exact definition, the word "E-Health" has become a well-known neologism that is frequently used by academic institutions, funding agencies, professional organisations, and people [10, 4]. E-health refers to the organizing and transmission of health information and services using the Internet and associated technologies. E-health is a new branch of medical informatics. The term alludes to a modern method of working, an attitude, and a dedication to networked, global ideas in a broader sense.

ICT is using to enhance health care regionally, locally, and globally. Aside from the increasing use of information technology in healthcare today, numerous research projects have been done to investigate strategies to improve E-Health distribution and management efficiency [3].

Thanks to recent technical developments the most current technological advancements provide, electronic medical equipment with memory or a network card capable of delivering patient records is now accessible. Both wired and wireless networking topologies can gather data in real time [1].In addition, health focuses on smartphone situations in which devices are used to gather, transport, and process important patient data, as well as in-home care. In general, these systems are intended to enable mobile and ubiquitous access to medical information.

E-health has become increasingly important in reducing the strain on health-care services. E-health, in other words, is the use of information and communication technology (ICT) in the health sector for clinical, educational, and administrative purposes. It enables the use of computer, network, and information technology to improve healthcare quality, patient safety, and secure confidential access to health information to enable individuals and communities to make the best possible health decisions while reducing healthcare costs [5, 6, 18, 17].

3.4. K-mean algorithm

The most prevalent clustered partitioning method is K-mean is an unsupervised computational approach for grouping comparable items into smaller partitions called clusters to group them [20] and is a partition-based technique Through a continuous iterative process, this algorithm clusters data When the algorithm converges with an end condition then the iteration operation is terminated. After that, output the clustering results. This method is effective for dealing with large amounts of data. Each cluster in k-mean is represented by the mean value of its items. We divide a set of n objects into k clusters with low inter-cluster similarity and high intracluster similarity. The mean value of items in a cluster is used to determine similarity. The algorithm is divided into two parts. The first phase selects k - centroid at random, with the value k predetermined. In phase two the closest centroid is assigned to each object in the data collection. The distance between each data object and the cluster centroid is measured using Euclidean distance for example [20].

In another hand, this method tries to increase intracluster variance while minimizing within-cluster variation. The initialization of the cluster centroids process involves identifying the number of clusters first and then randomly allocating cluster centroids to each cluster from the whole datasets. A distance metric is then used to determine the distance between each point in the entire dataset and each cluster centroid (e.g. Euclidean distance) [7].

The minimal distance between each data point is then calculated, and that point is allocated to the cluster with the shortest distance. Cluster assignment is the next phase, and it is repeated until all of the data points have been allocated to one of the clusters.
Finally, the mean for each cluster is computed using the total number of points in that cluster and the collected values of those points [7, 19].

Those means are then assigned as new cluster centroids, and the process of finding distances between each point and the new centroids is repeated, where points are re-assigned to the new closest clusters. The process iterates a fixed number of times until points in each cluster stop moving across to different clusters. This is called convergence. The steps of the K-means can be summarized in Figure (2) below [19].

4. Method

4.1. Formalizing EBDI Agent as Observer in E-Health

We have two kinds of observer agents, so, in this point we can formalize EBDI agent as observer architecture in E-health in two ways:

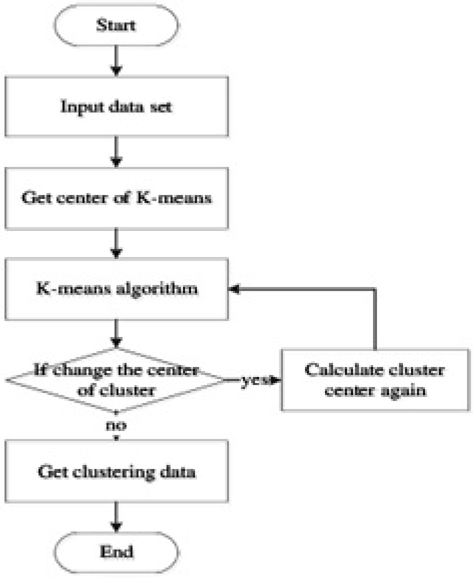

Figure 2: K-mean clustering Algorithm

In this point we can formalize EBDI agent as observer architecture in E-health in two ways:

For monitoring agent: To formalizing EBDI agent to observing patient the system made use of seven crucial sensors' reading (oxygen, temperature, blood pressure (Bb), blood glucose, Respiratory Rate (RR), and Pulse Rate (PR)..). So, to evaluate the Boolean functions, starting from receive a reading, if the reading in acceptance range, so keep observing the patient. If the reading is not in acceptance range, send alert for administrative agent (server). The reward function of EBDI agent for solving Acquisitiveness Boolean function is:

$$R(S,a) = \begin{Bmatrix} W(S_{int}) & if & a = KeepObserv \\ W(s) & if & a = sendAlert \end{Bmatrix} \tag{4.1}$$

Where $S_{int} \in S$ refers to the state the agent intends to be in while currently being in state s. Thus denotes whether the agent keep observing the patient as no danger in his life or sending an alert message for administrative agent (server) if there is any uncertainty about the patient's health, and abnormality in the acquired data, which means the patient's condition, is abnormal.
Also, for deliberating the observer makes its decision (What to do) dependent on utilities. At this stage there are two actions:
u_1 The sensing reading represent there is a risk, so, send alarm message.
u_2 The sensing reading represents there is no risk, so, keep observing.

The decision in above depend on the overall system parameters can be views as the monitoring agent receiving sensor's readings (parameter's measures such as oxygen, temperature, blood pressure (Bb), blood glucose, Respiratory Rate (RR), and Pulse Rate (PR)..)

For Administrative agent (server) in this stage the agent will receive message from monitoring agent which is abnormal case for patient, and the agent will compute membership about the patient case. Computing membership either will lead to; alternative plan, or keep the same plan, or need second opinion. So, in this case we need to evaluate the Boolean functions, for the effects of computing membership process.

If there is alternative plan available, suggest it. If there is no alternative plan send message to ask for second opinion (such as ask Consultant). The reward function of EBDI agent for solving Acquisitiveness Boolean function is:

$$R(S,a) = \begin{Bmatrix} W(S_{int}) & if & a = sendPlan \\ W(s) & if & a = AskFHelp. \end{Bmatrix} \tag{4.2}$$

Where $S_{int} \in S$ refers to the state the agent intends to be in while currently being in state s. Thus denotes whether the agent sending an alternative plan, if it found after using K-means algorithm to compute memberships, or sending a help message to get second opinion from consultant, as follow;
u_1 There is no alternative plan, so, send message for second opinion (ask consultant for help).
u_2 There is alternative plan, so, send alternative plan.
The decision in above depend on the results of computing membership process, that used k-means algorithm.

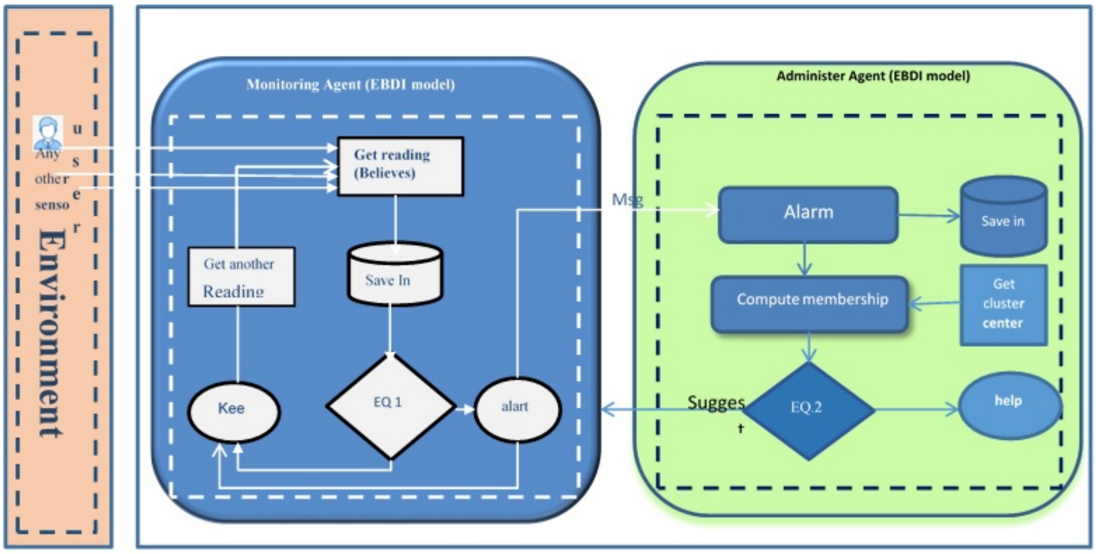

Figure 3: describe implementing EBDI model in E-health system

4.2. K-mean algorithm Step

This stage is an important stage of the model stages in our proposed system (implementing intelligent software agent for E-health system) and through which it is possible to cluster the sensors' readings depending on the degree of similarity between them into groups. The dataset that is used in this stage is collecting from the hospital by tracking the patients' stats which were 900 samples each sample has 7 features these features are represented the sensors' reading(vital parameters: oxygen, temperature....) There are several algorithms used to clustering these data such as the K-mean algorithm that we used at this phase because it is more active with the numeric and big data the result for applying this phase are shown in figure (4) below which is described for five groups obtained after the applied k-mean algorithm.

Algorithm: K-means

Input: 900 sensors reading.
Output: 5 clusters center;
Begin
Initialize-cluster-centers () // Cluster Center Generation
Initialize –data -points () // Get Data points we want to cluster
Boolean change = true //check if there is change the process continues else break
Total in cluster=0 // Total for elements in each cluster
Minimum = max-d //suppose the smallest distance as biggest integer number to find the smallest distance
Distance: 0.0
Cluster: 0
While (change)
While (loop ¡ 900)
Begin
Total in cluster: 0
Loop ++
Compute-centers ( ) // function to calculate cluster centers
Change: false
For: $i = 0$ to $i < N$-points
Begin
Minimum: max-d
Distance: 0
For: $j = 0$ to $j < N$-clusters
Begin
Distance: get-distance (dataset[i].vector1, centroids[j].center)
If distance¡minimum
Begin
Minimum: distance;
Cluster: j
End if
End for If Dataset [i].cluster ¡ ¿ cluster
Begin
Print (dataset[i].cluster +" "+ i+" "+cluster)
Change: true
Dataset [i].cluster: cluster
End if
Print (dataset[i].cluster +" "+ i+" "+cluster)
I++
End for
End for
End while
Save clusters center
End Algorithm

Figure 4: K-mean algorithm

4.3. Computing membership step

In this step when implementing the membership to the cluster center, n membership values are calculated for each alarm vector. This approach uses a function named (get-distance) to determine the Euclidean distance between two vectors. The get-distance algorithm is depicted in Figure (5).

Algorithm: get-distance function

Input: two vectors x_1, x_2,
Output: distance as real
Begin
For i : 0 to x_1 .length-1 do
Sum: $sum + (x_1[i] - x_2[i])^2$
$\sqrt{sum}$
Return sum
End
End function

Figure 5: The get-distance function

5. Empirical Validation & Results Analysis

5.1. Empirical Validation

This section is intended to display the design specs and experimental results for constructing a situated agent (EBDI model) for an electronic health system, we test the proposed system and describe the results obtained by implementing the system using different parameters(sensors reading) in E-health.
The suggested system has been implemented in the provided prototype system utilizing the JADE agent platform, which is a distributed system implemented on a computer.

The goal of this suggested model is to filter the information that an agent perceives from the environment and then decide on a treatment plan based on the results of these filtering steps (which might include an alert if there is an aberrant state). So, this suggested system also has Machine-learning algorithms, such as the K-mean algorithm discussed previously, which are used to cluster this data, and membership function also used by this system.
Figure (6) describes the rustles for both agent's interfaces (monitoring and administrative) when executed our proposed system.

5.2. Results Analysis

We have used two ways to evaluate the proposed system; first way is by using a case study with real data and comparing the agent's behaviour with real data. In a second way, we formalize the Reward function somehow to calculate Reward values that represent like scores for agent's behaviour. In both previous ways, the proposed system proved excellent behaviour working correctly 100%.

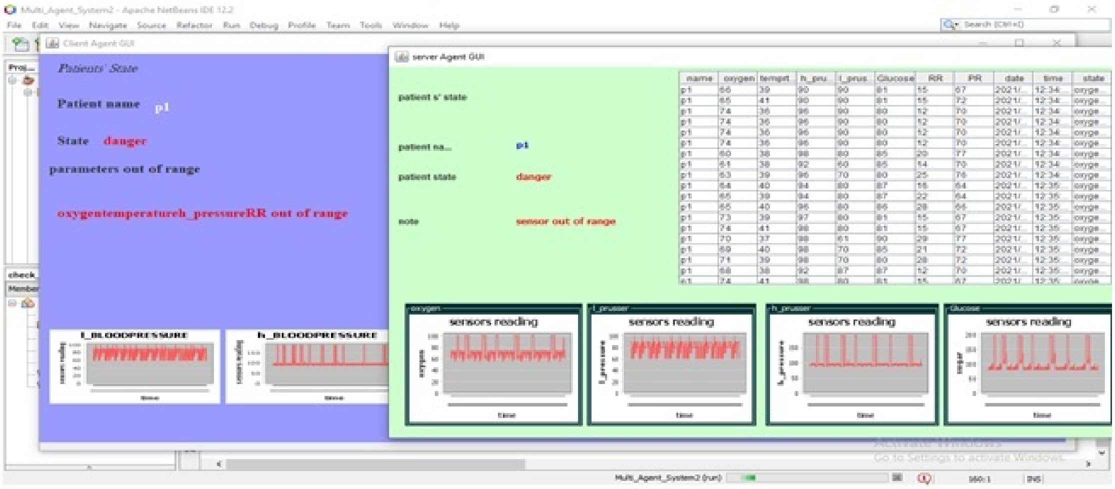

Figure 6: describes the monitoring and administrative agents' interfaces when executed the proposed system

5.2.1. *Case study*

We discuss the system in this section, which runs on the data in the specific size of the samples each sample has seven vital parameters (which are oxygen, temperature, high pressure, low pressure, glucose, heart rate, and Respiratory rate) shown in Figure (7). The data which used in this proposed system are collected from the hospital by tracking the vitals' parameters (explained previously) for n patients. Each patient was examined 4 times in a day and this tracking is continuous for many days also this study contains more than cases normal (which mean the sensors' reading in the natural range look to the Table(1)and Figure(8)) and also this study contains abnormal states (out of range). So, the system we suggested filters these values and this proposed system can decide which are normal and which abnormal sensor data are.

	id	date	tour time	oxgyen	temprtuer	H_prssuer	L_prssuer	Glucose	Respiratory Rate	Pulse Rate
1	1	20/4/2021	10:00am	66	39	90	90	81	15	67
2	1	20/4/2021	3:00 PM	65	41	90	90	81	15	72
3	1	20/4/2021	8:00 PM	74	36	96	90	80	12	70
4	1	20/4/2021	1:00 AM	60	38	98	80	85	20	77
5	1	21/4/2021	6:00 AM	61	38	92	60	85	14	70
6	1	21/4/2021	12:00 PM	63	39	96	70	80	25	76
7	1	21/4/2021	8:00 PM	64	40	94	80	87	16	64
8	1	21/4/2021	1:00 AM	65	39	94	80	87	22	64
9	1	22/4/2021	6:00am	65	40	96	80	86	28	66
10	1	22/4/2021	12:00 PM	73	39	97	80	81	15	67
11	1	22/4/2021	8:00 PM	74	41	98	80	81	15	67
12	1	22/4/2021	1:00 AM	70	37	98	61	90	29	77
13	1	23/4/2021	6:00 AM	69	40	98	70	85	21	72
14	1	23/4/2021	12:00 PM	71	39	98	70	80	28	72
15	1	23/4/2021	8:00 PM	68	38	92	87	87	12	70
16	1	23/4/2021	1:00 AM	59	40	93	86	84	20	77
17	1	24/4/2021	6:00 AM	67	40	94	60	86	14	70
18	1	24/4/2021	12:00 PM	72	41	95	70	81	15	76
19	1	24/4/2021	8:00 PM	66	36	96	80	85	19	64
20	1	24/4/2021	1:00 AM	62	38	97	90	88	22	66
21	1	25/4/2021	6:00 AM	74	37	98	70	80	25	67
22	1	25/4/2021	12:00 PM	66	41	99	100	95	20	72
23	1	25/4/2021	8:00 PM	55	41	95	84	81	26	70
24	1	25/4/2021	1:00 AM	61	40	90	90	81	22	77

Figure 7: Screenshot for the samples that are collected for n patients

Table 1: shows the natural range of parameters values

Parameter	*minimum value*	*maximum value*
Oxygen	*95*	*100*
temperature	*36*	*37*
H-pressure	*100*	*130*
L-pressure	*60*	*90*
Glucose	*80*	*120*
PR / HR	*60*	*100*
RR	*12*	*24*

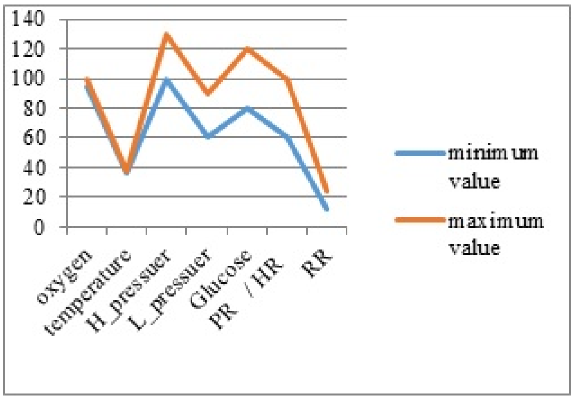

Figure 8: chart for the natural range of parameters values

5.2.2. Evaluation via Reward Function

We can formalize Reward function in this section to evaluate the proposed agent's behavior in previous case study more than one way: one of these ways is

- Reward Values for all Observations:
  We can formalize Reward function by generalize observation with agents' action for the same patient as the following:
  Let $x_i = x_1,\ x_2,\ x_3 \ldots .x_n$ {represent observations (vitals' parameters)for patient j}
  Let $[\ a_i, b_i]$ {the closed intervals that represented the minimum and Maximum accepted natural values for each sensor's reading}.
  where if $x_i \in [\ a_i,\ b_i]$ then the patient has a normal state (i.e. the patient in the safe health condition), so, the agent behavior will be keep observing without any alert message. And if $x_i \notin [a_i, b_i]$ then the patient is in an abnormal state (i.e. the patient in danger), so agent behavior will be sending an alert message to the server (Administrative agent)
  Also, we know the agent's behavior depending on the Boolean function Eq. (4.1) which is in simple way:

$$B = \begin{cases} \text{Act (keep observing)} \\ \\ \text{Del (send alert message)} \end{cases}$$

  Therefore in this way, we can calculate the reward function value as flowing:

  if $x_1 \in [\ a_i, bi]$, and $x_2 \in [\ a_i, bi]$,... and $x_n \in [\ a_i, bi]$, and $B =$ *keep observing* $\Rightarrow R = 1$

  if $x_1 \in [\ a_i, bi]$, and $x_2 \in [\ a_i, bi]$,... and $x_n \in [\ a_i, bi]$, and $B =$ *send alert message* $\Rightarrow R = -1$

  if $x_1 \notin [\ a_i, bi]$, or x2 $\notin [\ a_i, bi]$,... or $x_n \notin [\ a_i, bi]$, and $B =$ *keep observing* $\Rightarrow R = -1$

  if $x_1 \notin [\ a_i, bi]$, or $x_2 \notin [\ a_i, bi]$,... or x $n \notin [\ a_i, bi]$, and $B =$ *send alert message* $\Rightarrow R = 1$

The Rustles of Evolution by using this way to evaluate the proposed system's behavior. We have obtained the results which showed in figure (9) that represent screenshot for the evaluation for our proposed system where the test is applied on the samples. However, we can see that the reward value was 1 and this means that the agent is worked correctly 100%.

no	oxygen	Temperature	h-pressure	l-pressure	Glucose	RR	PR	check-action	reward-value
1	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	Keep observing	1
2	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	Keep observing	1
3	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	Send Alert message	1
4	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	Send Alert message	1
5	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	Send Alert message	1
6	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	Send Alert message	1
7	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	Keep observing	1
8	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	Send Alert message	1
9	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	Send Alert message	1
10	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	Send Alert message	1

Figure 9: Screenshot for result for tested and evaluated our system (implementing EBDI in E-Health)

Where true means that the sensor's reading value is in the natural range ,and false refers that the sensor's reading value is out of the natural range which shown in table 1.

6. Conclusion and future work

6.1. Conclusion

This section is shown the conclusions during designing and building the intelligent software agent implemented in the E-health system utilizing MAS:

The system was created by using AI (intelligent Software agent) with E-health. We have formalized the EBDI model that outperforms all deliberative situated agents in dynamic environments in E-Health. EBDI defines the degree of commitment and the intention reconsideration of the agent that guarantees the best policy in a dynamic environment .The designed system is based on the EBDI model, which is active in a dynamic environment, and it is expected to be a future alternative to the existing system for improving life quality.

We have built a model(multi-Agent-System) that can implement a software intelligent agent(EBDI model) for E-health to work as monitoring for patient's state which gives an alarm where there is a problem at any sensor reading (filtering a piece of information that an agent percept it from an environment). The method is meant to allow the doctor to make the best decision possible at the appropriate time, saving the patient's life. The technology is intended to keep track of patients' states in hospitals.

6.2. Suggestions for Future Work

The following are some recommendations for further work:
The GUI window appearance can be changed depending on the institution that will use the proposed system. Adding many more sensor readings to the system to improve it . Improving the system by designing one administrative agent as a server and many monitoring agents working in parallel in run time .

Also, a diagnosis system may be created, and the system can be made portable so that it can be utilized at any time and from any location while keeping track of the patients' condition. The system can be developed by using another function for clustering such as DBSCAN algorithms or FCM algorithms.

References

[1] F. Bergenti and A. Poggi, *Multi-agent systems for e-health: Recent projects and initiatives*, 10th Work. Objects Agents, WOA'09, 2009.

[2] M.E. Bratman, D.J. Israel and M.E. Pollack, *Plans and resource-bounded practical reasoning*, Comput. Intell., 4(3) (1988) 349–355.

[3] F. Cao, N. Archer and S. Poehlman, *An agent-based knowledge management framework for electronic health record interoperability*, J. Emerg. Technol. Web Intell. 1(2) (2009) 119–128.

[4] E. Domnori, G. Cabri and L. Leonardi, *Designing and implementing intelligent agents for e-health*, Proc. Int. Conf. Emerg. Intell. Data Web Technol. EIDWT 2011, 2011 pp. 78–85.

[5] A. Hussein Jabber and A. Obied, *Intelligent software agent in e-health system: Review*, J. Al-Qadisiyah Comput. Sci. Math. 13(1) (2021) 99–108.

[6] A. Hussein, *A Multi-agent System in E-Health SystemImplementing model*, Turk. J. Comput. Math. Educ. 12(14) (2021) 2845–2859.

[7] H.M. Hussain, K. Benkrid, H. Seker and A.T. Erdogan, *FPGA implementation of K-means algorithm for bioinformatics application: An accelerated approach to clustering microarray data*, Proc. 2011 NASA/ESA Conf. Adapt. Hardw. Syst. AHS 2011, 248–255.

[8] D. Isern, D. Sánchez and A. Moreno, *Agents applied in health care: A review*, Int. J. Med. Inf., 79(3) (2010) 145–166.

[9] H. Jiang, J. M. Vidal and M. N. Huhns, *EBDI: An architecture for emotional agents*, Proc. Int. Conf. Auton. Agents (2007) 38–40.

[10] H. Lieberman and C. Mason, *Intelligent agent software for medicine*, Stud. Health Technol. Inf. 80 (2002) 99–109.

[11] L. Males and S. Ribaric, *A model of extended BDI agent with autonomous entities (integrating autonomous entities within BDI agent)*, 2016 IEEE 8th Int. Conf. Intell. Syst. IS 2016 - Proc., no. ii, pp. 205–214.

[12] A. Obied, M.A. Hajer and A. H. Hasan, *Computerized situated agent as mediator in centralized computing market*, J. Adv. Res. Dyn. Control Syst. 11(2) (2019) 344–350.

[13] A. Obied, *Deliberative regulation with self-organising sensing*, Adv. Comput. 7(3) (2017) 80–94.

[14] A. Obied, A. Taleb-Bendiab and M. Randles, *Self-regulation in situated agents*, Int. Trans. Syst. Sci. Appl. 1(3) (2006) 1–20.

[15] Y. Sánchez, T. Coma, A. Aguelo and E. Cerezo, *ABC-EBDI: An affective framework for BDI agents*, Cogn. Syst. Res. 58 (2019) 195–216.

[16] D. Singh, L. Padgham and B. Logan, *Integrating BDI Agents with Agent-Based Simulation Platforms*, Auton. Agent. Multi. Agent. Syst. 30(6) (2016) 1050–1071.

[17] A. Sumari, *Information fusion in C4ISR framework: Concept and application in nation defence*, Int. Semin. C4I 2008 (2008).

[18] N. Vlassis, *A Concise Introduction to Multiagent Systems and Distributed AI*, Univ. Amsterdam, Amsterdam, 2003.

[19] P. Vora and B. Oza, *A survey on k-mean clustering and particle swarm optimization*, Int. J. Sci. Mod. Eng. 1(3) (2013) 24–26.

[20] J. Yadav and M. Sharma, *A Review of K-mean Algorithm*, Int. J. Engin. Trend. Tech. 4(7) (2013) 2972–2976.