IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727 Volume 5, Issue 5 (Sep-Oct. 2012), PP 22-27
www.iosrjournals.org

Pretext Knowledge Grids on Unstructured Data for Facilitating Online Education

Rahul Jassal[1], Chaman Singh[2]
[1](Department of Computer Science and Application, Punjab University Regional Center Hoshiarpur India)
[2](Department of Computer Application Govt. P.G. College Chamba, H.P. University Shimla India 176310)

Abstract : *The paper describes a plan, synchronized steps to cluster the unstructured data obtained from different sources i.e. internet, web crawler, mobiles in a form that it results in phrased information satisfying someone needs. Reach of such unstructured data in each one door faces constraints like limited financial resources, insufficient number of qualified instructors, and lack of diagnostic test for imparting individual training. The paper describes the wrapped way out for the data available on nets, mobiles or some another source through agents, algorithms and some designed function kept in repositories. The proposed system contains authoring agents, algorithm and repository to lead out a traditional teaching approach.*

Keywords: *Repositories, knowledge grid, Authoring Algorithms, Agent Architecture, Activation Dongle*

I. INTRODUCTION

The internet is the largest, powerful computer network or data provider in the world. As more and more colleges, universities, schools, companies and private citizens connect to the Internet. It is being difficult to determine what difference lies between local and distance education. The unstructured, semi structured and structured data on servers carries eighty percent of internet space. On one side, the explosion in data has accelerated like you tube started in 2005 and today users upload 35 hours of video every minute. Falling costs of data storage and transmission cheap web access over the phone lines has brought millions more to the internet and perhaps the most significantly, the emergence of mobile technologies. Companies are ready to invest in good visualization techniques. Focus should be on increasing the most recent information used and put on priority so as to not have historical biases. Huge amount of information is accumulated in variety of physically, logically, distributed, autonomous and heterogeneous information repositories [3] [4] [5]. Agents act as a coordinator and receive user's requests. After the analysis of the request it dispatches wrappers to local information repositories. Each wrapper issues local commands to the local information repository and receive intermediate results. The wrapper translates it into a predefined data representation form and sends it back to the agent; finally the agent collects data returned by the wrapper and produces final result. Agent resembles chat but they do so for the sake of accomplishment of a task. Normally when two nodes are in a state of a chat they usually start with authentication, connection establishment and will be in a state of sharing of files, hyperlinks, symbols, text and wrapped data. Being proactive they can take the initiative to perform a given task even without an explicit stimulus from a user and to design such agent either it is convenient to build multi agent systems on top of an agent oriented middleware that provides the domain independent infrastructure, allowing the developers to focus on the production of the key business logic and the other way out is start from scratch. Data may come on mobile communication on dual stack [10] or in other case. While design such an agent architecture leave a segment for the information the agent is carrying about its environment which may be complete or incorrect. Next to it is an segment keeping the task allocated to the agent so corresponds to the objective or goals and a plan specifying some courses of action that may be followed by. While converse the agent either opts for master/slave or client/server architecture for task and resource allocation among slave agents by a master agent. The master controller can gather information from the agents in the group, create plans and assign tasks but it seems impractical as it is difficult to create such a controller. So way out is to create an agent which is independent from such parameters. In this paper the agent represent itself as shown in Figure 1.

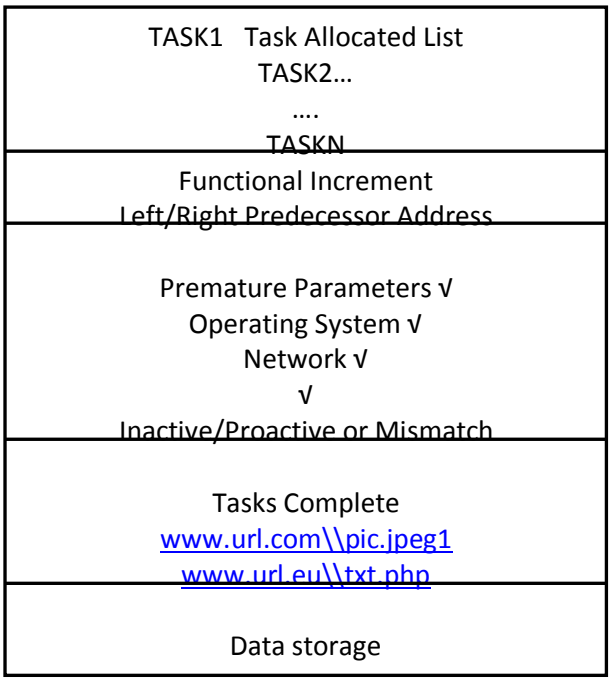

Fig 1 :- Agent Architecture)

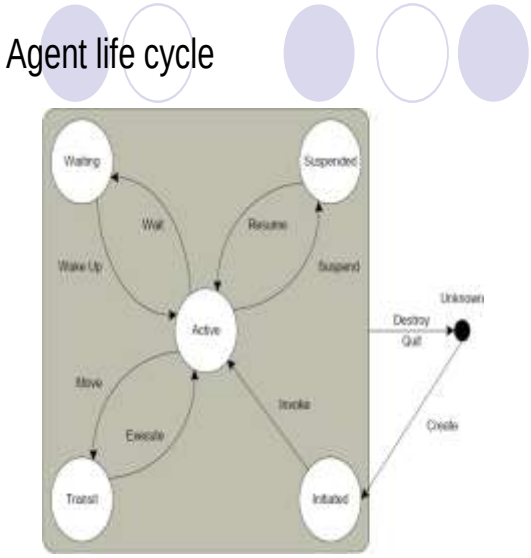

Fig-: 2

Agent Life Cycles

II. AGENT ARCHITECTURE

Agent Architecture first segment contains list of all those tasks need for completion. Each task in them represents a workflow like obtaining key information from Google so Task T_1 is using KQML or XML code constructs to fetch the information from meta tag in normal browsers and to interact with browsers and the operating system the agent must satisfy the list of parameters helping him in interaction with the corresponding environment. To accomplish, the FIPA communication stack can be separated into several sub-layers within the application layer of the classical OSI or TCP/IP stack [1]. The wrapped meta data in OSI or TCP/IP helps the agent in communication with network within its life cycle, the inactive ports determines whether the required parameters matches with the environment to initiate for communication. The same is the case with mobile communication wrapping the required protocols in WAP [2]. Functional Increments the pointer to the next task list until no task should left in the segment.

AGENT ARCHITECTURE{A[n],U, θ,temp, parm,flag}
{

A[n] is an array list containing the task list, u is the Union operator to gather the data in result section, temp is the container used for keeping the task going beyond the value θ, and total number of tasks $M \leq \theta$. Parm holds the checklist for communication. By default flag is assigned value false after scrutinizing the parameters its vlue set to true.

For each (task value in A[T_1, T_2, T_3, T_4, T_5, Tm]
{
A[T_1]=google.srch.url1//containing a search for text
A[T_2]=google.srch.url2//containing a ppt.
A[T_3]=google.srch.url3//some jpeg format
A=AU {i}//U is for Union and put into Result Section.
If (Tm>θ)
{
//parameters
Parm={sender ,receiver ,reply-to ,content ,language…}
Flag=true
Perform A [T_1]
}
Else
{

temp[0]=google.srch1.//meta tag1.
temp[1]=google.srch2//meta tag2
}
End Loop
}
exit AGENT INTERFACE.
}
A [T_1]
{
Request
:sender(agent-identifier:name rahul@mydomain.com)
:receiver(agent-identifier:name chaman@yourdomain.com)
:ontology travel-assistant
:language FIPA-SL
:protocol fipa-request
:content
“”((action
{agent-identifier: name chaman@yourdomain.com) (book session:time 12/30/1899)
(school:public school)(village:chabewal))”” .
}

The following figure depicts the the agent interface meant for establishing multi connection between various schools of the village whose status is online, the activate key sends a signal to each single machine’s ***“Activation Dongle”*** in schools for filling simple E- form meant for updating details like Village name, school name and the strength there.

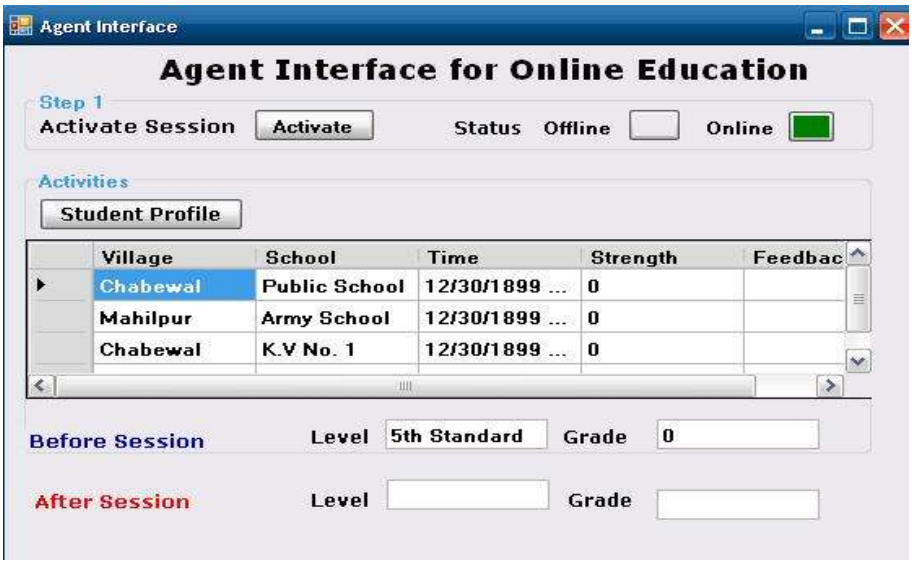

Fig:- 3 Agent Interface for Online Education

In response to this, the activation dongle which is again an agent on other side replies the predecessor as
(query-ref
:sender (agent-identifier:name rahul)
:receiver(set (agent-identifier:name chaman))
:content
“((iota?x(P?x))”//proposition P asking for chabewal school in query x
:language fipa-sl
:reply-with query1)
(inform
:sender (agent-identifier:name chaman)
:receiver(set (agent-identifier:name rahul)
:content
“((=(iota?x(p?x))Public school))”
:language fipa-sl
:in-reply-to query1)

The agent at school side aired the replies on queues as shown in Figure. Further the filled cells of the Queue can streamed to "***VisitingNodes***" .

Port Open(9:00-10:00; 11:00-1:00;3:00-5:00)

Clear Operation: - To clear the Queue

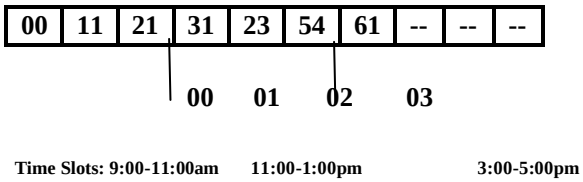

Fig:- 4 Visiting Nodes Data

K.V	-	-	X	-	-	$SCL1_{10}$
ARMY	-	-	-	X	-	$SCL2_{20}$
P.S	-	-	X	-	-	$SCL3_{30}$
D.P.S	X	-	-	X	-	----------
S.S	-	-	-	-	X	SCLN

Fig:-5 Fetched Knowledge Grid

The Figure 4 displays value like {00, 11, 21, 31, 23, 54, and 61} which is fetched from Knowledge grid as shown in Figure 5. So Agent Interface is executing the tasks like data gathering on grids and streamed on ***VisitingNode***.For streaming out the data from knowledge grids to Queue [Figure 4].Following algorithm can be used

Figure 4[VisitingNode][dataset,temp,size,queue]

Suppose the size of the grid is nxm, so the dataset contains n number of rows and m number of columns, temp is the temporary location used to store the last index of dataset.

//Fill the grid by NULL values

1. for each(data value in dataset(00,01,02…))
2. dataset[i][j]=NULL
3. Temp=0,k=nxm, size=0
4. Queue[size]=k
5. Repeat the loop as in step 1 or 2 for
6. Temp=dataset[i][j]
7. If(temp!=NULL)
8. Queue[k]=dataset[i][j]
9. Increment k
10. End of Loop
11. Increment j and i

Well, what happens years back, programmers kept the things in "Code behind" format and the hyperlink processes the relevant keywords for information. The labeled links agile the way in getting the desired outputs. But to exempt out the knowledge without keeping the clicks is just useless. To add learner's ability clicks into account or previous knowledge into account the project is divided into two sections which remains part of Agent Architecture.

The working of Agents is applied on unstructured data and help in assembling the unmanaged data for facilitating online education. The Figure1 is divided into two sections of communities where one reflects a picture of urban area where accessibility of WAN, LAN is frequent and the other section is rural lacks this access. The control centre of figure or scheduler is using "***Agent Interface"*** for establishing a connection with village's school using Activate key

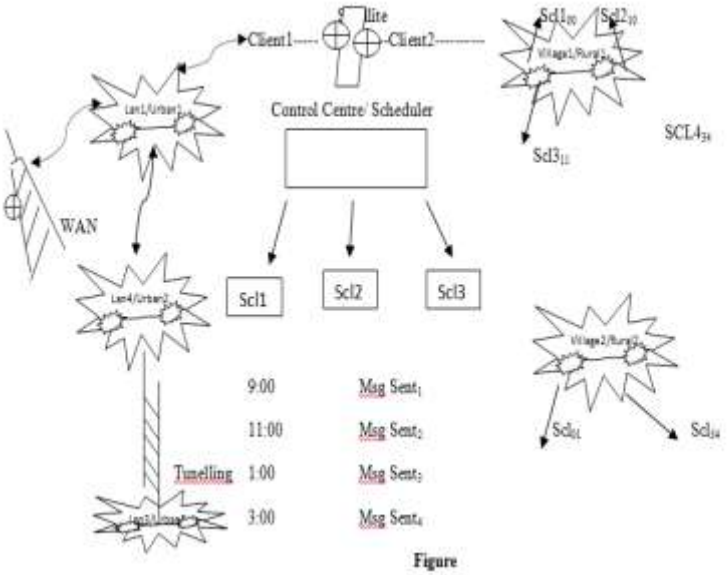

Fig:- 6 Agent Interface

Working

The above Figure 6 shows an various communication patterns depicting different LAN networks of urban and rural regions. The upper layer of the figure carrying zones where communication can perform from one LAN to another either though WAN or through tunneling. Right side of the Figure showing an village1 of rural area1 with school SCL1, SCL2 so-on. The control centre scheduler is the section performing chat with SCL1, SCL2, SCL3 through agents while ports are scheduled to open at time interval {9,11,1,3}.

III. TYPES OF AGENT ARCHITECTURE

There are two types of agent architecture given below

III.1 Assembly/ Scheduling

This is the stage where operation like filling up of knowledge grids or executing out "VisitingNode" algorithm takes place. Figure 5 depicts the Interface at School side, as many students can be added and send their respective query to Agent, Agent has a query section where these queries are kept for a while and after determining their respective solution they are sent back to students for their approval, students authentication further compels to store that data in repository for different students in grading their level to higher ones.

III.2 Collection/Execution Process

This section includes the "Result section of Agent Architecture" which is information carried from different sources. The whole procedure whether it is lying in Assembling or scheduling section is repeated for intervals t1, t2, t3 and so-on. The collection helps in creating a knowledge base for a particular student and helps them for improving their grades. The session falls in two categories like theoretical and practical. Agent has to work upon prefix or syntax like 1,2,3 or a, one, two, three and so-on. If a query is carrying numeric phrases as defined above then their outcome or processing is interlinked with predefined libraries otherwise if the topic remains a search for definition or graphics then "***Integrator***" remains one of the options for maintaining unstructured data for future use.

III.3.1 Integrator

But for executing out Query containing Numerical syntax like i.e.

"Addition of 2 numbers"
"2 numbers of Addition"
"Addition of two numbers"
"Numbers of 2 addition"

Query can be of any of the four phrases. For finding out the solution for this particular problem Agent divide any of the statements into subject, object, conjunction and the numerical values to parameters. Further this syntax compared with the repository and after finding a same match the particular parameters is replaced with new values and action can be carried out

IV. REPOSITORY

Q_type{x1,x2,x3}	Noofvalues{x2}	Function
[Number Addition]	2, two	{x,y}
{theory details}	3, three	{x,y,z}
	4, four	{x,y,z,k}
	N	{x,y,…….n}

Inputs can be anything this time student enters and next time may be it changes like Addition Numbers: - wrong interpretation. Note: - Results are successful, if the query words matches with all the column words and fails if query is missing some words means numbers of words in column is greater than query words

IV.1 Weights of Subject Modules

Generally given topic or course may be divided into set of modules. Let S be the course which is divided into m modules, S={M1,M2,M3.....Mm}. These modules are arranged in the order of degree of difficulty. DD(M1)<DD(M2)…DD(Mm), where DD is the degree of difficulty of the module. The degree of difficulty of a module can be given as weight, Wi=X*I, where I starting from 1 to m-1, where value of x depends on the level of the difficulty of the module that may vary from 10 for high school level, 20 for graduate level and soon. The interlinked weight in creases as the modules get more advanced. All the subjects' modules have been placed in ascending order of conceptual view or proposed Model. For example assume that subject S has 4 modules {m1,m2,m3,m4} and the subject domain model is built at the post graduate level then the weights of the modules m1, m2, m3, m4 are w1=50, w2=100, w3=150, w4=200 respectively.

V. CONCLUSION

A prototype design of subject model for web based subject material has been presented. One can design the structure allowing the diverse users to offer relevant navigation possibilities. Paper describes a plan, synchronized steps to cluster the unstructured data obtained from different sources i.e. internet, web crawler, mobiles in a form that it results in phrased information satisfying someone needs. The paper describes the wrapped way out for the data available on nets, mobiles or some another source through agents, algorithms and some designed function kept in repositories. The proposed system contains authoring agents, algorithm and repository to lead out a traditional teaching approach.

Acknowledgements

The author would also like to express their sincere thanks to Mr. Chaman Singh for his worthy Help and support for this work.

REFERENCES

[1]. Fabio Bellifemine, Giovanni Caire and Dominic P.A. greenwood "Designing Multi Agents through JADE".
[2]. AP , article based on Dictionary of computing prior to 1 november 2008.
[3]. A.K. Elmagarmid and C..Pu, (eds.). Special issue on Hetergenous Databases. ACM Computing Survey, *vol. 22*, no.3, 1990.
[4]. A.R. Hurson, M.W.Bright and S. Pakzad, (eds.). Multidatbase systems, An advanced solution for global information sharing. IEEE Computer Society Press, 1994.
[5]. E. Pitoura, O. Bukhres and A. Elmagarmid. Object orientation in Multidatabase systems. ACM computing survey, *vol 27*, no,2, pp. 141-195 June 1995.
[6]. O.P. Rishi, Rekha Govil and Madhavi Sinha, " Distrubuted Case Based Reasoning for Intelligent Tutoring System".
[7]. Ioannis T. Christou, Sofoklis Efremidis and Thanasis tiropanis, "Grid Based Virtual Laboratory Experiments for a Graduate Course on Sensor Networks".
[8]. Somchart Fugkeaw, piyawit Manpanpanich and Sekhon Juntapremjit, "Multi-Application based on multi Agent System
[9]. Anandi Giridharan and P Venkataram ,"A Design of Subject Model for Web-based Education System" IIS IEEE 2002.
[10]. K.L.Bansal,Chaman Singh , "Dual Stack Implementation of Mobile IPv6 Software Architecture " FCS IJCA *Volume 25– No.9, July 2011*

Authors Profile

Rahul Jassal is working as Assistant Professor in Department of Computer Science & Application, Panjab University Regional Centre, Hoshiarpur, India. He received Master of Computer Application in year 2007 and clear the UGC-NET examination for subject "Computer Science & Application in the same year. He is with the post from last 5 years.

Chaman Singh (B.Sc., MCA, NET, Ph.D.) Has received the Master of Computer Application Degree in 2007 and completed Ph.D. in Computer Science 2012 from Department of Computer Science H.P.University Shimla, India. He also qualified UGC NET 2006. Have more than 4 years of Working Experience in Teaching, Guidance for PGDCA, MCA, BCA students, Software Development (Programming) and Networks. Published number of National and International Papers in like IJCA, IJCSI, IJCSCN, IJCSE etc.