Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Cognitive Agents for Understanding the Complexities Involved in Web-based Knowledge-gathering Tasks

Balaji Rajendran, Iyakutti K
Madurai Kamaraj University
Madurai, India
balaji@cdacbangalore.in, iyakutti@gmail.com

Abstract—**A Web-based knowledge-gathering task is a common, but complex activity carried out by several users on the Web. The complexity is of two types: One is the inherent complexity involved in the task, which is essentially the information need of the user, and the other is the perceived complexity by the user – that varies according to the proficiency of the user, in the particular subject matter of the task. We present an Agent-based Cognitive model for understanding both the inherent and perceived complexities. We conducted simulation of our model using logs of user sessions to arrive at the typical complexity values and to qualitatively represent them. We also carried out live experimentation through a prototype system that made estimates of the complexities as users were involved in Kwowledge gathering tasks, and the results vindicated our estimations.**

***Keywords*—Cognitive Model, Agent-Based Cognitive Model, Web-based Knowledge gathering Task, Cognitive Agents, BDI Agents, User Proficiency Estimation, Perceived Complexity, Inherent Complexity**

I. Introduction

Web-based Knowledge gathering (WKG) tasks are one of the frequent and common tasks carried out by users through the web. A user carrying out such a task, generally need to discover appropriate resources, understand and learn from those resources, to accomplish a particular task. The task may typically involve collecting and correlating information from various heterogeneous sources, and performing explorative learning, typically within time limits [1].

The knowledge sought by the users of Web-based knowledge-gathering tasks would be of analytical or embrained [2] in nature rather than encyclopedic [3] or operational knowledge. Such users typically discover the resources required for the task by giving queries related to the task in a search engine. The type of queries used would predominantly be of information queries [4]. Web search, in this entire process of knowledge-gathering is an important tool, available at the disposal of the user. However it is left to the user, to effectively use it to address his needs without any support system.

Hence, a WKG task is a complex activity involving the requirement of sufficient related knowledge about the task and demanding higher levels of cognitive functions, and at the same time imposing high cognitive loads on the user, because of the structure of the information retrieval processes on the Web. A user may not always be able to express all his information needs by using appropriate keywords. In such cases, cognitive models can definitely be helpful in understanding the needs of the user, and guiding them to the appropriate resources. The motivation behind our model is to provide active assistance to the users, intended towards suggesting appropriate entities that can include keywords, resources, tools (such as discussion forums, or other interactive tools), and contacts of users who might have executed similar tasks and who wish to provide assistance with respect to a given task, during the execution of a task. The user then has to only sequentially explore them, and has to acquire knowledge. The acquisition process is a learning process performed the user.

A. Cognitive Models and Agents

Cognitive models are built with the intention to mimic the performance of human cognitive functions, in their ways of representing and processing information. The models will help us in understanding the knowledge-representations, strategies and skills adopted by humans, as well as the limitations experienced with respect to a given cognitive task. Cognitive models are typically built by studying and decomposing the human behavior, with respect to a given task. A basic tenet of any cognitive model is that the functions of that model need to be dynamic and evolving continuously, by learning, in order to emulate the human cognitive functions.

Agent-based computing paradigms are better suited to implement the continuously evolving cognitive models, through autonomous communication and learning mechanisms. The agent-based implementations of cognitive models can later help in composing larger models emulating various functions and capabilities, such as in RETSINA [5] - a multi-agent architecture that can support heterogeneous agents. A tutorial on Agent-Based Modeling and simulation can be found in [6], and a good introduction with a step-by-step approach can be found in [7]

In this paper, we present an Agent-based Cognitive Model, to arrive at an estimate of the complexity of the WKG task undertaken by the user, as well as to understand the perceived complexity for the given task by a user, so that active and intelligent assistance can be provided. We evaluated our model, by conducting simulations over a dataset that contained the activity logs collected from the users involved in different knowledge-gathering tasks through a Web-based Information system. The simulation helped us to understand the various complexity levels and helped us to fine-tune our model and estimates about complexity. We also conducted live experiments where users involved in Web-based Knowledge-

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

gathering tasks found the assessment of complexity by the agents to be correct to a good extent. The next section discusses some of the methods to identify user information needs, techniques to improve web browsing and surfing, and the use of agents for assisting users. Section III discusses our model of Cognitive Agent designed to understand the complexities of the WKG task as well as it is perceived by the users and gives the algorithms. Section IV discusses the implementation and validation of the Cognitive Agent model through simulation and experiments.

II. RELATED WORK

Agents have been used to model and implement several cognitive functions in order to address a particular need of a system or environment. A survey of the Computational Agents in the development of Cognitive Systems can be found in [8]. We focus on some of the related work, with respect to Agent-based user assistance in Web information systems.

A number of systems and techniques have been designed and attempted to provide personalized assistance to users, primarily for browsing and searching the web. Assisting users through Agents during browsing has been implemented in systems such as Letizia [9] that proactively fetches links from the page currently being viewed by the user, and recommends those links that may be of interest to a particular user, by analyzing their activities of browsing. Another example of an agent assisting the user is the Web Watcher [10] that searches the web autonomously on the behalf of the user, and provides interactive assistance to the user, using machine learning techniques. More examples, of agent-based assistance for browsing and searching can be found in [11]. These techniques are modeled from the Information Retrieval perspective, and have not generally considered the information needs of the user. Agents for user assistance also applied collaborative filtering techniques to recommend Web pages such as in [12], [13], that were chosen by other users whose overall tastes in those Web pages matched the user in question, though explicit rating of pages. More recently, the need for assistance to users in Web information retrieval is being explored as a separate topic as Web Information Retrieval Support Systems [14].

Cognitive approaches for agent-based assistance have also been addressed in some of the works. An ostensive model of progressively identifying the information needs is presented and discussed in [15]. A fuzzy technique based cognitive approach for agent-based personalized recommendations can be found in [16].

However most of the approaches in general have not focused on estimating the task complexities and the knowledge-level of the user, trying to accomplish a given task. Our approach is to identify and establish parameters for estimating the complexities, which will serve as a representation of the user's knowledge with respect to a given task.

III. AGENT-BASED MODEL

The goal of our cognitive agent is to understand the information needs of the user, involved in a web-based knowledge-gathering task and assist them accordingly. To accomplish this, the agent firstly needs to understand the task of the user and the complexity of the task, which we call as inherent complexity, and secondly estimate the proficiency of the user in their task, in order to assist him accordingly. We call the latter as the perceived complexity which is a reflection of the knowledge levels or proficiency of a particular user with respect to a particular WKG Task.

In order to achieve these goals, the agent is cognitively designed using the Belief, Desire and Intention model [17], which forms the basis of intentional systems [18] that are designed for understanding and exhibiting rational behavior in complex systems. We designed the rational agents using three sets - set of beliefs, set of desires, and set of intentions. The set of desires are pre-built for these agents. Then the agent is fed with the set of intentions, which are the activities carried out by the user through our system. The agent then should work to deduce the set of beliefs from the intentions of the user and from the assumed desires.

In our case of web-based knowledge gathering task, a user's desire would generally be to gather the required knowledge within the time constraints. However a user may also desire to gather as much knowledge as possible within the time constraints, over and above the requirements for a given task. The user-dependent factors of perceived complexity (of the task), and time (that is available to be spent – a constraint), greatly influence the knowledge-gathering activity, and these become the part of default desires in every agent.

The intentions indicate the actual desires of the user. The actions and the resultant activities carried out by the user form the intention set of that user and forms the basis for all our inferences about the desires and beliefs, with respect to a particular task. However in a WKG task, a user is bound to make wrong selection of a resource, or use inappropriate keywords or queries, and hence need to be examined carefully. So, a task may not be complex, but the activities carried out by the user may some times give the impression that it is a complex task – indicating the level of perceived complexity. It is for this reason, that every individual action by the user cannot be construed as an indicator of the desire or belief, and hence a sequence of actions has to be examined.

The beliefs of the agent should reflect the knowledge of the users, with respect to the task, being carried out by them. The knowledge and beliefs of a user are in turn reflected in the complexities – inherent complexity of the task, and the perceived complexity. Hence through the activities in the intention set, we design the estimation algorithms for estimating both the complexities that form the representation of the knowledge of the user, as a belief for our agent. The complexity estimates will also serve as an aid to understand the knowledge gaps of the users.

Relating the above Belief-Desire-Intention sets with the goals of the agent, the first goal is to understand the task complexity or the inherent complexity, which ideally requires determining the task, while the second goal of understanding the perceived complexity requires arriving at an estimate of the proficiency of the user involved in the task. The agent can then use these estimates to provide appropriate references to resources to assist the user accomplish their task. In the following sub-sections, we introduce mechanisms for the agent to estimate the inherent and perceived complexities.

A. Understanding the inherent complexity of the task

The inherent complexity is a reflection of the intricacies of a given task. There can be a lot of difference in the inherent complexity levels of the tasks, and even between similarly looking tasks. For instance, in the domain of finance and investments, finding a best performing international feeder mutual fund – fund of funds across all categories is of higher complexity than finding the best performing mutual fund in equity category as the sources of information are more and wider in the former than in the latter.

The ideal means for deriving the inherent complexity of a task is to understand the task, and its intricacies behind it, which requires the support of an expert system. As this is infeasible for every kind of task, our approach, through the agents is to deduce the complexity estimate from the activities of the user. First we organize the activities of the user in the intention set into the following cognitive states: Search, Filter, and Gather [1]. The agent then analyzes the following: Keywords used in the search state; links selected in the Filter state, corresponding to the keyword, and the resources explored in the Gather state, corresponding to the links selected in Filter state along with the amount of time spent on a particular resource in the Gather state.

The aggregation of keywords used, query refinements all in the search state indicate the kind of resources the user is looking for. The resources used in the gather state are classified as passive and active; passive resources typically are the documents with varying levels of complexity, while active resources, represent interaction mechanisms undertaken by the user such as raising a question in discussion forum. We organize all this information using a semantic link network – SLN – that constructs the keywords and the resources explored by an individual user, and compare it with a reference SLN, in order to discover the appropriate resources for a given task [19]. The reference SLN is a result of several users executing same or similar kind of tasks, and its purpose is to reveal the cognitive structure of the knowledge required for addressing a WKG task. The basic concept and usage of SLN can be found in [20].

We use this kind of SLN for identifying the core attributes of a task, being executed by the user. The SLN can segregate between similar keywords and totally different keywords used by the user, for their queries. It can also aggregate related resources associated with a keyword. The reference SLN can serve the purpose of identifying the appropriate keywords, and the resources related to the task, along with an estimate of the complexity level of the commonly used resources – which could serve as an indication of the inherent complexity. Hence an agent can feed in their intentions to the SLN and can have the parameters for calculating the inherent complexity of the task. The SLN can also identify the same task executed by the user in multiple sessions, and can be used for segregating the activities of the user related to the task, from other activities in a session.

The core attributes of a task identified by the SLN, when mapped to domain ontology can reveal the complete structure of the task and its inherent complexity. The agent can then arrive at an estimate using the various attributes and relationships that are associated with a concept. Hence, the role of ontologies in this situation can assist the agent in understanding a user's task and its inherent complexity. However, in many instances, complete domain ontologies may not be available, for all domains. We assume such cases, and therefore leverage on the SLN for estimating the inherent complexity.

We utilize the aggregation and segregation mechanisms of SLN, with respect to the keywords and resources to arrive at the following parameters for estimating the inherent complexity. Every different keyword used would indicate a sub-topic related to the task, and every subsequent keyword can reflect the different stages and overall depth of the task. Hence we use this as one of the parameter. Similarly the number of resources used by the user and their types (active or passive) in the Gather state can also serve as another parameter. The reference SLN can add to it, with the complexity level at which a resource is generally tagged. So, with the combination of these factors, we arrive at the following algorithm for our agent to arrive at an estimate for the inherent complexity.

```
float inherentComplexity(kword[])
Begin
    double est = 0.0;
    int actCount = 1;
    for each 'i' in kword[i] do
    /* indicates a different sub-topic */
        est += (i+1 * 0.25);
        for each resource res[j] associated with kword[i]  do
                if (res[j].type == active) then
                /* Active Resource */
                        est += actCount* 0.50;
                        actCount++;
                else  /* Passive resource */
                        est += res[j].level * 0.25;
                end if
        end for
    end for
    return est;
End.
```

The inherent complexity estimation algorithm work as follows: As every different keyword indicates a new sub-topic (as we detect and exclude rephrasing and refinement of keywords, through the States), the estimates are substantially increased for every subsequent keyword. The active resources would generally be an indicator of higher complexity. Hence for every, subsequent active resource utilization, the complexity estimates are increased substantially. Similarly for passive resources, the estimates are increased by atleast 0.25 for each utilized resource. Also, a resource gets registered only if it is associated with a proper keyword in the SLN, which is an indication, that the user has derived something from that resource, for his task. The resource complexity level, as estimated by the SLN, would typically range from 1 to 5, indicating low to high complexity, which is multiplied by a factor of 0.25. In other words, accessing a level 2 resource is equivalent to accessing two different level 1 resources.

B. Understanding the perceived complexity of the task by the user

Perceived complexity is about how the users perceive and execute the task, and therefore highly dependent on the knowledge levels of the individual user. For instance, a user who does not understand the concept of mutual funds have to first acquire knowledge about it, before performing an analysis of various top ranking equity-oriented mutual funds. Hence, a task that may look simpler to one user may look difficult to another, and therefore will have an impact on the user actions. The beliefs and knowledge of the user along with the inherent complexity of the task influence the perceived complexity.

The following factors were identified for estimating the perceived complexity: rephrasing of the query, query refinement, selection of resources from the search engine results, time spent in a session, and total sessions, related to the task. Rephrasing of query is observed, when the user has not moved out from the current search state, or observed when no links were selected for a given search query. The more the number of successive rephrasing, the estimate value goes up higher. Query refinement is observed when the user moves from Search to Gather state (by spending some time in atleast one of the resources) and then comes back to the search state with a new query. This factor, though indicates the progression of the user, also indicate the perceived levels of complexity in the task. The selection of resources from the search engine results were also considered as a factor, as it indicates the proficiency of the user indirectly. A resource that appears beyond the first page or the 10th item, if selected may indicate that the user might not have used an appropriate query. Similarly the amount of time spent, and the task being undertaken in multiple sessions are indicators of proficiency.

All the above factors were given a uniform value of 0.25, except for the time spent, which was set to 0.10. The perceived complexity estimate was calculated whenever the operations corresponding to these factors were observed. The higher the estimate value, higher the perceived complexity of the task by the user. Each successive repetition of query rephrasing was penalized with a multiple of 0.25, while repeated sessions (more than one) were penalized with a multiple of 0.25 and selection of resources beyond the first 10 results or the first page of the search engine results, were also imposed a penalty. Hence, more further the resource, higher the estimate value for the complexity. The time spent in minutes is multiplied by a constant factor of 0.10, and so for every hour spent by the user, the complexity goes up by 6, and every query refinement was multiplied by a constant factor of 0.25, as query refinement is generally an indication of progress of the user in the given task. The following algorithm is used by the agent to arrive at an estimate for the perceived complexity.

float **perceivedComplexity**()

Begin

double est = 0.0;

/* rephrase is a global variable initialized to 1 */

for each user activity in the sequence **do**

if query was rephrased **then**

/* Observed, when no links are selected for a search and when the user has not moved out from the current search state. If moved out to a different search state, then reset rephrase to 1. */

if curSearchState == prevSearchState **then**
est += rephrase * 0.25;

else

rephrase = 1;

est += rephrase * 0.25;

end if

rephrase++;

end if

if query was refined **then**

/* Observed, when the user, moves from Search to Gather State and then comes back to search state with a new query */

est+= 0.25;

end if

if first resource is picked only at the nth Page of the search results, **then**

/* Filter state has the page number of the listing associated with the resource */

est+= (n-1) * 0.25;

end if

end for

for each session, a user involved in the same task **do**

/* Session is from Login to Logout, SessNo, indicates the Nth session, (starts from 1) the user is working on for his/her task. */

est+= Tot time spent in minutes in this session * 0.10;

est+= (SessNo - 1) * 0.25;

end for

return est;

End.

IV. EVALUATION

A. Simulation and Learning

There were two main purposes for the simulation exercise: One is to determine the typical estimate values for the inherent and perceived complexity, such that the values can be categorized into qualitative representations, such as low, medium and high. The other is to determine how effective were the agents in estimating the complexities during the execution of the task.

The log of the activities carried out by many users involved in various knowledge-gathering tasks in multiple sessions, from a prototype web-based portal was used as a dataset for the simulations. The portal facilitated knowledge-gathering activities, which consisted of search interfaces (provided by popular search engines), and other knowledge-gathering tools such as E-notes, resource management, opinion-gathering tools and interacting tools. The activities in the log were recorded with the timestamp that was used by the agent to weave the sequence of activities of a user beyond sessions. A total of 80 sessions related to knowledge-gathering tasks was filtered out from the logs. The total number of users within the filtered logs was 18. Using the SLN, we identified 4 unique tasks.

The agent was then fed with the logs of sessions in a sequence, related to a particular user and a particular task. The agent has to invoke the inherent and perceived complexity estimation algorithms, at the end of each session, as well as at the end of the task. In a live environment, the agent would be invoking these functions, during the execution of the task by the user. The execution will start as soon as an activity gets registered in the Gather state. The details of the experiment are discussed in the next section. The results at the end of the task are given below in Table 1.

TABLE I. SIMULATION-ESTIMATES

Complexity	Value Range			
	Task 1	***Task 2***	***Task 3***	***Task 4***
Inherent	4.5 – 4.75	6-6.75	3.5 – 3.5	4.25-5.25
Perceived	9 – 20.10	12.20-18.40	6.60-10	8.80-17.20

In the logs used above, the task belonged to two different domains: One in the domain of Computer Network Security and another in the domain of Investments in Mutual Funds. We wanted to have these simulations done for diverse domains, in order to analyze their values, so that they can later be applied to any knowledge-gathering task. Table 1 – Simulation Estimates, reveal that the inherent complexity for given task is between a narrow range, and not equivalent for all users. This is because, our deductions are based on the effective results obtained from the actions of the users. In other words, if a user accesses resources that may not be required for the task in hand, then the inherent complexity will also increase. However, the simulation results indicate a wide variation in the perceived complexity, which depends on the skills of the user, pertaining to the task.

An interesting observation is that the lowest perceived complexity of different tasks is approximately two times that of the lowest inherent complexity estimates of the task. It can be noted that the values for perceived complexity were in a wider range, because, of the varying proficiency of the different users, indicated through their behaviors. Based on the above complexity levels, for the four different tasks, belonging to two different domains, we arrive at the following qualitative estimates, for the complexity.

TABLE II. QUALITATIVE COMPLEXITY ESTIMATES

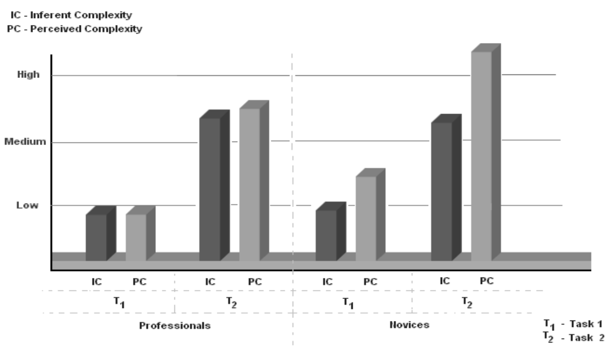
Complexity	Qualifiers	Range
Inherent	Low	<=4
	Medium	>4 & <6
	High	>=6
Perceived	Low	<=9
	Medium	>9 & <15
	High	>=15

B. Live Experiment and Results

The main purpose of experimentation is to evolve a method to invoke the complexity estimation algorithms at appropriate instances, during the execution of the task by the user. Also, at the end of the task, we will be able to determine the fairness of our complexity estimates.

We identified two classes of users: One class of users, professionals, consisting of three members, were proficient with the domain of Network Security, and others - novices, consisting of three members had knowledge about the system-level security, but not at the network level. We identified two tasks, T1 and T2. Task T1 is to find out the “Concepts and Techniques behind Intrusion Detection system”, and Task T2 is to understand “Flow based anomaly Detection Techniques in Network Traffic”. Task T1 is of low inherent complexity, while Task T2 is of high inherent complexity. Both the tasks were executed by both the classes of users. We used the same prototype web portal that was used for simulations, for the experiment. The users were also set a deadline to complete the task and prepare a report through the system within 72 hours. However the users can work on this task, in multiple sessions, and preferably may not interleave with other activities.

As the users logged in, an agent created for a particular user, monitored the activities of the user, and made estimations of inherent and perceived complexity of the task, as the user progressed through the task. The agents were programmed to provide their first estimate, after an activity got registered in the Gather state. After the first estimate, the second estimate was made after the third activity gets registered in the Gather state or at the end of the session, whichever happens at the earliest. In our experiment, the number of times the algorithms were invoked ranged between 2 to 8. Figure 1 – shows the results, that vindicates our estimations. As it can be observed, for the complex task T2, the inherent complexity estimate for the novice users shows lesser complexity than for the professionals, which can be explained that the professionals had related knowledge and hence had used more specific keywords and resources, than the novices, who were unaware of it. Also our experiments indicated that, given a task, a user generally decomposes the task, after a few iterations, depending on their information need and proficiency, and in this case, the professionals had used more decompositions than expected, that resulted in more different keywords. This might due to the reason that the professionals were enthusiastic in discovering and learning more. The overall results indicate that all the estimates were up to our expectations.



V. Conclusion

In this paper, we had devised mechanisms for estimating the inherent complexity of a Web-based knowledge-gathering (WKG) task, and the perceived complexity of the task by the user, using a number of observable parameters. We had implemented this through cognitively modeled BDI Agents that formulated the beliefs of the user, using the intentions (activities of the user) and the assumed desires. The typical quantitative estimates that were arrived at, using the simulations were then assigned qualitative values, to make better sense of them. The experiment and the results also vindicated our estimation methodology.

The estimation of the inherent complexity of the task, and the corresponding perceived complexity by the user, has lead to the quantification of the knowledge gaps in the user, seeking information, for accomplishing a WKG task. The complexity estimates arrived by the agent is completely based on the observable behavior of the user, and these estimates indicate the complexity from the beginning of the task by that user to the current situation the user is in. These estimates can then be used for guiding the user to better and more appropriate resources, during their process of WKG. The agents performing the estimations can also share their estimates with other agents that might be involved in assisting users in similar WKG tasks, in order to gain a better understanding [21].

References

[1] Balaji Rajendran, "Socio-Contextual filters for Discovering Similar Knowledge-Gathering Tasks in Generic Information Systems," in *Proceedings of the IEEE ISI 2008 PAISI, PACCF, and SOCO international workshops on Intelligence and Security Informatics,* Lecture Notes In Computer Science, 2008, pp.384-389.

[2] Collins, H, "The structure of knowledge", Social Research, Vol. 60, No. 1; 1993, pp. 95-116.

[3] Atsushi Fujii, "Organizing encyclopedic knowledge based on the web and its application to question answering", *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* , 2001, pp.196-203

[4] Andrei Broder, "A taxonomy of Web search", SIGIR Forum, Vol. 36, 2002, pp. 3-10.

[5] Sycara. K, Paolucci, M, Giampapa, J, and van Velsen, M. "The RETSINA MAS Infrastructure", Journal of Autonomous Agent and MultiAgent Systems, 2001.

[6] Macal M Charles, North, J, Michael, "Tutorial on Agent-Based modeling and simulation Part 2: how to model with Agents", in *Proceedings of the 2006 Winter Simulation Conference*, 2006.

[7] Gilbert Nigel, Agent-Based models, Sage Publications, London, 2007, ISBN: 9781412949644

[8] David Vernon, Giorgio Metta, and Giulio Sandini, "A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents", IEEE Transactions on Evolutionary Computation, Vol .11, 2007, pp. 151-180

[9] Lieberman, H., "Autonomous Interface Agents", in *Proccedings of ACM Conference on Computers and Human Interaction (CHI-97),* 1997.

[10] Joachims, T., D. Freitag, T. Mitchell, "WebWatcher: A Tour Guide for the World Wide Web", in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan, August 1997*

[11] Huhns, M., ed. Readings in Software Agents, Morgan Kauffman, 1997.

[12] Hill, W., Stead., L., Rosenstein, M., Furnas, G., "Recommending and Evaluating Choices in a Virtual Community of Use", in *Proceedigns of ACM Conference on Computers and Human Interaction (CHI-95)*, 1995.

[13] Wittenburg, K., Das, D., Hill, W., Stead, L., "Group Asynchronous Web Browsing on the World Wide Web", in *Proceedings of the Fourth World Wide Web Conference, December 1995*, 1995.

[14] O. Hoeber, "Web information retrieval support systems: The future of web search," in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '08), vol. 3, 2008, pp. 29-32.

[15] Campbell and C. J. van Rijsbergen, "The ostensive model of developing information needs," in *Proceedings of COLIS-96, 2nd International Conference on Conceptions of Library Science*, Kobenhavn, DK, 1996, pp. 251-268. [Online]. Available: http://citeseer.ist.psu.edu/campbell00ostensive.html

[16] Chunyan Miao, Qiang Yang, Haijing Fang and Angela Goha, "A cognitive approach for agent-based personalized recommendation", Knowledge-Based Systems, 20(4), 2007, pp.397-405

[17] Rao, A.S., and Georgeff, M, "BDI agents: from theory to practice", in *Proceedings of th first international conference on Multi-agent systems, (ICMAS-95), San Francisco, CA*, 1995, pp.312-319.

[18] Dennett, D.C, The intentional stance, MIT Press, Cambridge, MA, 1987

[19] Balaji Rajendran, Iyakutti K, "Semantic Link Networks for Modeling Cognitive Structures related to a Web-based Knowledge-Gathering task," unpublished.

[20] H. Zhuge, The Knowledge Grid. World Scientific Publishing Co., Singapore, 2004.

[21] Balaji Rajendran, Iyakutti K, "Socio-contextual Model of Knowledge Sharing among User Assisting Agents", *Proceedings of the IEEE ISI 2009 international workshop on Social Computing (SOCO)*, 2009, in press.