*I.J. Information Technology and Computer Science,* **2015, 05, 48-54**
Published Online April 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.05.07

# Automatic Generation of Agents using Reusable Soft Computing Code Libraries to develop Multi Agent System for Healthcare

**Priti Srinivas Sajja**
Department of Computer Science, Sardar Patel University, India
Email: priti@pritisajja.info

***Abstract*—** This paper illustrates architecture for a multi agent system in healthcare domain. The architecture is generic and designed in form of multiple layers. One of the layers of the architecture contains many proactive, co-operative and intelligent agents such as resource management agent, query agent, pattern detection agent and patient management agent. Another layer of the architecture is a collection of libraries to auto-generate code for agents using soft computing techniques. At this stage, codes for artificial neural network and fuzzy logic are developed and encompassed in this layer. The agents use these codes for development of neural network, fuzzy logic or hybrid solutions such as neuro-fuzzy solution. Third layer encompasses knowledge base, metadata and other local databases. The multi layer architecture is supported by personalized user interfaces for friendly interaction with its users. The framework is generic, flexible, and designed for a distributed environment like the Web; with minor modifications it can be employed on grid or cloud platform. The paper also discusses detail design issues, suitable applications and future enhancement of the work.

***Index Terms*—** Multi Agent, Neural Network, Fuzzy Logic, Neuro-Fuzzy Hybridization, Healthcare

## I. Introduction

Information and Communication Technology (ICT) plays an important role in our day to day business as well as supports decision making in spectacular activities. It can play a major role in healthcare domain also. Medical diagnosis and healthcare related decisions are very sensitive and critical. Besides, the data relating to disease, drugs, symptoms and pathological tests come from various sources in different structures. The dynamic nature of such data further increases the complexity of the diagnosing and decision making. Now a days, though majority of the data is in electronic form, knowledge oriented decision making is lacking. Often such data are structured in a common, understandable format; analyzed and used manually for decision making. In this case the data is not used in effective manner for quality, safe, quick and reliable diagnosing as well as future use.

Though there are barriers such as cost and efforts required for implementing ICT, increased quality, usability and quick decision support accelerates ICT intervention in healthcare domain. It is obvious that the ability to obtain information to better manage their condition and to communicate with the health system can improve the efficiency and quality of diagnosing and care for patients, doctors and other experts in the field. In addition to this, the knowledge oriented access adds benefits of effective and quality decision and information accessibility for business. For such intelligent decisions, soft computing techniques are advisable.

The work illustrated in this paper is organized as follows. Section 2 of the paper presents brief literature survey and presents a report on work done so far in private as well as public health sectors. This section also presents a brief note on major activities observed in the domain and identifies requirement of the proposed work. Section 3 of the paper illustrates a generic multi agent system with three layers namely agent layer, library layer and knowledge layer. Design of each layer as well as working methodology is described with necessary information in this section. Section 4 demonstrates automatic generation of agents through libraries with help of use case diagrams, agent's generation process diagram and library structures with an example code. At the end, the paper concludes with applications and enhancement possible in future.

## II. Work Done So Far

Both the private as well as public health sectors welcome use of ICT in the domain. There are readymade solutions / packages, which can be customized and used for decision support in the domain. There are some tailor made software solutions developed considering a specific enterprise in the healthcare domain.

For disease diagnosing many neural network based solutions have been developed. Work done by Ganesan, Venkatesh, & Rama [1], Filippo et al. [2], Alkim, Gürbüz & Kiliç [3], Barwad, Dey & Susheilia [4], Catalogna et al. [5], Dey et al. [6], Elveren & Yumuşak [7], Er, Temurtas & Tanrıkulu [8], Sajja & Shah [9] etc. can be considered as the latest contribution towards disease diagnosing using intelligent techniques. These solutions emerged as stand alone solution for a specific disease. Work done by Swan [10], Frost & Massagli [11] etc. illustrates healthcare models using typical non intelligent ICT in diagnosing as well as clinical jobs.

Work proposed by Hernando et al. [12] uses agents for intelligent alarm for diabetes management. Work

 

presented in Giampiero et al. [13] illustrates the payoff method in the valuation of the cost-effectiveness of competing HPV immunization programs. State of the art survey for use of fuzzy logic in medicine is presented in the work of Mahfouf, Abbod & Linkens [14] and Massimo, Ivanoe & Giuseppe [15].

Majority of the readymade solutions and customized software allows to collect, store, retrieve, and transfer information electronically. Typically following major activities are considered for overall healthcare system [http://www.medpac.gov/publications%5Ccongressional_reports%5CJune04_ch7.pdf].

- administrative and financial facilities for billing, accounting, patient registration, personnel and payroll, inventory management and other administrative operations;
- clinical systems for drug entry, laboratory test results and analysis, electronic monitoring of patients (ICU), reporting, transcriptions, filmless imaging and diagnosing along with other clinical procedures; and
- infrastructure management such as network of computers, voice recognition, medical devices and information security systems.

Most of the information is static in nature. At different edge these information exists in different format; which requires some effort to make them uniform and accessible automatically. To enable such automatic accessing of the information in different structures and at various distributed resources, there is a need to have knowledge based mechanism. The knowledge about the information stored (metadata) and their structure might be helpful in achieving this task. Further, there is need of some independent and co-operative agents that perform various tasks on demand. These agents must be reusable, detachable and share common information/knowledge. For knowledge oriented behavior of these agents, intelligent techniques must be utilized. These requirement leads towards development of a generic framework of a multi agent system, which is discussed in next section.

## III. Generic Multi Agent System

To access information from various sources in different forms in a knowledge oriented manner, a general architecture is proposed in this section. The architecture is divided into three layers. The first layer is called **Agent Layer**. This layer comprises of multiple agents for different activities in a system. For each activity an agent is designed. These agents, in order to perform their intended activities use pre-developed code or tools provided by the **Library Layer**. The library layer may encompass some readymade tools to execute the task assigned to them. The third layer is the **Knowledge Layer**. The documented knowledge for a specific task, temporary results and log / history information and user profile are stored here. Fig. 1 illustrates the architecture in pictorial form.

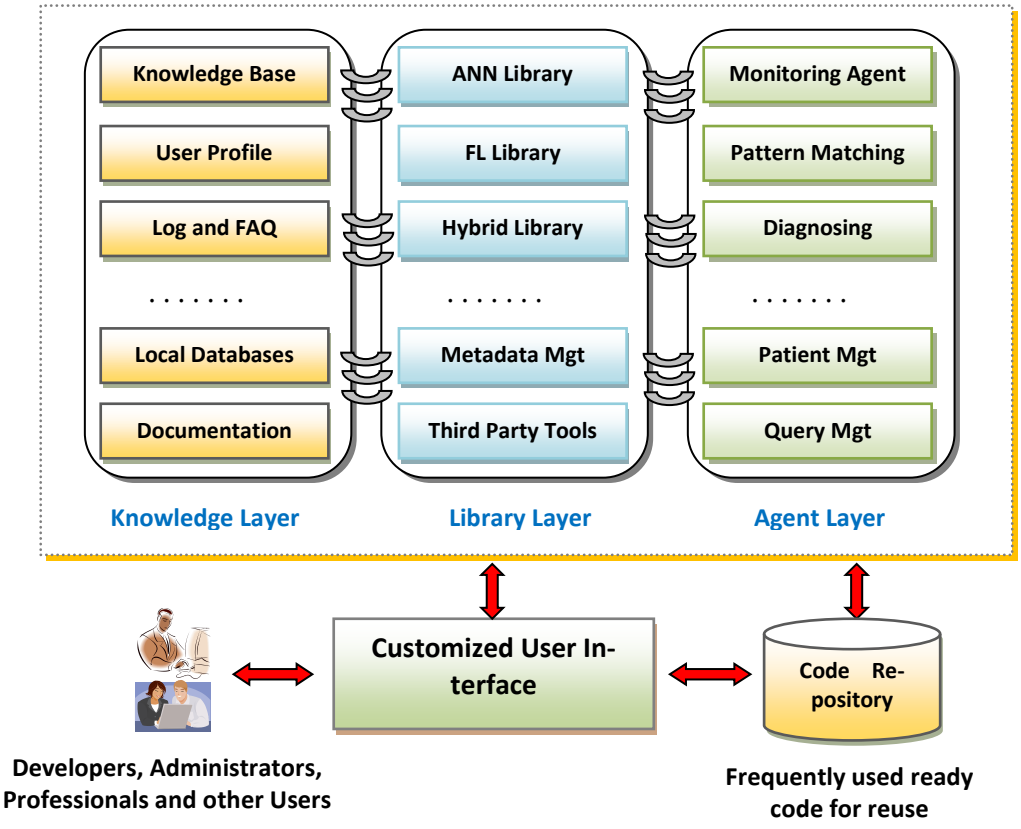


Fig. 1. Multi agent architecture of the generic healthcare system

*A. Agents in the Architecture*

For a healthcare system, some tasks are very basic and commonly required. Some of the tasks include resource management, patient management and querying. As stated in the previous section, one can plan for dedicated agents for administration, clinical job and infrastructure management. These jobs are common for any type of healthcare management system. Proposed architecture encompasses agents as listed below.

- Resource management – The agent considers common resources such as instruments, drugs inventory, faculty, supporting staff and other equipments. Each resource is codified and stored in a database. These databases may be distributed on platforms like the Web, grid or cloud. To access the information, the agent may need help of some protocols, web services or third party tools. The agent needs to identify and use the tool available in the library layer.
- Pattern matching – Once information as well as history about patient are registered and available through the patients database, patterns can be identified through the library tools such as neural network code. The agent in this case may have a friendly interface that asks necessary information about the pattern matching process. The agent then prepares an application specific code and documents it into the knowledge base for its future use.
- Patient management – This agent manages information about the registered patient in the system. Patient basic information such as name, address, age, etc. along with the complaints and symptoms are stored in various databases. The database may be appended with information such as diagnosis, history of the treatment and prescribed medicines along with the laboratory test results and medical images. This agent compiles patient information from various distributed databases. Given proper infrastructure and protocol support, these databases can be on private cloud of the organization for which the system is designed.
- Diagnosing agent – This agent works in collaboration with other agents in the architecture. Diagnosing is an expert task and needs expert level knowledge as well as complete information of patients. The agent uses information provided by patient management agent in the same layer. It may use other agents such as pattern matching agent or resources from other layers such as reusable library code from the library layer. Expert on duty (doctor) may use this agent as an intelligent assistant for the decision making process and cross verify the decision made. At this stage, any explanation and reasoning agent is not incorporated in the architecture and hence, it is not possible to have explanation and reasoning in detail for the decision taken. However, the framework is flexible enough to add the agent for the same.
- Query management – This agent uses the predefined fuzzy membership functions for interacting with experts, support staff and patients in more human way. It interacts with users to identify their information need and retrieves information from the databases as well knowledge base from one or more locations.

One may consider additional agents such as agents for security management for data as well infrastructure of the system, a mobile agent that installs some specific software to all remote computers (such as taking back up or checking virus), and an agent for video conferencing for guided surgery.

Communication between agents of the system is supported through a declarative communication language like KQML [16]. It is conceived both as a message format and a message handling protocol to support run-time knowledge sharing among agents [17]. An example of KQML block is shown below.

```
(register
: sender agent_Knowledge_Mgt
: receiver agent_Rule_Induction
: reply-with message
: language common_language
: ontology common_ontology
: content “content.data”
)
```

*B. Library Layer of the Architecture*

The library layer includes reusable code to develop software on demand in dynamic manner. Library codes are of two categories. The first category consist of executable software that agent can use to access for completion of its tasks. For example, fuzzification and defuzzification method provided by Fuzzy Logic (FL) based library. These methods are well defined and once selected it provides result based on the data provided. Another category of the library code is dynamic in nature, which is used for query generation and neural network generation to employ learning and training of the network created. Suppose the pattern matching agent needs artificial neural network to be developed, the ANN library can be used. The agent needs to provide information, such as which type of neural network is to be created, what would be the learning algorithm and where the training data set is available, if required. The neural network code will be provided to the agent for pattern matching. At the end, the code generated can be placed in knowledge base.

Since the fuzzy logic and artificial neural network code libraries are available in ready fashion, they can be reused on requirement to generate highly complex and hybrid functions. That is, the pattern matching agent may use fuzzy logic to interact with its users and get data/information in natural language. The information can be converted into non-fuzzy (crisp) information suitable to the neural network. By co-operative mixing of the code, an agent may employ a hybrid neuro-fuzzy approach. Fig. 2 and Fig. 3 illustrate the FL and ANN library components respectively. As shown in these

 

figures, besides the standard components and techniques, some innovative techniques such as novel activation functions are also contributed in FL as well as ANN [18]. In FL, type-2 fuzzy logic is also incorporated along with type reduce facility. For ANN, novel activation functions and modified learning algorithms are suggested and implemented.

Most of the library code designed here are inspired from the mother nature (bio-inspired) and known as soft computing techniques. Depending on the requirement, libraries of rough set, statistical analysis, ant colony optimization, swarm intelligence and genetic algorithm, etc. can be developed and accommodated into the architecture. The library also includes metadata management and third party tools such as protocols and web services.

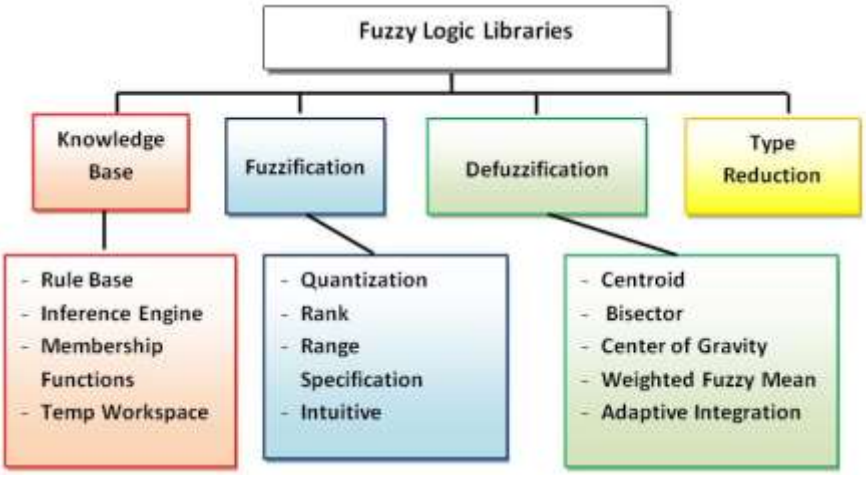


Fig. 2. Fuzzy logic library components

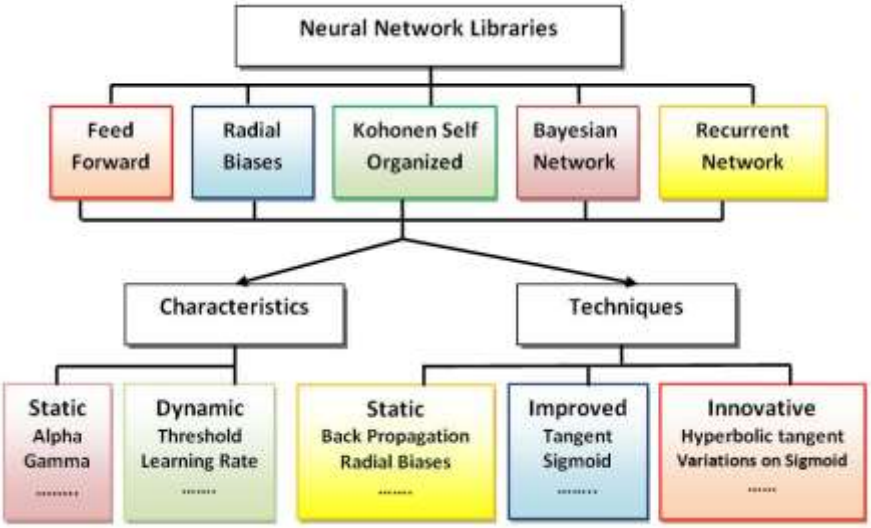


Fig. 3. Artificial neural network library components

*C. Knowledge Layer of the Architecture*

Agents used in the architecture designed here are developed for an integrated healthcare system in a specific domain for a given organization. Since they are contributing towards a common objective and goal, they must share some information and resources. Further, some meta-knowledge is also required. The meta-knowledge as well as knowledge generated through the agent can be documented in the knowledge base. The knowledge layer consists of knowledge base as well as databases in distributed fashion. User profiles, user manuals and frequently asked queries can also be stored for ready use.

User profile plays an important role in customized representation of the information. User profile normally consists of information such as user's identification, basic information, user type (access rights), current status, location, interest and type of job. If required, user profile may store fuzzy information also. Depending on the user profile, tailor designed information is presented to the user.

A sample user profile [19] is given below:

```
< profile class="user_profile">
        <user_name = "myname"/>
        <user_age = "myage" />
        <user_jobtype = "myjobtype" />
```

<user_mail = “mymail” />
<user_purpose = “mypurpose” />
. . .
</profile>

## IV. Automatic Generation Of Agents Using Libraries

The agents in the proposed multi agent system are generated with the help of the soft computing code librates designed. Structure of the libraries is discussed in section 3 of this article. When the framework of agents is provided to the system through the customized interface by user with administrative rights, the control agent takes the charge for agent’s generation and call for execution. The customized user interface uses the user profile to know about its users. Other information about the requirements regarding agent can also be acquired through the customized interface. The agent server facilitates use of the soft computing libraries to generate appropriate code. For example, to generate diagnosing agent, the customized interface acquires related information regarding the narrow domain of diagnosing, critical parameters for the rule generation, definition of membership functions, if any, and choices regarding inference engine conflict resolution strategy. It is not necessary to generate code for agent every time, but once. Many times popular task such as patter matching does not lead to the dynamic generation of a neural network based agent. In this case, the system uses ready code from the cache maintained by the system as code repositories. One may think to develop ready components such as a class file in appropriate format, which can be used with little modification.

Fig. 4 illustrates sample use case diagram from the system.

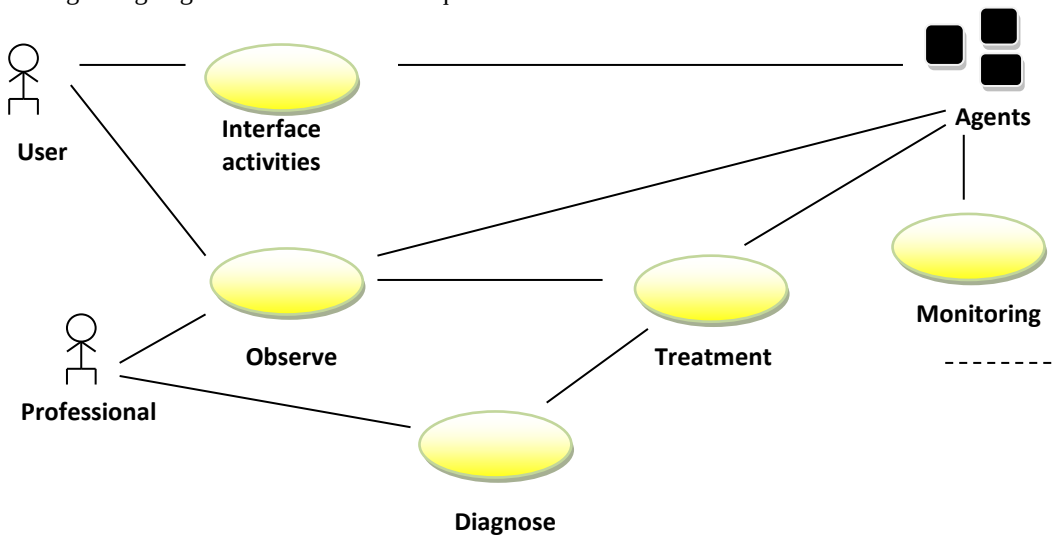


Fig. 4. Sample use case diagram of the system

Following is the sample code generated for neural network based agent by the proposed system [20].

```
public class My_Net
{  public Lyrs[] lyrs;
public int ni;
public LrngAlgo la;
public int no_inputs;
{ get { return ni; } ……. }
public int no_outputs;
{ { return lyrs[N_Lyrss - 1].N_Neurons; } …….    }
public int N_Lyrss
{ get { return lyrs.Length; } }
public LrngAlgo LearningAlg
{ { return la; }
set { la = (value != null) ? value : la; } }
public Lyrs this[int n]
{ { return lyrs[n]; }}
public My_Net(int inputs, int[] lyrs_desc, ActiFunction
n_actsig, LrngAlgo learn)
{    if (lyrs_desc.Length < 1)
   throw new Exception("To create neural network at-
least 1 layer should be provided");
   if (inputs < 1)
   throw new Exception("To create neural network at-
least 1 layer should be provided");
   la = learn; ni = inputs;    lyrs = new
Lyrs[lyrs_desc.Length];
   lyrs[0] = new Lyrs(lyrs_desc[0], ni);
   for (int i = 1; i < lyrs_desc.Length; i++)
   lyrs[i] = new Lyrs(lyrs_desc[i], lyrs_desc[i - 1],
n_act);  ….} }
   public void rndmizeWeight()
   { foreach (Lyrs l in lyrs)
   l.rndmizeWeight(); }
   public void rndmizeThreshold()
   { foreach (Lyrs l in lyrs)
   l.rndmizeThreshold();  }
   public void rndmizeAll()
   { foreach (Lyrs l in lyrs)   l.rndmizeAll();  }
   public void setActiFunction(ActiFunction f)
   { foreach (Lyrs l in lyrs)   l.setActiFunction(f);  }
```

```
{ foreach (Lyrs l in lyrs)   l.setRndmInt(min, max); }
public float[] Output(float[] input)
{ (input.Length != ni)
throw new Exception("Exceeding size limit");
float[] fnlresult;
result = lyrs[0].Output(input);
for (int i = 1; i < N_Lyrss; i++)
fnlresult = lyrs[i].Output(result);
return fnlresult;
}
}  }
```

The above code from the designed soft computing libraries can generate an artificial neural network. It is necessary to provide information such as number of input nodes, output nodes, number of layers, and training set etc. through the user interface.

Fig. 5 illustrates the agent generation process as discussed above.

Once agents are created, they need to be verified by the host or administrator for reliable and guaranteed execution. Once thorough testing is done, the code of agent is entered into the code repository for future use.

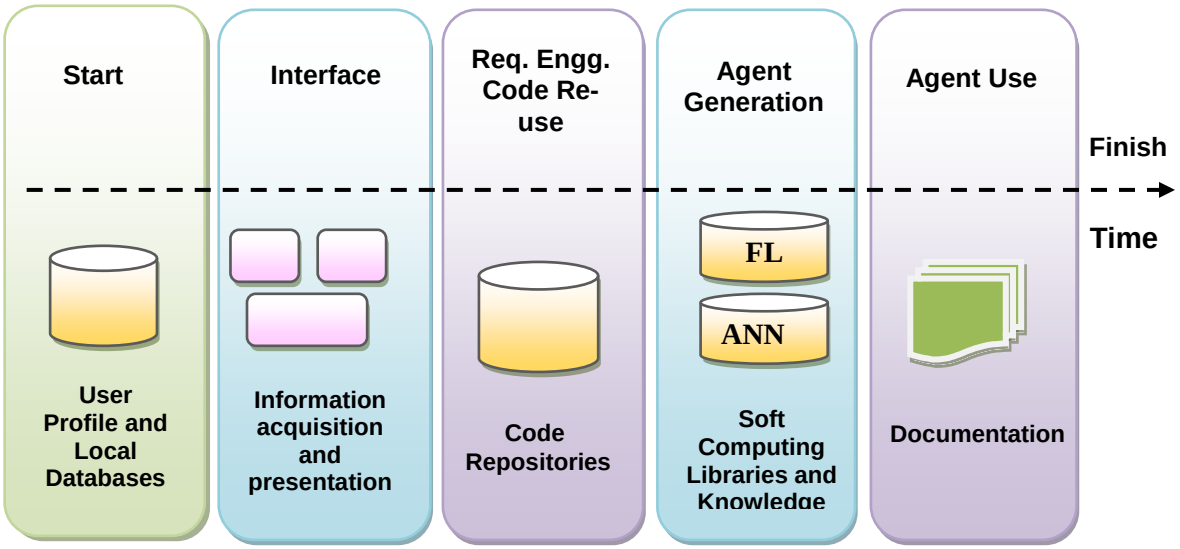


Fig 5. Agent generation process

## V. Conclusion

The system based on this architecture is agent based system comprising of multiple agents dedicated for a predefined tasks. Since agents are involved in the architecture; the system provides all benefits of agents as listed below.

- Autonomy – The agents are autonomous and do their jobs in autonomous manner.
- Co-operation – To complete the given job, an agent may interact with its environment as well as other agent.
- Learning – The agents may use the knowledge base and metadata management facility as well as the soft computing code library to learn from cases and data.

Besides these basic advantages of the agent based technology, the system also provides modularity, reusability and flexibility to add some more typical of hybrid agents.

The proposed system also offers advantages related to the knowledge oriented information access and diagnosing. It will provide an opportunity to document knowledge in electronic form; which can be used in future.

The proposed architecture can be used on a cloud or grid managed by an organization or a public sector. An extra step to manage necessary web service or third party tool for enabling the facility is required.

The architecture described here is general purpose and used for an integrated healthcare. However, the architecture can be used for only infrastructural management diagnosing tasks by incorporating suitable agents.

## References

[1] N. Ganesan, K. Venkatesh, and M.A. Rama. "Application of neural networks in diagnosing cancer disease using demographic data," *International Journal of Computer Applications*, vol.1(26), pp.76-85, 2010.

[2] A. Filippo, L. Alberto, M. Eladia, M. Peña, V. Petr, H. Aleš, and H. Josef. "Artificial neural networks in medical diagnosis," *Journal of Applied Biomedicine*, vol.11, pp.47-58, 2013.

[3] E. Alkim, E. Gürbüz, and E. Kiliç "A fast and adaptive automated disease diagnosis method with an innovative neural network model," *Neural Networks*, vol. 33, pp.88-96, 2012.

[4] A. Barwad, P. Dey, and S. Susheilia. "Artificial neural network in diagnosis of metastatic carcinoma in effusion cytology," *Cytometry B Clyn Cytom*, vol. 82, pp.107-111, 2012.

[5] M. Catalogna, E. Cohen, S. Fishman, Z. Halpern, U. Nevo, and E. Ben-Jacob. "Artificial neural networks based controller for glucose monitoring during clamp test," *PloS One*. Vol.7(8), pp.e44587, 2012.

 

[6] P. Dey, A. Lamba, S. Kumari, and N. Marwaha. "Application of an artificial neural network in the prognosis of chronic myeloid leukemia," *Anal Quant Cytol Histol*, vol.33, pp.335-339, 2012.
[7] E. Elveren, and N. Yumuşak. "Tuberculosis disease diagnosis using artificial neural network trained with genetic algorithm," *International Journal of Medical Systems*, vol.35, pp.329-332, 2011.
[8] O. Er, F. Temurtas, and A. Tanrıkulu. "Tuberculosis disease diagnosis using artificial neural networks," *International Journal of Medical Systems*, vol.34, pp.299-302, 2008.
[9] P.S. Sajja, and D.M. Shah. "Knowledge based diagnosis of abdomen pain using fuzzy prolog rules," *Journal of Emerging Trends in Computing and Information Sciences*, vol.1(2), pp.55-60, 2010.
[10] M. Swan. "Emerging patient-driven health care models: An examination of health social networks, consumer personalized medicine and quantified self-tracking," *International Journal of Environmental Research and Public Health*, vol.6(2), pp.492-525, 2009.
[11] J.H. Frost, and M.P. Massagli. "Social uses of personal health information within patients like me, an online patient community: What can happen when patients have access to one another's data," *Journal of Medical Internet Research*, vol.10(3), e15, 2008.
[12] M.E. Hernando, G. Garcia, E.J. Gomez, and F.D. Pozo. "Intelligent alarms integrated in a multi-agent architecture for diabetes management," *Transactions of the Institute of Measurement and Control*, vol.26(3), pp.185-200, 2004.
[13] F. Giampiero, B. Gianluca, C. Alessandro, M. Andrea, and M. Francesco. "A novel method to value real options in health care: The case of a multicohort human papillomavirus vaccination strategy," *Clinical Therapeutics*, vol.35(7), pp.904-914, 2013.
[14] M. Mahfouf, M.F. Abbod, and D.A. Linkens. "A survey of fuzzy logic monitoring and control utilisation in medicine," *Artificial Intelligence in Medicine*, vol.21(1-3), pp.27-42, 2001.
[15] E. Massimo, F. Ivanoe, and P. Giuseppe. "An evolutionary-fuzzy DSS for assessing health status in multiple sclerosis disease," *International Journal of Medical Informatics*, vol.80(12), pp.e245–e254, 2011.
[16] M. Genesereth, and R. Fikes. "*Knowledge interchange format,"* Version 3.0 Reference Manual, Technical Report Logic 92-1, Computer Science Department, Stanford University, 1992.
[17] T. Finin, Y. Labrou, and J. Mayfield. "KQML as an agent communication language," in *Software agents*, J. M. Bradshaw, Ed. CA: AAAI Press, 1997, pp. 291-316.
[18] J.A. Trivedi, and P.S. Sajja. "Framework for automatic development of type-2 fuzzy, neuro and neuro-fuzzy systems," *International Journal of Advanced Computer Science and Applications*, vol.2(1), pp.131-137, 2011.
[19] P.S. Sajja. "Multiagent knowledge-based system accessing distributed resources on knowledge grid," in *Knowledge Discovery Practices and Emerging Applications of Data Mining: Trends and New Domains*, A.V. Senthilkumar, Ed. Hershey, PA: IGI Global Book Publishing, 2010, pp.244-265.
[20] P. S. Sajja. "Research Directions in New Artificial Intelligence: A Case of Neuro-fuzzy System for Web Mining" in *Prajna: Journal of Pure and Applied Science*, vol. 21, 2013.

**Author's Profile**

**Priti Srinivas Sajja** is working as a Professor at Department of Computer Science, Sardar Patel University, India since 1994. She specializes in Artificial Intelligence especially in Knowledge-Based Systems, Soft Computing and Multi-Agent Systems. She is co-author of Knowledge-Based Systems (2009) and Intelligent Technologies for Web Applications (2012). She was Principal Investigator of a major research project funded by UGC, India. She has 140 publications in books, book chapters, journals, and in the proceedings of national and international conferences. Her five publications have won best research paper awards.

**How to cite this paper:** Priti Srinivas Sajja,"Automatic Generation of Agents using Reusable Soft Computing Code Libraries to develop Multi Agent System for Healthcare", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.5, pp.48-54, 2015. DOI: 10.5815/ijitcs.2015.05.07