Healthcare Expert System based on Group Cooperation Model*

Romeo Mark A. Mateo and Jaewan Lee
School of Electronic and Information Engineering, Kunsan National University
{rmmateo, jwlee}@kunsan.ac.kr

Bobby D. Gerardo
Institute of Information and Communications Technology
West Visayas State University
bobby_gerardo@engineer.com

Abstract

Expert systems for healthcare are widely studied where accuracy of diagnosis and efficiency of the system for various healthcare services are examined. Also, recent researches include decision support services, expert medical services and autonomous management, which are based on multi-agent systems. The cooperation of agents is crucial in analyzing, and managing the data of patient in order to detect abnormal patterns so that it can provide an advance treatment. This research shows the implementation of healthcare expert system (HES) based on group cooperation model. Communications through the components of the group cooperation model are designed based on multi-agent approach. The proposed agent managers effectively coordinate the processing of client requests. Replica managers provide quality of service for clients by replicating objects. An adaptive scheme using round robin and fuzzy least load algorithms is managed by load balancing service. And in this paper, we presented the implementation by using CORBA for distributed objects and Jade Framework for agents..

1. Introduction

Implementation of information technology in healthcare system is necessary in order to provide fast, effective and efficient services to the clients. Automated healthcare services provide information of the diagnosis and continuous monitoring of patient to acquire immediate response in order to save lives. Expert systems applied in healthcare are important in giving correct information for diagnosis and providing immediate medical services. Current research in decision support system (DSS) uses multi-agents approaches for the expert systems [1]. Also, algorithms that are accurate in classifying diseases are studied to be used in medical diagnosis. The neural network is one of the most commonly used techniques [2, 3]. Successful application examples show that neural diagnostic systems are better than human diagnostic capabilities. Neural network techniques are used to analyze medical images as demonstrated in [4, 5]. An ontology-based intelligent healthcare agent is used for the respiratory waveform recognition to assist the medical staff in judging the meaning of the graph reading from ventilators [6]. Neural network-based agents are used for discovering rules in medical database [7]. However, the classification of diseases depends on the tools and techniques that doctors and specialists are currently using to diagnose and treat

* This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce Industry and Energy of the Korean Government.
the patients. There are also varieties of data mining algorithms that are used to effectively help diagnose a patient. These distributed algorithms and tools are needed to interoperate and also give the expert a way to choose the appropriate algorithm on providing the diagnosis.

This research shows the object implementation of the healthcare expert system (HES) based on the group cooperation model. The proposed system consists of object groups where an expert chooses the objects that are appropriate for processing the request. A service is consisted of one or more objects where a single object is assumed to be a specific data mining tool. An agent manager is proposed to coordinate the processing of client requests and communicate with other agents for the efficient load distribution. Replica manager handles the creation and deletion of object replicas. Object replicas in several servers are used to provide quality of service. An adaptive load distribution scheme is managed by the agents of the load balancing service. The communication of the component is based on multi-agents for effective coordination of tasks.

2. Related works

2.1. Expert systems in healthcare

Expert systems are computer programs that are derived from a branch of computer science research called artificial intelligence [8] or popularly known as AI. The expert systems are reserved for programs whose knowledge base contains the knowledge used by human expert. AI is concerned with the concepts and methods of symbolic references, or reasoning, by a computer, and how the knowledge used to make those inferences will be represented within the machine. The area of human intellectual endeavor to be captured in an expert system is called task domain. Domain refers to the area within which the task is being performed. Task refers to some goal oriented, problem-solving activity. In the domain of healthcare, it is important that the system is accurate in diagnosing because it deals with lives of persons which can never be replaced if an error occurred. There are various techniques on implementing the expert system and almost uses accurate and established algorithms. Most of these use data mining techniques. In [9] summarizes the design of an expert system for the diagnosis of this health problem. The expert system knowledge base is composed of a set of production rules written in classic bi-valued logic and by a set of potential facts. A research study in [10] suggests a model of a chronic diseases prognosis and diagnosis system integrating data mining (DM) and case-based reasoning (CBR). Using data mining approach nuggets out potential knowledge from vast amounts of data [11]. The study explores the hidden knowledge among medical history data of developmentally-delayed children. A decision tree is constructed to classify delay levels of each type according to physical illness, and association rule is applied to locate correlations between cognitive, language, motor, and social emotional developmental delays.

The design of the proposed HES is an expert system for healthcare where the services are customized by choosing appropriate objects. For example, a cardiologist service needs data mining tool object to classify the client’s input on the previous history of other patients.

2.2. Group cooperation model

The specifications focus on object technology with a client-server model to provide a uniform view of an enterprise’s computing system and everything in the network is an object.
The necessity to manage these objects is modeled in the object groups to perform cooperatively for an efficient service is realized by many object designers and programmers. There are researches integrating their methods in the object group model. The object groups in the previous study [12] uses a cooperation model to implement the adaptive load balancing scheme which is based on two algorithms namely round robin and the fuzzy least load algorithm. Interaction of the group components are presented to interoperate from other distributed object group. The system analyzes the equal distribution of loads and it switches the algorithm to adapt the system requirements based on the threshold value. The object group modeling was suggested for providing the efficient and convenient management of distributed objects, and for making the individual object independent and reusable.

3. Group cooperation model for HES

Cooperation is the behavior of the system components working together for attaining the same goal. This paper proposes the cooperation model for the object groups to implement the load balancing of HES. The proposed healthcare expert system (HES) shown in Figure 1 is designed in group cooperation model.

![Figure 1. The global view of group cooperation model is presented in three layers.](image)

The global view of the proposed group cooperation model is consisted of three level parts. It also shows the intercommunication of the clients, object groups, load balancing service and server replicas. On the top-level, clients are represented as heterogeneous hardware like desktop PC, notebook and personal digital assistants (PDA). Clients are interconnected by a network and software components that communicate to object groups using the Object Request Broker (ORB). Object groups are shown in the middle-level part where it has the object references of each object in the server replicas. Objects included in the object group are used to process the client request. In this research, object groups are services for the healthcare expert system. These services
have sub-component which resides in the server replica shown in the bottom-level of the group cooperation model. Communication is managed by the agent managers or AMs. The bottom-level part consists of server replicas and load balancing service. These components are used to implement the load distribution and dynamic replication. Server replicas are managed by replica manager (RM). A server replica refers to a single server which contains variety of objects. The load balancing service is consisted of global load monitor, agent load analyzer and agent forwarder which manage the efficient load distributions to object groups.

![Figure 2. Client consultation process in a single service of HES](image)

Figure 2 shows the interaction of a client to a service. AM received the request and processes the information through the sub-component objects. After processing, a return output of consultation is sent to the client.

### 3.1. Components of group cooperation model

**Agent manager (AM)** – The AM is the main component of the service. This does all the management procedure. All subcomponent objects that are specified by an expert are connected to the AM. And other AM outside the group are coordinating to implement the efficient load distribution.

**Global load monitor** – monitors the loads. Each replica is registered to the global load monitor. Upon receiving the loads, the global load monitor checks the value of the current distribution to compare on the threshold value. If the load distribution is greater than the threshold, then it communicates to the AMs to switch the scheme from round robin to fuzzy least load algorithm [12]. The same procedure of switching from fuzzy least load to round-robin is done if the global load monitor determines that the load distribution is lower than the threshold.

**Agent load analyzer** – analyzes the load of each servers and objects. This is a sub-component of the load balancing service which intercepts every request to process the fuzzy least load algorithm for distribution of request to the replica objects and determine additional object replicas needed. By processing the values to the fuzzy system, it determines the appropriate least loaded object to forward the request. Finally, it sends the object reference and issue a location forward.

**Agent forwarder (AF)** – forwards the request to appropriate object. Every time the load distribution scheme switches to fuzzy least load, the agent forwarder determines which objects has least loaded and should be used for the request.
Replica manager (RM) – does the creation and deletion of objects in the server. The RMs within the server replicas is coordinating to other RM to manage the replicated objects. If object that changed it values then RM of that object communicates through other RM to change the value of the same object

Security component – checks the validity of the client. The security verifies if the client has the access privilege to the object. The proposed system processes this with the encryption module of the security. If the client is not allowed to access the object then it sends a message that it requires a valid user id and password or the request is not allowed at all. After the procedure, it sends message to AM the result of verification.

Healthcare expert service – composite of tasks which consist of a several object services to provide the service to client. In this research, a healthcare expert service is also considered as an object group. The service consists of AM, security and object services.

Object – sub-components of the services. These sub-components are requested by the AM to process the request. Moreover, the object references are managed by the AM to be grouped. The purpose of grouping the objects is to gather the related object based on the properties of the service. In the group cooperation model, each object group uses a plurality of request. This is a property of an object group to behave like a singleton object where it can act as an identifiable, encapsulated entity that may be invoked by a client.

3.2. Interactions of the group cooperation model’s components

The load balancing service consists of interactive components for intercepting and forwarding request. AM is the main receiver for client’s request. The default scheme of the load distribution is round robin. Each AM is aware of the current scheme it uses by sending message of their status. Once the AM receives the request, it sends messages to other AM. The content of the message is the location of the objects and server replica which it invokes currently. If another AM receives a request then it begins invoking the object that is not used by other object groups. Figure 3 shows the communication between the services.

Figure 3. Agent managers informs other agents of the current object it accessing

Upon the request, the object sends an update loads to the global load monitor. An increment of loads is done from the global load monitor and after the servicing the client, it sends again an update load to decrease the load. While the global load monitor is on the update procedure, it compares the status of the load distribution from the objects if it already
exceeds the threshold value. After invoking the object by the AM, the object sends a load update to global load monitor. If the value of load distribution is greater than the threshold then round robin scheme is switched to fuzzy least load [12]. At this point, request of AM is directed to the agent load analyzer where it determines the suitable object to forward and the procedure is done in the fuzzy system. Agent load analyzer issues LOCATION_FORWARD() reply to the AM request with the reference object. Client sends the request to the new reference.

Figure 4. Round robin accessing the objects from replica servers

Figure 4 presents the procedure of the round robin algorithm. Round robin is the default scheme of the system. The δ determines how well the loads are distributed to the objects and Φ is the threshold value for the load distribution. At this case, the system status implies that there is no need to forward the request because the system resources are utilized enough in distributing the loads. The switching of algorithm occurs only if the load distribution is greater than the threshold where the variance of the load distribution is large.

Figure 5. Fuzzy least load accessing the objects from replica servers

Figure 5 shows the case of the system switches to fuzzy least load algorithm. The ETD of analyzing the loads from each replica servers and the loads of objects is illustrated. Server loads refers to the number of current requests accessing the objects in the server while the object loads refers to the current request accessing the object. These loads are analyzed in the fuzzy system. After analyzing the most candidate object, the agent load analyzer sends location forward to the AM with the object reference of the most candidates. In fuzzy least
load scheme, the same procedure of updating the global load monitor is done. The analyzer continues to compare the threshold to the load distribution and when the threshold is greater the load distribution then the system scheme switches again to round robin.

4. Function analysis of the group cooperation model components

In the context of function analysis of multi-agents, an organization or a group can be seen as a system of roles. The interaction of the components from the group cooperation model is described in this section based on function analysis of multi-agent [13]. Table 1 shows the functions and links of each component.

<table>
<thead>
<tr>
<th>Agents/Actors</th>
<th>Functions</th>
<th>Communication links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent manager (AM)</td>
<td>object group manager</td>
<td>other AMs, AFA</td>
</tr>
<tr>
<td>Global load monitor</td>
<td>monitors the load of objects and servers</td>
<td>objects, ALA</td>
</tr>
<tr>
<td>Agent load analyzer (ALA)</td>
<td>analyses the load distribution</td>
<td>AMs, AFA, RM</td>
</tr>
<tr>
<td>Agent forwarder (AFA)</td>
<td>forwards the request</td>
<td>AMs, ALA</td>
</tr>
<tr>
<td>Replica manager (RM)</td>
<td>create additional objects</td>
<td>objects, ALA</td>
</tr>
</tbody>
</table>

There are two states defined in this paper, \( \Sigma = (\sigma_1, \sigma_2) \), where \( \sigma_1 \) means the load distribution is normal which implements the round robin scheme, and \( \sigma_2 \) means the variance of load distribution from each object replica is large in which fuzzy least load scheme is necessary. Round robin scheme is the procedure where each AM are accessing the objects alternately. Equation 1 presents the round robin algorithm.

\[
RR(x) = object_n
\] (1)

The object reference is chosen by AM using the method in Equation 1. The \( n \) is initialized at a value of 1 and appends it on the next object index by \( n = n + 1 \) every time a client accesses an object. If \( n \) is the last index then \( n = 1 \). Each AM is aware of the current scheme it uses by sending message of their status and using the round robin scheme. Once the AM receives the request, it sends messages to other AM. The first state is shown in Equation 2.

\[
\sigma_1 = AM_n \text{Exec}(RR(x), \text{Informs}(AM_1,\ldots,AM_m))
\] (2)

The execution of round robin and informing other AMs is expressed in Equation 2. Also, the content of the message includes location of the objects and server replica which it currently invokes. Upon the request, the object sends an update loads to the global load monitor. An increment of loads is done from the global load monitor and after servicing the client, it sends again an update load to decrease the load. While the global load monitor is on the update procedure, it compares the status of the load distribution from the objects if it already exceeds the threshold value. The \( \delta \) determines
the load distribution and $\Phi$ is the threshold value where $\Phi$ is configured by the administrator. The value of load distribution is calculated in Equation 3.

$$\delta = \frac{\sum \left( (R_n \times \mu_s_n) - \mu \right)^2}{N}$$  \hspace{1cm} (3)

This uses the number of the current loads in an object $n$ ($R_n$) and its mean service time ($\mu_s_n$) as variables for the standard deviation of the load distribution. If the value of load distribution is greater than the threshold then the round robin scheme is switched to the fuzzy least load scheme. The loads are analyzed using fuzzy method presented in Equation 4.

$$FF(x) = \min \{ \sum_{n=1}^{N} \mu_{SER_n}, \mu_{OBJ_n} \}$$  \hspace{1cm} (4)

Server loads refer to the number of current requests accessing the objects in the server and the object loads refer to the current request accessing the object are fuzzified by the functions $\mu_{SER}$ and $\mu_{OBJ}$, respectively. The second state intercepts the AMs by the agent forwarder expressed in Equation 5 in order to process the fuzzy least load algorithm. The agent forwarder issues LOCATION_FORWARD() reply to the AM request with the reference object based on the result from the agent analyzer.

$$\sigma_x = AM \_\_\_\_Intercepte d \_\_\_\_AF \rightarrow ALA(FF(x))$$  \hspace{1cm} (5)

After determining the best candidate object from the Equation 4, the analyzer sends location forward to the AM with the object reference of the candidate. The analyzer continues to compare and if the $\Phi > \delta$ then the system scheme switches to round robin.

4. Implementation

The experiment used the Borland Visibroker 7.0 to implement the proposed system which is CORBA compliance. The object group and the load balancing service component were developed in Java. The portable object adapter and portable interceptor were used to implement the load balancing service. The proposed fuzzy least load is encoded in the agent load analyzer where agent manager accessing the interface of agent load analyzer every time the load distribution is greater than the threshold. Figure 5 present the screenshots of the executing the agent manager (a) and replica manager (b), respectively.
Figure 5. (a) Running the agent manager and (b) replica manager

After initializing the AM, the load balancing scheme is operational in Figure 8. Figure 6 shows the client accessing the GM and a login interface of the HES for verification.

Figure 6. Input form of Healthcare Expert System for consultation

After the verification, the client has already verified and accessing the demo program of the HES is shown in Figure 7. The user is required to input his or her profile to be able to proceed for the consultation.

Figure 7. Information details about the input

Figure 8 shows the execution of the global load monitor. Each replicated object from the server replicas is registered to monitor each load. If the objects are requested by AM then it updates the load.
Figure 8. The global load monitor displays the current loads of objects being accessed

5. Conclusions

This research presented the group cooperation model for the healthcare expert system (HES) to implement the adaptive load balancing. The components of the group cooperation model were discussed to implement the efficient healthcare expert system. A single object group or service is managed by the agent manager where it chooses the necessary objects from the server. The global load monitor and agent load analyzer manage the adaptive load balancing scheme to implement the efficient distribution of loads. The replica manager is used to perform the replication of the object. The adaptive load balancing was based on two algorithms which are the round robin and fuzzy least load algorithm. The fuzzy least load algorithm is used to process the fuzzy decision of distributing the loads. This research study implemented the HES based on group cooperation model for efficient load distribution.

References

Authors

Romeo Mark A. Mateo

Romeo Mark A. Mateo received his B.S. degree in Information Technology from West Visayas State University, Philippines in 2004 and M.Eng. degree in Information and Telecommunications from Kunsan National University, South Korea in 2007. He is now a Ph.D. candidate in Information and Telecommunications Engineering at the Kunsan National University, South Korea. Also, he is currently working as research assistant at the Distributed Systems Laboratory (DSL). His research interests include distributed systems, ubiquitous computing, data mining and artificial intelligence.

Bobby D. Gerardo

Bobby D. Gerardo received his B.S. in Electrical Engineering degree from Western Institute of Technology, Philippines in 1989 and M.A.Ed. Mathematics degree from University of the Philippines, in 2001. He finished his Ph.D. in Engineering major in Information and Telecommunications from Kunsan National University, South Korea in August 2007. Currently, he is an assistant professor at the Institute of Information and Communications Technology, West Visayas State University, Iloilo City, Philippines. His research interests lie in the areas of distributed systems, telematics systems, mobile computing, data mining and ubiquitous computing.

Jaewan Lee

Jaewan Lee received his B.S., M.S., and Ph.D. degrees in Computer Engineering from Chung-Ang University in 1984, 1987, and 1992, respectively. Currently, he is a professor at the School of Electronic and Information Engineering and supervisor of the Distributed Systems Laboratory (DSL) in Kunsan National University, Kunsan City, South Korea. His research interests include distributed systems, database systems, data mining and computer networks.