World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:12, 2011

Web Server with Multi-Agent Support for Medical Practitioners by JADE Technology

O. Saravanan, A. Nagappan, P. Gnanasekar, S. Sharavanan, D. Vinodkumar, T. Elayabharathi and G. Karthik

***Abstract*—**The multi-agent system for processing Bio-signals will help the medical practitioners to have a standard examination procedure stored in web server. Web Servers supporting any standard Search Engine follow all possible combinations of the search keywords as an input by the user to a Search Engine. As a result, a huge number of Web-pages are shown in the Web browser. It also helps the medical practitioner to interact with the expert in the field his need in order to make a proper judgment in the diagnosis phase [3].A web server uses a web server plug in to establish and maintained the medical practitioner to make a fast analysis. If the user uses the web server client can get a related data requesting their search. DB agent, EEG / ECG / EMG agents' user placed with difficult aspects for updating medical information's in web server.

***Keywords*—**DB agent, EEG, ECG, EMG, Web server agent, JADE

I. Introduction

THIS deficit has been pointed out as one of the main barriers to the large-scale uptake of agent knowledge. Thus, the constant development and alteration of methodologies for the development of multi-agent systems is imperative, and consequently, an area of agent technology deserving significant attention. The proposed methodology for multi-agent system does not attempt to extend object-oriented techniques, instead focusing on agents specifically and the concepts provided by the agent model and including the web server agent.Agent can be defined as a part that, given a goal could act in the place of a user within its area knowledge. Agents are also called smart agents, as intellect is a key component of agency. Agent oriented approach can be viewed as next step of Object Oriented approach. The paper attempts to demonstrate the concept of developing Multi-Agent platform for processing of Bio-signals. It also demonstrates the concept of developing agents using JADE – Java Agent Development framework. Though agent technology provides a means to effectively solve problems in certain application areas, where other techniques may be deemed lacking or cumbersome, there is a current lack of mature agent-based software development methodologies.

The proposed methodology for multi-agent system does not attempt to extend object- oriented techniques and web server, instead focusing on agents specifically and the abstractions provided by the agent paradigm. Furthermore, it combines a top-down and bottom-up approach so that both existing system capabilities and the applications overall needs can be accounted for. As mentioned above, not explicitly accounting for existing systems is a point lacking in many of the currently available methodologies for multi-agent system development [4]. The proposed methodology attempts to formalize the analysis and design phases of the agent- based software development life cycle.

Essentially, JADE is a middle-ware, which simplifies the implementation of multi-agent systems by providing a set of graphical tools that support the debugging and deployment phases. The agent platform can be distributed across multiple machines, regardless of the underlying operating system, and the configuration controlled via a remote graphical user interface [1]. By specifically focusing on the JADE platform in the design phase, the process can move straight to implementation afterwards, without having to tediously adapt the results of the design phase to an agent platform of choice. This will obviously result in significant time gains for the design phase, in addition to providing with a much clearer picture on how to progress in implementation.

II. Literature Survey

A. *Merits of Jade*

The features of JADE that offers to the agent programmer:

The JADE schedules the agent behaviors in a non-preemptive fashion. Intra-platform agent mobility, including transfer of both the state and the code (when necessary) of the agent. Distributed agent platform.The agent platform can be split among several hosts.

- Graphical user interface to manage several agents and agent containers from a remote host.
- Debugging tools to help in developing multi agents applications based on JADE.
- Support for application-defined content languages and ontologies in Process Interface to allow external applications to launch autonomous agents.

O. Saravanan is a research Scholar in the Department of Computer Science & Engineering at the Vinayaka Missions Research Foundation Deemed University, Salem, Tamilnadu, India. e-mail: osaravanan.it@gmail.com

A. Nagappan is the Principal of VMKV Engineering College, Salem, Tamilnadu, India. e-mail: nags_slm@yahoo.com.

P. Gnanasekar is a research Scholar in the Department of Computer Science & Engineering at the Vinayaka Missions Research Foundation Deemed University, Salem, Tamilnadu, India. e-mail: peegeeyes@gmail.com.

S. Sharavanan is a research Scholar in the Department of Computer Science & Engineering at the Vinayaka Missions Research Foundation Deemed University, Salem, Tamilnadu, India. e-mail: sharavanan33@gmail.com.

D. Vinodkumar is a research Scholar in the Department of Computer Science & Engineering at the Vinayaka Missions Research Foundation Deemed University, Salem, Tamilnadu, India. e-mail: vino.kd@gmail.com

T. Elayabarathi is a research Scholar in the Department of Computer Science & Engineering at the Vinayaka Missions Research Foundation Deemed University, Salem, Tamilnadu, India. e-mail: bharathieee2k6@gmail.com

G. Karthik is a research Scholar in the Department of Computer Science & Engineering at the Vinayaka Missions Research Foundation Deemed University, Salem, Tamilnadu, India. e-mail: gkarthikme@gmail.com

B. Bio-Signal Processing

It demonstrates the concept of developing agents using JADE – Java Agent Development framework [7]. The agents are trained, intelligent system that is capable of setting up the platform for processing the multi-agent system and web server. The agents themselves communicate with each other in decision making process.

The methodological goal is to develop a multi agent platform for processing of bio-signals aiming at assisting medical practitioners in developing standard examination procedures.

The Generic Agent in turn will search for the Specific Agent –based on the signal type on the network and if found, the corresponding information will be passed to the specific agent by the Generic Agent. The medical practitioner wants to have an expert estimation about the multi-agent system; Generic Agent can be invoked to which he has to specify the SSN (Social Security Number) of the patient, the type of the signal and the corresponding data file [7]. For example the EMG medical practitioner wishes to have an expert opinion, the EMG Agent with all necessary information, will look for an EMG Expert System. Getting the Expert knowledge, the interpretation will be sent back to the Generic Agent through EMG Agent [8]. The expert opinion will be displayed on the user side as well as it will be stored in Database by DB Agent for further references.

III. Proposed work

The proposed work to clarify the problem without any (or minimal) concerns about the solution. In the proposed methodology, the analysis phase is carried out through a number of steps, described in the following sections.

A. Procedures

To capture the possible practical requirements of a new system is Use cases and also this system is most effective and efficient way. Each use case presents one or more states that express how the system should interact with the end user or another system to achieve a specific goal. There are a number of standards for representing use cases. [2] the most popular is the Unified Modeling Language (UML). This defines a graphical notation. Though use cases are used extensively by object-oriented practitioners, their applicability is not restricted to object oriented systems, because they are not object orientated in nature Hampton et al. Finally the information's are stored in web server. Hence, it is also possible to apply use cases to capture the functional requirements of multi-agent systems.

The multi-agent circumstances given in the previous section and after discussion the potential system users, it is possible to build up a prelude list of possible circumstances. When want the any information that time access the web server. Easley find out the information. Accordingly, the use cases are defined, and a use case diagram produced as shown in Fig 1.

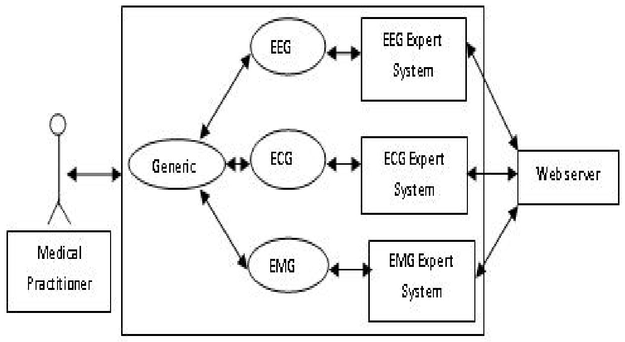

Fig. 1 System Model

B. Identifying the System Of Agent

In this step we are going to identify the subsequent formation of a first draft of the agent diagram, and also identify the main agent type system. Add one type of agent per user/device. Add one type of agent per resource. The agent diagram includes the following elements:

People that must interact with the system under development, represented by the UML actor symbol, the external systems that must interact with the system under development, represented by rectangles is the resources for the agent system. The agent types, represented by circles.

It also represents by an arrow linking instances of the above elements, specifying that the linked elements will interact in some way while the system is in operation. Note that, at this stage, only acquaintances between agents and resources / humans are shown in the agent diagram which makes a connection between two agents.

It should be noted that in the agent diagram, unlike UML use case diagrams, a distinction is made explicitly between humans and external systems. Interacting with a human through a user interface presents additional problems with respect to interacting with an external system.

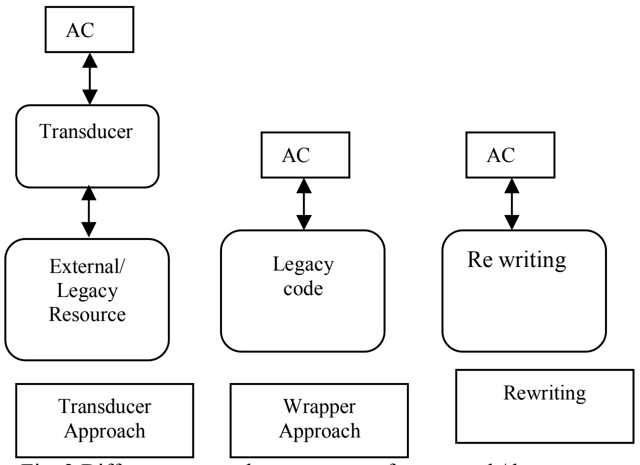

Fig. 2 Different approaches to account for external / legacy systems

The legacy way of systems and people that interact with the agents are accounted for in a multi-agent system, is an important consideration.An interface agent: The transducer agent serves an interface between a legacy system and the other agents in the system.

Package insertion**:** A code is injected into the legacy resource, provided the legacy resource's code is available. This inserted code will allow the resource to communicate in agent communication language, thus, converting it into an agent.Modifying the code: It involves rewriting the code to mimic the operation and capabilities of the legacy resource, but with the added ability to communicate in agent communication language.

In the agent diagram produced in this step (see Fig 2), the agents are acting as transducers, i.e. as an interface between the external/legacy systems/people, and the other agents in the system. Transducers are seen, in general, as the most practical and efficient method for accounting for legacy systems and are advocated in the proposed methodology. The reason is that by treating the legacy systems as a black-box, there is no need to tamper with or rewrite code, thus providing a quick means to get that resource functioning as part of the multi-agent system. However, in some cases a wrapper may be more relevant, and to an extreme rewriting, but such considerations should be deferred to the design stage.

C. Identifying the Process

For each agent identified, an initial list is made of its main responsibilities in an informal and intuitive way. Derive the initial set of responsibilities from the use cases identified. Consider the agents where these responsibilities are clearer first and delay the identification of responsibilities for other agents to later steps.

The existing methodologies Atomic roles[2] are initially identified and then possibly merged into agent types. However, this approach is considered less intuitive because in some cases it may become difficult to determine how the atomic roles should be aggregated into agent types, i.e. how many agent types there should be and which type should cover which atomic role(s). The definition of agent types then responsibilities, as in the proposed methodology, removes this ambiguity.

D. Identifying the Linkage

The focus is on who needs to interrelate with whom and the agent diagram is updated by adding proper acquaintance relations connecting agents that need to have one or more interactions. The term acquaintance comes from Gaia Wooldridge and Jennings, and is used in the same sense in the proposed system.

An obvious acquaintance relation in the multi-agent system is required between different agents: the user and the provider. Then, since a Generic agent must present the detailed report to its user and this information is stored in the database and made available by the relevant provider agents, there will certainly be an acquaintance relation between the Generic agent and the DB agent. Thus, going one step backward, to Step 3, some new responsibilities can be added to the Generic agent and the DB agent.

E. Developing the Agent

The set of agent types initially identified are refined by applying a number of considerations. These are related to:

Support: what supporting information agents need to accomplish their responsibilities, and how, when and where is this information generated/stored in web server. Once the Generic Agent is fed with the required details, it in turn searches for the specific agent to pass the required messages.

Discovery: how agents linked by an acquaintance relation discover each other. In the system the processing agents are named after the type of signal it processes – EEG Agent, ECG Agent, and EMG Agent.

Management and monitoring: is the system required to keep track of existing agents, or the starting and stopping of agents on demand.

F. Web Server

In this step, the application server works with a web server to handle request for dynamic content such as servlet, from web applications. A web server uses a web server plug in to establish and maintained the medical practitioner to make a fast analysis. The multi-agent system for processing Bio-signals will help the medical practitioners to have a standard examination procedure in online.

1. Data Transfer

In this upcoming world and in the present world, the entire in running only by the thing (i.e.) Data Transfer. Without this data transfer no things will be passed or transferred. At the present World the entire world depends upon this Data Transfer alone. The stored file of any agent, containing the user information at client side is being transferred to the server-end via Data Transfer. It uses the network (LAN/WAN/WWW) connection for transmission of data file. The well known 'GET' and/or 'POST' method [5] has been used to transmit client data to server through Web browser.

2. Data security

Most of the agents use to store the data's to be as secured manner in the database. But most of the data's in databases are Trustworthy. But when the agents use Data security in the sense, no third parties or unauthorized persons will be allowed to access the Data security.

3. The monitoring

The other agent types can be added to address issues such as monitoring agent faults and restoring them, creation of supporting agents that are needed only under certain conditions, or providing presence information. Having refined the set of agent types, the process is to go back previous work and iterate until adequately detailed descriptions of the agent types, their responsibilities, and acquaintance relations, respectively, are reached.

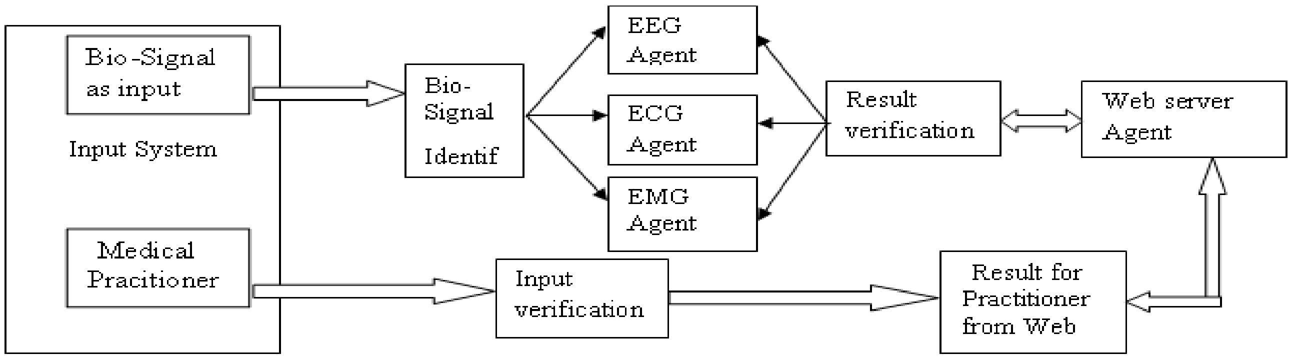

Fig. 3 The proposed system architecture

IV. Design

Here we're passing the input in two different ways. They are Bio-signal and Medical Practitioner. Then the Bio-signal input identifies the multi agent systems (EEG Agent, ECG Agent and EMG Agent). Whereas the web server agent contains the solution in order to verify the results that whether it's EEG or ECG or EMG agents. Another way to find, (i.e.) the medical practitioner verifies the agent in the beginning and finds from the web. One can move between the two. Since from this point on, the proposed methodology focuses on the JADE platform (and hence, the constructs provided by it) with web server. Carrying out the design phase allows reaching a level of detail that is sufficient enough to have an implementation, with the possibility of a significant amount of code being generated.

V. Deployment and Testing

After completing the steps of work and design, the features of JADE platform have been explored for the implementation of multi-agent system and using web server. With reference to the design phase, the following agents have been developed using Java language: Generic Agent, DB Agent, EEG Agent, ECG Agent and EMG Agent and web server agent. The required behaviors and actions were implemented as per the design guidelines and FIPA recommendations. The required table has been created in Database which is accessible through Java Data Base Connectivity (JDBC).

VI. Conclusion

In the Earlier days, either it may be data's or some prescription use to store only in the database alone and also in the system's local disk. And the main drawback in the previous system is each and every medical practitioner has different opinion and also approach in different ways where the multi user can't access the particular medical practitioner.In order to overcome all the problems here we're using the web server in the most popular platform called JADE and FIPA also is one of the most popular compliant platforms which is used for developing the multi agent system. Here the Bio-Signal is used as input, which has been proposed and also designed for developing the multi agent system, which is mainly developed under JADE platform and web server. As the agents on the JADE environment run on Threads, the response time is very less which helps the medical practitioner to make a quick diagnosis.

References

[1] Caire G. and Cabanillas D. (2004), 'JADE tutorial: creating and using application specific ontologies', see: *http://jade.tilab.com/doc/CLOntoSupport.pdf.*

[2] Campo C. (2002), 'Directory Facilitator and Service Discovery Agent', FIPA Document Repository, see: *http://www.fipa.org/docs/input/f-in-00070/f-in-00070.pdf.*

[3] Dennis A. and Wixom B.H. (2000), 'Systems Analysis and Design: An Applied Approach', *John Wiley and Sons.*

[4] FIPA Interaction Protocol Specifications, see: *http://www.fipa.org/repository/ips.php3*.

[5] Foundation for Intelligent Physical Agents (FIPA), see: *http://www.fipa.org/.*

[6] JADE-Java Agent DEvelopment Framework, see: *http://jade.tilab.com/.*

[7] Luck M., Ashri R. and D'Inverno M. (2004), 'Agent-Based Software Development', *Artech House Publishers, 2004.*

[8] Quarta F. (2003), How to use the XML support with JADE, see: *http://jade.tilab.com/doc/tutorials/XMLCodec.htm*.

[9] Shoham Y. (1993), 'Agent Oriented Programming', *Artificial Intelligence*, Vol. 60, No. 1, pp.51-92.

[10] Turci P. (2001), How to use the RDF support with JADE, see: *http://jade.tilab.com/doc/tutorials/RDFCodec.html.*