An Agent Oriented approach with Proposed Cultural Algorithm in Healthcare for Acquisition of Knowledge and Integration

Subiksha K.P
Research Scholar, Madurai Kamaraj University,
Madurai, Tamil Nadu, India

Abstract: In Neuro medical applications forward for standard terminologies such as NeuroSurLex tool is unambiguous and warmly needed. NeuroSurLex tool helps in developing and managing the standard terminology. Hence the standard conventions for Neurology being and widely used by the Neurotherapists, will help in serving the patients better. Patient reports with this standard terminology will effectively help the Neuro therapists to treat promptly without any delays. However the development and maintenance of these terminologies are very challenging tasks. Few specialized tools are available in markets will help developers to accomplish this task. Protégé, which is an open-source terminology-editing tool, helps to create Ontologies to represent the tacit knowledge which is being acquired from Neuro Therapists. It is hand-me-down as editing gadgetry, as it not counting provides a pontoon for developers; hence it is window-card to deliver appropriately the terminologies in end-buyer applications. The NeuroSurLex motion has overdues adopted Protégé for road its Neuro medical terminologies. Protégé provides span especially gainful for managing Neuro therapeutic terminologies: a gurry graphical user interface for navigating unsparing taxonomies, visualization Ways Management helps to view the complex term relationships and an easy programming interface so developers can create terminology-driven Neuro therapeutic applications. In confederate, Protégé has an adjustable plug-in, with a broad amid of components to implement the desired functionalities. In this paper, an Agent Oriented approach with Proposed Cultural Algorithm in Healthcare for Knowledge Integration and Acquisition has been mentioned. This Agent oriented methodology helps to attain the desired functionality and the report that highlights here emphasized the need for standard terminologies. So this paper shows how to attain the convention standards, value of managing terminology in healthcare especially in Neurology domain.

Keywords: NeuroSurLex, Healthcare, Terminology driven, Agent Oriented, Cultural Algorithm.

1. INTRODUCTION
Health info sharing between patients and physicians helps to enhance diagnosis, promotes self-care, and patients conjointly recognize additional info concerning their health. The use of Electronic Health records remains scarce currently however is increasing in North American country, and British primary care. Tending info in Electronic Health Record area unit, and it is necessary sources for clinical, research, and policy providers. Health info privacy (HIP) and security has been an enormous concern for patients and suppliers. Prescribing errors area unit the most important known supply of preventable errors in hospitals. A 2006 report by the Institute of medication calculable that a hospitalized patient is exposed to a drugs error on a daily basis of his or her keep. Computerized supplier order entry, (formerly referred to as doc order entry, will cut back total medication error rates by eightieth, and adverse (serious with hurt to patient) errors by fifty fifth. The shortage of standards in terminology conventions creates such huge issues. International Classification of Diseases focuses on the domains of functioning (disability) associated with health conditions, from both medical and social perspectives [9].

With the help of NeuroSurLex tool which is presented in this paper helps to fix the issues in an appropriate manner. NeuroSurLex tool is mainly used to deal with the medical specialty which mainly focuses with disorders of the nervous system.

Broad and consistent utilization of NeuroSurLex tool will provide:

- Enhanced health care quality or effectiveness;
- Enhanced health care productivity or efficiency;
- Stop medical errors and increase health care accuracy and procedural correctness;
- Cut back health care costs;
- Increase body efficiencies and tending work processes;
- Decrease work and unproductive or idle work time;
- Extend period of time communications of health info among health care professionals; and
- Expand access to reasonable care.

2. RELATED WORK/LITERATURE
SURVEY
Hospitals contain a diversity of processed info systems, and that they usually ask similar procedures, findings, and diagnoses refer to completely different terminologies. The shortage of standards in terminology conventions creates a barrier for comparison studies across enterprises and among health organizations, and it conjointly hampers analysis as a result of using tough to retrieve indexed case material in an exceedingly consistent manner. RadLex is a project to form a comprehensive
medical imaging terminology for capturing, indexing, and retrieving a spread of radiology info resources. RadLex was initially developed victimization word processors and spreadsheets. However, it shortly became evident that these tools were too restricted for accessing and managing RadLex. First, as RadLex is a giant taxonomy, it is tough to ascertain and navigate RadLex victimisation standard tools like word processors and spreadsheets. Second, RadLex contains several attributes associated with every term, and it is tough to take care of this info in correct structured type as RadLex grows. Third, though text documents and spreadsheets were a straightforward initial format for aggregation the RadLex terms, they are not appropriate for distributing RadLex on the net or for creating it accessible to applications. RadLex was an initio developed victimisation word processors and spreadsheets. However, it shortly became evident that these tools were too restricted for accessing and managing RadLex. Third, though text documents and spreadsheets were a straightforward initial format for aggregation the RadLex terms, they're not appropriate for distributing RadLex on the net or for creating it accessible to applications. A tool tailored to the wants of language development and dissemination in radiology is required. Protégé has been used because the primary development environment for several ontologies within the life sciences. These projects embrace the Foundational Model of Anatomy, Cerner’s Clinical Bioinformatics ontology, the DICE TS, and therefore the MGED ontology. Besides providing a set of practicality that helps curators edit and manage terminologies like RadLex, Protégé provides associate degree protractible design to that custom practicality specific to radiology wants are often additional, and it conjointly provides a platform for deploying radiology applications that exploit terminologies like RadLex.

3 EXISTING SYSTEM
Currently there is no system exists as similar to the NeuroSurLex tool. The main reason for developing this tool is due to the shortage of standards in terminology conventions in Neurology domain a subspecialty of Healthcare.

4 PROPOSED SYSTEM
In this we have explored the limited domain in Neurology namely Myopathy, Encephalopathy, Neoplasms, Spinal Cord Disorders, Traumatic Injuries.

4.1 NEED FOR AGENT ORIENTED APPROACH (AOSE)
Agents are heterogeneous; in this totally different agents could also be enforced using different programming languages, architectures, and techniques. AOSE helps in such a way that assumptions about the delivery platform need not be concerned.[15] Main reason for need for AOSE is because of :
 • Centered around specific agent-oriented abstractions
 • The adoption of OO methodologies would manufacture classes, objects, client-servers: very little to do with agents. Each methodology might introduce more abstractions
 • Not directly translating into concrete entities of the software package.

In this NeuroSurLex Tool, development has been made using Agent oriented approach, each agents performs different services made of different framework and techniques. The developed agents that are the hosted application can be plugged and run in any environment.

FIG -1: PLUG-IN MODEL

4.2 CLASSICAL APPROACH VS AGENT ORIENTED APPROACH
A software methodology is nothing but the set of pointers for covering the entire system development lifecycle each technically and managerially provides methods a comprehensive set of ideas and models.

FIG -2: CLASSICAL CASCADE PROCESS

It can be outlined as full set of techniques (rules, guidelines, heuristics), absolutely delineated set of deliverables, modelling language set of metrics, quality assurance, coding (and other) standards, pointers for project management. The phases of software development: independent of programming paradigm; Methodologies are usually organized around this classical process; Inputs, outputs, internal activities of “phases”.

FIG -3: AGENT BASED PROCESS
Instances of two classes will collaborate with one another by exchanging messages, and instance of a class contains an instance of another class i.e. Container class. Besides this basic relation, there are several alternative abstract similarities between object-orientation and agent-orientation which will be mapped onto one another as summarized within the following table.

<table>
<thead>
<tr>
<th>Table - 1: OOP Vs AOP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Structural Elements</strong></td>
</tr>
<tr>
<td>abstract class</td>
</tr>
<tr>
<td>class</td>
</tr>
<tr>
<td>member variable</td>
</tr>
<tr>
<td>method</td>
</tr>
</tbody>
</table>

**Relations**

<table>
<thead>
<tr>
<th></th>
<th><strong>OOP</strong></th>
<th><strong>AOP</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>collaboration (uses)</td>
<td>negotiation</td>
<td></td>
</tr>
<tr>
<td>composition (has)</td>
<td>holonic agents</td>
<td></td>
</tr>
<tr>
<td>inheritance (is)</td>
<td>role multiplicity</td>
<td></td>
</tr>
<tr>
<td>instantiation</td>
<td>domain specific role + individual knowledge</td>
<td></td>
</tr>
<tr>
<td>polymorphism</td>
<td>service matchmaking</td>
<td></td>
</tr>
</tbody>
</table>

These ideas correspond to the agent-oriented world by replacement class with role, state variable with belief/knowledge and technique with message. So the job definition describes the agent’s capabilities, the information that's required to produce class results and also the requests that trigger a selected service.

4.3 GAIA METHODOLOGY

Requirements capture phase are considered independent of the paradigm used for analysis and design. For this reason Gaia does not deal with the requirements capture phase [12, 13, 16]. The analysis phase consists of the following models:

- Role definition (permissions, responsibilities and protocols)
- Interaction model (used for protocol description)

**Algorithm- 1 for Cultural learning of an artifact capability**

Create social network
Initialize global belief space
Add agents to social network
Each agent gets artifact
Learning element selects ability
Learning element formulates a goal
Performance Element initializes local belief
Performance Element initializes goal and capability
While goal not achieved
Performance Element provides attribute value sequence
Critic tests attribute value sequence
Critic generates feedback
Learning element generates changes
Performance Element applies changes
End
evaluation process of the population space. After the successful sequence is accepted Global Belief space adjusts its situational and normative knowledge using the same rules as when it receives top performers. This allows the agents to benefit from the success of others. The contributions are transmitted through an acceptance function and the knowledge in the belief space is adjusted accordingly. That knowledge influences the evolution of the individuals in the population space via an influence function. CA supports the use of any kind of evolutionary algorithm in the implementation of the population space. After the successful sequence is accepted Global Belief space adjusts its situational and normative knowledge using the same rules as when it receives top performers. This allows the agents to benefit from the success of others.

### 4.5 TECHNICAL SPECIFICATION AND IMPLEMENTATION

The designer of the MAS can simply implement agents whose internal structures and ways square measure predefined [17]. The info structures and also the AI tools that square measure to be used ought to be outlined during this stage. The Gaia methodology is a trial to outline an entire and general methodology that its specifically tailored to the analysis and style of MASs. Gaia could be a general methodology that supports each the degree of the individual agent structure and also the agent society within the MAS development method. MASs, consistent with Gaia, square measure viewed as being composed of variety of autonomous interactive agents that board associate organized society during which every agent plays one or additional specific roles. Gaia defines the structure of MAS in terms of a task model. The model identifies the roles that agents ought to play inside the MAS and also the interaction protocols between the various roles. The target of the Gaia analysis method is that the identification of the roles and also the modeling of interactions between the roles found. Roles comprise four attributes: responsibilities, permissions, activities and protocols. Responsibilities square measure the key attribute associated with a task since they verify the practicality.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>Permissions</th>
<th>Responsibilities</th>
<th>Liveness</th>
<th>Safety</th>
</tr>
</thead>
<tbody>
<tr>
<td>name of role</td>
<td>short English description of the role</td>
<td>“rights” associated with the role</td>
<td>“tasks” associated with the role</td>
<td>liveness responsibilities</td>
<td>safety responsibilities</td>
</tr>
</tbody>
</table>

Responsibilities are of two types: liveness properties the role must prevent and disallow that something bad happens to the system. Liveness describes the tasks that an agent must fulfill given certain environmental conditions and safety ensures that an acceptable state of affairs is maintained during the execution cycle.

In order to realize responsibilities, a role has a set of permissions. The activities are tasks that an agent performs without interacting with other agents. Finally, protocols are the specific patterns of interaction, e.g. a seller role can support different auction protocols. Gaia has formal operators and templates for representing roles and their attributes and also it has schemas that can be used for the representation of interactions between the various roles in a system.

In the Gaia design process the first step is to map roles into agent types and to create the right number of agent instances of each type. An agent type can be an aggregation of one or more agent roles. The second step is to determine the services model needed to fulfill a role in one or several agents. A service can be viewed as a function of the agent and can be derived from the list of protocols, activities, responsibilities and the liveness properties of a role.

Finally, the last step is to create the acquaintance model for the representation of communication between the different agents. The acquaintance model does not define the actual messages that are exchanged between the agents it is rather a simple graph that represents the communication pathways between the different agent types.

### 4.6 SOFTWARE DEVELOPMENT FRAMEWORK

JADE is a software development framework fully implemented in JAVA language aiming at the development of multi-agent systems and applications that comply with FIPA (Foundation for Intelligent Physical Agent) standards for agents [24]. JADE provides standard agent technologies and offers to the developer a number of features in order to simplify the development process:

- Efficient transport of ACL (Agent Communication Language) messages between agents.

All agent communication is performed through message passing and the FIPA ACL is the language that is used to represent the messages.
TABLE 3: ACL MESSAGE DEFINITIONS

<table>
<thead>
<tr>
<th>ACL Message</th>
<th>RequestNeuroTerminology</th>
<th>RespondNeuroTerminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender</td>
<td>Physician</td>
<td>Physician</td>
</tr>
<tr>
<td>Initiator Agent</td>
<td>Guide Agent</td>
<td>Guide Agent</td>
</tr>
<tr>
<td>Receiver</td>
<td>Physician</td>
<td>Physician</td>
</tr>
<tr>
<td>Guide Agent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIPA performative</td>
<td>REQUEST</td>
<td>FIPA performative :</td>
</tr>
<tr>
<td>Protocol</td>
<td>Request Terminologies</td>
<td>Protocol : Request</td>
</tr>
<tr>
<td>Terminologies</td>
<td></td>
<td>Terminologies</td>
</tr>
<tr>
<td>Language</td>
<td>SL</td>
<td>SL</td>
</tr>
<tr>
<td>Ontology</td>
<td>Image</td>
<td>Image</td>
</tr>
<tr>
<td>Content : Ontology action</td>
<td>Request Terminology</td>
<td>Content : Ontology</td>
</tr>
<tr>
<td>Concept : Terminology</td>
<td>Request Terminology</td>
<td>Concept : Terminology</td>
</tr>
</tbody>
</table>

- FIPA-Compliant agent platform, which includes the Agent Management System the Directory Facilitator and the Agent Communication Channel.
- Distributed agent platform. The agent platform can be distributed on several hosts, each one of them executes one Java Virtual Machine.

Each agent is equipped with an incoming message box and message polling can be blocking or non-blocking with an optional timeout. Moreover, JADE provides methods for message filtering. The developer can apply advanced filters on the various fields of the incoming message such as sender, performative or ontology. FIPA specifies a set of standard interaction protocols such as FIPA-request, FIPA-query, etc. that can be used as standard templates to build agent conversations. For every conversation among agents, JADE distinguishes the role of the agent that starts the conversation (initiator) and the role of the agent that engages in a conversation started by another agent (responder). According to the structure of these protocols, the initiator sends a message and the responder can subsequently reply by sending a not understood or a refuse message indicating the inability to achieve the rational effect of the communicative act, or an agree message indicating the agreement to perform the communicative act. When the responder performs the action he must send an inform message. A failure message indicates that the action was not successful. JADE provides ready-made behaviour classes for both roles, following most of the FIPA specified interaction protocols. Because the FIPA interaction protocols share the same structure, JADE provides the AchieveREInitiator/Responder classes, a single homogeneous implementation of interaction protocols such as those mentioned above. Both classes provide methods for handling all possible protocol states.

The below figure depicts the prototype of the NeuroSurLex Tool.

In JADE, agent tasks or agent intentions are implemented through the use of behaviors. Behaviours are logical execution threads that can be composed in various ways to achieve complex execution patterns and can be initialized, suspended and spawned at any given time. The agent core keeps a task list that contains the active Behaviours. JADE uses one thread per agent instead of one thread per behaviour to limit the number of threads running in the agent platform. A scheduler, hidden to the developer, carries out a round robin policy among all Behaviours available in the queue. The behaviour can release the execution control with the use of blocking mechanisms, or it can permanently remove itself from the queue in run time. Each behaviour performs its designated operation by executing the core method action(). Behaviour is the root class of the behaviour hierarchy that defines several core methods and sets the basis for behaviour scheduling as it allows state transitions (starting, blocking and restarting). The children of this base class are SimpleBehaviour and CompositeBehaviour. The classes that descend from SimpleBehaviour represent atomic simple tasks that can be executed a number of times specified by the developer. Classes descending from CompositeBehaviour support the handling of multiple Behaviours according to a policy. The actual agent tasks that are executed through this behaviour are not defined in the behaviour itself, but inside its children Behaviours [14-16].

The FSMBehaviour class, which executes its children Behaviours according to a Finite State Machine (FSM) of Behaviours, belongs in this branch of hierarchy. Each child represents the activity to be performed within a state of the FSM, with the transitions between the states defined by the developers.

The other software’s which are needed for this NeuroSurLex tool development are Unified Medical Language and the UMLS provides data for system developers as well as search and report functions for less technical users,
Protégé, Matrix Laboratory, JAVA, XML, and Ontology Web Language

4.7 PROTÉGÉ APPLICATION AND GRAPHICAL USER INTERFACE

Protégé can be run as a complete separate application or through a Protégé client in communication with a remote server (the latter is especially helpful for cooperative ontology development). Once the Protégé desktop application is launched, the user will produce a replacement a brand new or open an existing ontology, or import ontology from a different type of formats. Protégé creates a file that records display-related info and also the location of the ontology source file; the project file simplifies the method of reopening ontologies, and it maintains user user interface customizations.

FIG – 5: PROTÉGÉ ARCHITECTURE

Protégé creates a file that records display-related info and also the location of the ontology source file; the project file simplifies the method of reopening ontologies, and it maintains user user interface customizations. Once ontology is opened, the user works with it in the Protégé user interface. The Protégé user interface is organized into separate panels (accessed by tabs on the top of the GUI) that offer completely different views into the contents of the ontology. The primary tab is that the categories tab, that is that the commonest read accustomed browse ontologies and terminologies. In that screen in one of the tabs ontology is shown as an expandable tree on the left of the user interface, and also the attributes of category elite by the user is shown on the correct. within the tree read, terms that ar indented and below a term ar known as “child” terms, whereas the term on top of is that the “parent”.

5 SECURITY IMPLEMENTATION

1. Security plan has been considered and it is in place, security will inevitably be compromised [19].
2. Have analyzed and used tested architectures, technologies and cryptography systems, have avoided the vendors who refused to disclose the inner workings of their security product.
3. Security should not be separate product. It has been handled along during the development of every module.
4. Better to stay ahead of threats by providing enough security with appropriate flexibility of function.
5. Have used secure hash algorithm rather Message digest algorithm [20] which is nothing but a cryptographic hash function.

6 KNOWLEDGE ACQUISITION AND INTEGRATION USING ONTOLOGY

Ontology is especially used to build domain assumptions express and easier to edit domain assumptions for e.g. genetics knowledge base is simpler to know and update bequest information [1]. For Agents communication ACL alone is not enough Ontology should also be common. Ontology is a part of the agent's knowledge base that describes what kind of things an agent can deal with and how they are related to each other [7].

FIG - 6: ONTOLOGY DEVELOPMENT PROCESS

And additionally to separate domain information from the operational knowledge re-use domain and operational knowledge singly (e.g., configuration supported constraints).

Ontologies are remodeled to become instance information with regards to the vocabulary of the meta-ontology. Ontology reflects the structure of the globe it's typically regarding structure of concepts; well actual physical illustration is not a problem. In object oriented structure: it reflects the structure of the information and code and it is typically termed as methods and it describes the physical representation of knowledge.

The main use of ontology is to share and communicate knowledge, each between individuals and between pc systems. variety of generic ontologies are made, every having application across variety of domains that permits the re-use of knowledge. During this means, NeuroSurLex project need not begin with a blank sheet of paper, however with variety of skeletal frameworks that may act as predefined structures for the knowledge being noninheritable. Ontologies conjointly offer guidance to the knowledge engineer within the forms of knowledge to be investigated.
In the development of NeuroSurLex tool the mix of this theory-driven, top-down approach with a data-driven, bottom-up approach with the Ontology engineering appropriate to the process has been followed. This process of knowledge acquisition will help to build an efficient system.

7. PERFORMANCE MEASURES

7.1 EVALUATION AND RESULTS

The distributed problems solving environment fit well with the multiagent system (MAS) since the solution requires autonomous behavior, collaboration and reasoning. The agents perform the underlying data analysis tasks very efficiently in distributed manner. The MAS offer architecture for distributed problem solving. The MAS deal with complex applications that require distributed problem solving. The MAS is also a distributed system, combining cultural algorithms with MAS for data analyzing will further enhance the processing power of the application. The multiagent system is used for the extraction of results from these nodes [24]. These agents can roam from one node to other node freely and can be stored at any node in the distributed network. The results of these agents can also be stored at anywhere in the network. The architecture of mobile intelligent agents is shown in figure 4. This is a multiagent system, capable of performing classification, interpretation of large datasets [21].

The above figure depicts that standard terminologies in Neurology domain, a subspecialty of Healthcare helps to represent the knowledge precisely.

8 CONCLUSION

Cultural algorithm implemented using Agent Oriented methodologies manipulate the acquired knowledge from his/her experiences which is nothing but tacit knowledge which has been gained from their work experiences and it helps to influence the direct evolution of population. It is quite cost effective and proved with good performance results when compared with existing algorithms. In future this algorithm can be applied to the real world problems of any domain in Healthcare. Accuracy of the algorithm can be improved by modifying dissimilar parts of belief space and acceptance function. Diversity can be enhanced by using various subpopulations with dissimilar situational or normative knowledge instead of single population.

REFERENCES


AUTHOR

Subiksha K.P PhD scholar, Madurai Kamaraj University (MKU). I have exposure towards Avionics Domain, Health care Domain and Finance Domain. I have been in IT for 6+ yrs. I have completed my master’s Degree in computers from Madurai Kamaraj University. Am a rank holder in Post-Graduation as well as in Under Graduation. Worked as a Design Engineer for GE Healthcare. Have worked as a Software Engineer with Larsen&Toubro Infotech Ltd, Soft brands Research & Development Private Ltd., I have worked in Operating Systems like Windows 95, 98, Windows NT/Win 2K, Windows 7, Windows XP, Linux and Programming Languages like C, C++, VC++, PL/SQL, JAVA, VB, Pascal. My core technologies are VC++, DTS, ATL, COM.My research interests are Artificial Intelligence, Data Mining, and Software Engineering.