Contextually Cooperating Agents for User Assistance in Web-based Knowledge Gathering Tasks

Balaji Rajendran
Centre for Development of Advanced Computing,
68, Electronics City, Bangalore 560 100

Iyakutti K
Department of Computer Science
School of Physics,
Madurai Kamaraj University, Madurai

ABSTRACT

The availability of rich and diverse nature of unstructured and hyper-textual content in the Web coupled with the power of modern search engines either necessitate or lure the users to undertake the task of Web-based Knowledge Gathering (WKG). WKG is typically a complex and cognitive task under taken by the users on the Web for their learning, exploration, discovery investigation or decision-making. Though search is the primary operation in WKG, the involvement of many other human factors makes it difficult for the search engines to address this special requirement.

In this paper, we argue that an active support system can be built around today's search engines that could assist the users in their WKG tasks. We propose an intelligent support system composed of cooperating agents that build contextual structures in the form of semantic link network by observing the task of their users. The agents share these structures and seek assistance from other agents who may provide their task structures based on the similarity of the tasks. The requesting agents then use these newly obtained task structures to extract out information that would help their users in achieving their goals. We implemented this as a prototype system by wrapping around a popular search engine and carried out user evaluation with tasks from diverse domains. The results showed that the users were able to save considerable efforts in the process of knowledge gathering and hence the system was able to support its users.

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

Web-based Knowledge Gathering Task, WKG, Contextual Structures, Contextual Cooperation, Intelligent Agents, Support System, Semantic Link Network, Exploratory Web Search

1. INTRODUCTION

The open, distributed, diverse, and massive structure of the Web has opened the gates for knowledge gathering tasks by its various users. This concept of Web-based Knowledge gathering (WKG) is being increasingly attempted by many users, for the purposes of learning, investigation, decision-making, discovery and exploration.

A WKG task is typically a complex task requiring: iterative search operations, references to multiple sources, spanning multiple sessions, and successive refinement of goals of the task. This complexity is compounded by the high cognitive loads experienced by the users due to the hyper-textual nature of the web documents, and the need to correlate information from multiple resources. It should be noted that the same characteristics of Web – distributed, diverse, large-scale - that makes it ideal for knowledge gathering, makes it a difficult task to perform by users.

The challenges faced by the user in WKG can be organized across a sequence of three states: Search state - identification of right keywords, query formulation, and query refinement, Filter state – selecting the appropriate results from a set of results; trying to grasp as much knowledge as possible from the snippets of the results; and Gather state – learning or gathering the information required for the task, from the selected resources, and organizing the information gathered so far in a coherent way (removing conflicts, and redundancies mentally). The actions taken in the Gather state can lead to any of the previous states or finish state – i.e., the learning occurred in Gather state will influence the use of keywords, and selection of resources in the next iteration of the search or filter state [17].

The search systems of today are tuned more towards addressing navigational queries or simple information queries and are primarily intended to solve the majority of the queries raised by various users. The characteristics of WKG differ from Web Information Retrieval [11], where the primary focus is on retrieving the best documents matching a query given by the user, while in the former, it is about addressing the various information needs required to accomplish a user’s task. WKG is more related with Contextual Web Information Retrieval [8], where the user’s need is given importance. However information retrieval based on a user’s context is only a part of WKG that consists of other activities from the human side. The present day search systems offer little support for WKG tasks, and hence a support system encompassing the search is required for assisting the users involved in the complex task of WKG.

Such kind of a support system can be of two types: passive or active. A passive support system can assist the user in keeping track of the queries used, links explored, creating, saving and retrieving the user notes related to the task, and other such interface assisting features. An active support system, extends the passive system, to be an intelligent support system that aims to understand the task of the user, and estimate the user’s proficiency with that task, in order to provide contextual inputs in the form of highly relevant resources at a particular instant of time, based on the user’s knowledge and task complexity [19] and the most appropriate keywords to use for exploring further to progress in the particular task.

Such an intelligent support system needs a framework equipped with the complete knowledge of the Web's information. As this appears infeasible today, and as attempts to achieve the vision of semantic web [12] is still in progress, users would continue to make their own efforts for knowledge gathering. However, we believe that the present day state-of-the-art tools can do much better to provide active support to the users in such kind of WKG tasks.

In this paper, we propose an active and intelligent support system based on the behavioral information gathered from the users involved in WKG tasks. We use agents that are modeled on BDI style architecture, and is associated with every user, who uses the system for WKG. The basic concept is that the agent learns and constructs a contextual structure in the form of a semantic link network [25] based on the keywords used and the links (resources) explored by the user. This structure reflects the contextual characteristics of the task that the user is trying to address. The agent uses this structure and seeks assistance from other agents that might have assisted their users in similar such tasks. The inputs collected from other agents are again aggregated into a semantic link network that could be used by the agent for assisting its user, as the user progresses. The process of seeking and obtaining assistance from other agents is made possible through the contextual structures represented in the form of semantic link networks, thereby paving the way for contextual cooperation among agents. Section 2 presents the background and related works to WKG and agent-based systems for user assistance. Section 3 details our approach, with algorithms for constructing the contextual structures, and for gathering knowledge from others agents. Section 4 gives the details of implementation, and Section 5 gives the process of user evaluation and results.

2. RELATED WORK

Web-based Knowledge Gathering tasks are kind of information foraging activity [16] carried out by users on their own in the web, with the help of search engines. The primary intentions could be to explore, learn, discover, or investigate in order to take a decision. The earlier works focused on improving the search engines to address these kinds of tasks.

The queries used by the users in search engines were analyzed and categorized into information queries, navigation queries and transaction queries [2]. The information queries were the ones that the users were looking for support, and soon the major search engines implemented query suggestion features. The query suggestion feature also included correction of spellings and use of different words in the keyword string. However, it was soon realized that search engines were looking at queries of the user and not the tasks of the user, for which the user has been approaching the search engine.

This lead to the research in identifying and focusing on the information needs of the users and the development of contextual strategies for performing contextual information retrieval being addressed [8]. These contextual strategies typically focused on user profiling and web mining techniques and most of them used implicit methods for identifying the user's need. For instance the online discovery of frequent path traversal patterns, through a user's click stream data has been explored in [13], and a cognitive structure for contextual information retrieval is proposed in [23].

The information needs of a user are mainly influenced by the task that the user needs to perform, at a given instance of time, which lead to the analysis of the Web-based information tasks and the proposal of different taxonomies [21] [9]. The kind of search involved in these types of tasks was referred as exploratory search [15] and the need to redesign the search interfaces or to build features such as WordBars [7] or to have a support system [6] for the users involved in such exploratory tasks were researched [24].

Software Agents have evolved as a paradigm of programming over the years. Their usefulness in implementing systems that support users is popular, because of their inherent autonomous and communicational designs. With respect to information retrieval, agents have been modeled as intelligent interface agents for Web information retrieval in many works as in [14] and [10]. Agents were also used for personalized information retrieval, particularly over the Web as in [3] and [1].

In our approach we observe the user behavior and convert it into a contextual structure to identify a task implicitly. The contextual structure is then used by intelligent agents for communicating, sharing and learning from similar agents located in an environment designed for Web-based knowledge gathering tasks. The inputs collected from other agents are utilized by the agent, for recommending and supporting the user involved in WKG tasks. This focused approach of assisting the users by learning from the activities of the users themselves is the salient and distinguishing feature of our technique, which is explained in detail in the next section.

3. OUR APPROACH

Our approach is first to identify and understand the WKG task of the user. We propose a simple BDI agent-based model that operates in an environment designed specifically for carrying out WKG tasks and continuously monitors the behavior of the user, to arrive at a fair description about the task. The agent's task is to deduce the desires, intentions and belief of the user with respect to the WKG task from the observable actions of the user. As a user proceeds with a task, the agent observes the actions (behavior) of the user, and represents them in a contextual structure corresponding to the WKG task. This contextual structure (SLN) is then refined and used for communication with other agents who evaluate the structure for similarity with their existing tasks structures and present the closest one as the result, which is used by the requesting agent for assisting its user.

3.1 An Environment for Web-based Knowledge Gathering

The agents operate in an environment designed for WKG tasks. The environment is a Web-based system that consists of a search interface that can be used by the users for querying. The keywords used, and links explored by the users are captured in this environment, and can be viewed by the user while he continues with his task. The system consists of agents that correspond to each user, and studies their patterns in order to assist them. The agents are cognitively designed with a simplified BDI architecture [20] and are designed to cooperate with each other without expectation of any rewards. The environment also consists of tools such as memory aids for note-taking, marking and labeling of particular resources, and features for recording the feedback on the usefulness a particular resource.

3.2 Construction of the Contextual Structure

Every user is associated with an agent. When a user starts with a task, the agent corresponding to the user gets activated and monitors the actions of the user. The user's state during the performance of a WKG task can be mapped to any of the following: Search, Filter, Gather and Decide states; and typically a user will be seamlessly switching from one state to another [17]. As the user uses various keywords through the available search interface they are recorded by the system which helps the agent to construct a contextual structure. The links explored by the user, through the results obtained from the search engine are also recorded and associated with the respective keywords used by the user.

The keywords used and resources explored by the user constitute the intentions of the user. The agent has to now implicitly arrive at the beliefs of the user with respect to the given task. The beliefs indicate the knowledge level of the user, for that particular task, and the interaction of this agent with other agents might lead to the identification of the required elements for accomplishing the task successfully, which could be used to guide its user and bridge their knowledge gap. The desires of the BDI agent are obvious – to obtain the knowledge required for the task at the earliest. These kinds of BDI agents form the basis for intentional systems [4].

The contextual structure is organized as a semantic link network [25] with the keywords as the anchors for the resources under them. The resources are represented as URLs along with their captions extracted from the results returned by the search engine. The keywords are in turn associated with the tasks, for whose purpose these keywords were used by the user. The tasks are represented as labels that are either set by the user, any time during the execution of a WKG task, or set automatically to a timestamp value by the system for unique identification. Figure 1 gives a conceptual description of two contextual task structures T1 and T2.

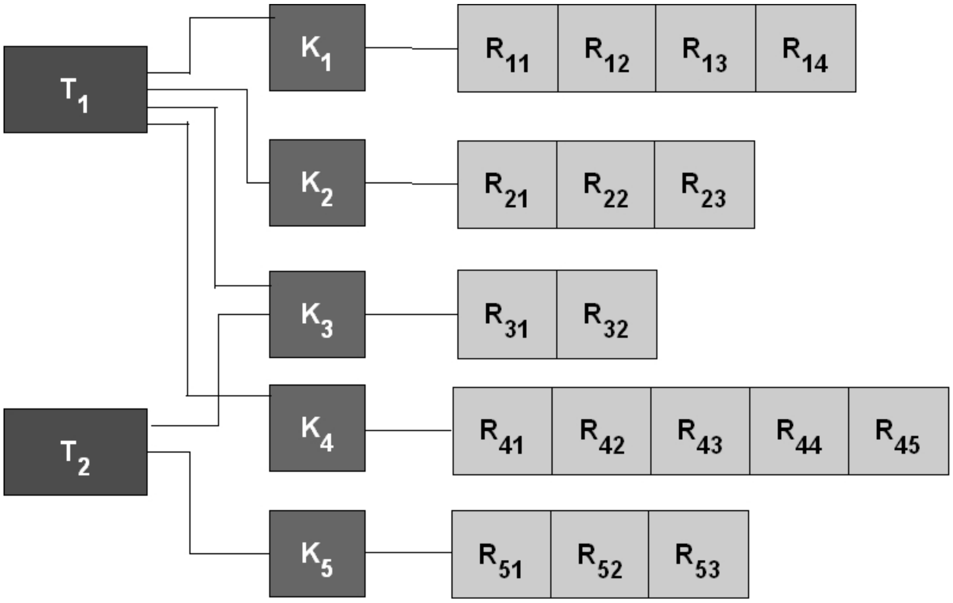

Figure 1. A Sample Contextual Structure of two WKG Tasks T_1 and T_2.

This contextual structure, being built by the agent, is reorganized during and after the construction process, in order to bring consistency and clarity for the identification and understanding of the task by the agent. It would also help an agent to arrive at an estimate for the complexity of the task based on the user's action [19]. The reorganization would effect changes to the structure in terms of resource, keywords and task associations. The process of reorganization is as follows: When a resource anchored with a keyword is already present in the structure anchored to a different keyword then the similarities between the two keywords and their associated resources are examined. If they share more than half the number of resources then one of the keyword needs to be removed, and those resources that are not present in the other has to be associated with it. The choice of the keyword for removal has to come from a master structure, being built by a master agent. We term the master structure as the reference semantic link network.

The reference semantic link network is an aggregation of all the structures constructed so far in the system, by various agents. This structure due to its nature of composition and continuous reorganization makes it a master structure that has tasks composed of the widely used keywords and prioritized resources associated with those keywords. It also has the ability to distinguish a task from another and the ability to integrate similar tasks. This master structure acts as the source of reference for the structures constructed by the agent, in determining a task, finding and using an appropriate keyword etc… However, this structure would be useful for an agent, only when its user is executing a task that is same or similar to an already executed task by some other user.

This process of constructing a contextual structure with assistance obtained from a master agent will help the agent to identify a task, and understanding it through the keywords and resources. Each of the keywords present in the reorganized structure would then indicate a concept in relation to the task. The construction of the contextual task structure for a task is summarized in Algorithm 1.

Algorithm 1: Constructing the Contextual Structure of the Task
Inputs: Keyword K and it's associated Resource List RL; Task Structure TS;
Output: New Structure TS

```
void constructTaskStruct (K, RL, TS )
Begin
        if (TS == NULL) then
                TS.append(K);
                exit;
        end if
        /* Find whether K already exist in TS. If yes, append
the new resources in RL to K in TS. */
        for each keyword $K_i$ in the existing structure TS do
                if (K == $K_i$) then
                /* Keyword K already exists in TS; so add
the new resources in RL to K */
                for each resource $R_j$ in RL do
                   if $K_i$ does not refer to $R_j$ then
                     $K_i$ .append ($R_j$);
                end for
                exit;
                end if
        end for
  /* Keyword K not found in TS; Request the reference semantic
link network for a better alternative keyword used commonly for
the Keyword K and its resources */
        newK = master.alternativeKeyword(K);
        if (newK == K)  then
/* K is either most commonly used or has not been encountered
by the master agent so far. So add K and RL to TS */
        TS.add(K);
        //K already contains the resource list RL
        exit;
        else
                /* Find whether newK is in TS */
                for each keyword $K_i$ in TS do
                  if (newK == $K_i$) then
/* User has used a modified version of the same query, and
hence has not moved to a different concept */
                   for each resource $R_j$ in RL
                     $K_i$ .append ($R_j$);
                   exit;
                 end if
                end for
        /* K and its replacement newK are new to TS;  */
                for each resource $R_j$ in RL
                  newK.append ($R_j$);
    TS.append (newK);
        end if
End.
```

3.3 Contextual Cooperation among Agents

As the agent constructs a contextual structure of the task being executed by its user, it puts that structure seeking assistance from other agents that might possess a similar contextual structure. Each of the agents in the system then examines the presented structure and its relatedness with their own existing structures. If there is a significant match of a structure that exists in their database, it is presented to the agent seeking assistance along with a value indicating their relatedness between the structures. The relatedness is calculated as a function of matching keywords, matching resources, and task labels. This makes sense because of the reorganization of the contextual structures carried out by every agent, leads to the use of uniform keywords in similar kind of tasks with inputs obtained from the master agent. The relatedness value falls into three categories: High Relevancy – when two structures share more than half the number of keywords and their associated resources, Potentially Relevant – when two structures share either half the number of keywords or half the number of resources, and Low Relevancy – when two structures neither share half the number of keywords nor half the number of resources. The responding agents will typically put their task structure for offer only if they are highly relevant, and only upon sensing repeated requests, will offer the potentially relevant structures. There is a special case within the cases of high relevancy, when the agent will not offer its tasks, if it does not contain any new keyword or resources that are not present in the contextual structure of the requesting agent.

The agent that sought the assistance then validates the offers from various agents that might have responded to its request. The requesting agent has to select such a structure that would enhance the knowledge of its user and at the same time be relevant to the current task. The agent can also select more than one structure and integrate them into a new structure. Basically the agent starts by selecting the most related structure received, and scans it for the new keywords and resources, that are not present in its existing structures and recommends them at appropriate instances to its user. For instance if two new keywords are found then they represent two concepts that might be required for accomplishing the task. The resources associated with the keywords will be presented one after the other as the user progresses in the task. In some instances, even the keywords would be suggested to the user by the agent. The algorithm for the enrichment of the contextual structures is presented below, which internally uses Algorithm 1 for the construction of the new task structure.

Algorithm 2: Enrichment of a Contextual Structure.
Inputs: Task Structure TS1, of the requesting agent; Task Structure TS2 of the responding agent selected for enrichment.
Output: A new enhanced structure TS3 that could be used to enrich TS1.

```
Task Enrich (Task TS1, Task TS2)
Begin
        K[] = List of keywords that are in TS2 but not in TS1
and TS3;
        for each keyword i in K[] do
            RL = Get the Resource List associated with K[i];
            constructTaskStruct (K[i], RL, TS3);
        end for
        NewResKey[] = List of keywords containing those resources that are in TS2, but not in both TS1 and TS3;
/* These are the resources associated with a keyword, that already exists in TS1; */
        for each keyword i in NewResKey[] do
            RL = Get the Resource List associated with NewResKey[i];
            constructTaskStruct (NewResKey[i], RL, TS3);
        end for
return TS3;
End.
```

4. IMPLEMENTATION

The above concepts are implemented as a proof-of-concept Web application. The application uses the standard struts framework and AJAX based client interfaces. We use the Google AJAX Search API [5] for implementing the search interface. The system has been designed in such a way that the user does not have to leave the environment, while performing the WKG task. This has been achieved through the use of AJAX-based dockable windows that open the resources clicked for exploration by the user, from the search results returned by the search engine.

The agents are implemented as a program that tracks the activities of their respective users and organizes them into a contextual structure at the server side of the system. The contextual structure is represented through a hash-map based data structure, which is converted to the relational format for storing in the database with each user agent having access to its own database. These structures can be retrieved back which will help the users to continue with their task in multiple sessions. The agents during the process of construction of the contextual structure, passes the information to a master agent that constructs the reference semantic link network. The master agent keeps receiving inputs from other agents during task execution by various users. The master agent will also receive requests from the agents for finding a suitable keyword to anchor resources in their contextual structures, by giving the current keyword.

The algorithms described are invoked by the agent program when required. The contextual cooperation among agents is implemented through a variation of the Contract Net Protocol [22], with an agent that plays the role of a manager in receiving support requests from agents and notifying other agents about it. The manager agent also passes back the received responses back to the requesting agent and therefore acts as an intermediary between requesting agents and responding agents. In this way, all an agent needs to do is to put its contextual structure to the manger and wait for its response. The manager agent is also responsible for handling simultaneous requests from various agents. Figure 2 illustrates the role of manager agent in handling the requests for assistance in various task structures.

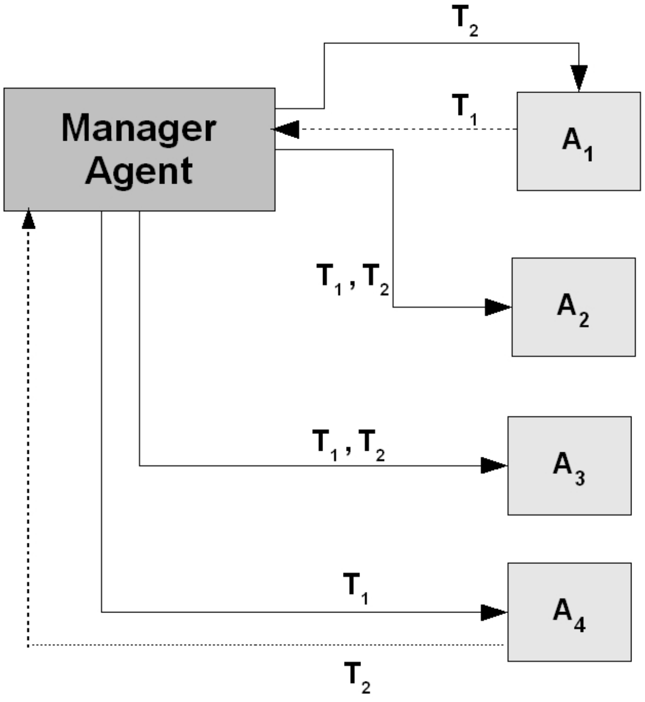

Figure 2. Manager Agent Receiving Requests from A1 and A4, and Requesting Assistance.

5. EVALUATION

The user evaluation was carried out through the following experiments. Eight users organized into four groups participated in the experiments, with each group consisting of two users. Four tasks T1 to T4 were identified from two different domains and allotted to different users. The participants were selected in such a way that they were naïve to the given WKG tasks. Eight agents A1 to A8 were spawned for assisting the eight users. The tasks T1 and T2 were from the domain of Computer Network Security in the topic of Intrusion Detection Systems, and tasks T3 and T4 were from the domain of Architectural Engineering in the topic of Deep Foundations. Task T2 was chosen in such a way it was more complex than T1, and T4 is complex than T3.

The users as a proof of their task completion had to submit the online notes prepared by them during their task execution. They were also given enough time and were allowed to carryout the task in multiple sessions. In round I, inputs were not sought by the agents, and hence the users were left on their own, with only the passive support system to perform the tasks. As the users performed the tasks, the agents A1 to A8 constructed the contextual structures based on the keywords and resources explored by their respective users. Table 1 summarizes the task structures of the agents A1 to A8, the complexity of the tasks, and the keywords and resources in each structure.

Table 1. Summary of Task Structures in Round I

Agent	Task	Complexity	No. of Keywords	No. of Resources
A1	T1	Medium	3	8
A2	T1	Medium	2	3
A3	T2	High	4	17
A4	T2	High	5	16
A5	T3	Medium	3	8
A6	T3	Medium	3	7
A7	T4	High	4	18
A8	T4	High	4	15

The experiments were repeated – round II - with users now executing tasks from the other domain. The users who executed tasks of Network Security now executed tasks in Architectural Engineering and vice versa. As the users executed the tasks, the agents started constructing the contextual structures of the respective tasks and sought assistance, as the users reached a position to explore a subtopic within their task. The agents were able to identify the subtopics by passing the user's keywords and its resources to the master agent, which uses its master structure and maps them to a commonly used string associated with same or similar resources.

Table 2. Agent requests and responses received

Agent	Task	Responses from Agents received during		
		Request 1	Request 2	Request 3
A1	T3	A5, A6,A7	A5, A6	-
A2	T3	A5, A6, A7	A5, A6	-
A3	T4	A7, A8	A7, A8	A7
A4	T4	A7, A8	A7, A8	0
A5	T1	A1, A2, A3	A1, A2	-
A6	T1	A1, A2, A3	A2	-
A7	T2	A3, A4	A3,	0
A8	T2	A3, A4	A3, A4	A3

Table 2 summarizes the requests made by the agents and the offers received for their task structures. A maximum of three requests were made by the agents to other agents during the experiment. For instance, when T3 was executed, the agent A1 upon encountering the second distinct keyword, sought assistance from others. At that instance, A1's task structure for T3 was consisting of one keyword and three associated resources. The assistance was received from agents A5, A6, A7. This was possible, because the task of A7 was in the same domain, but the complexity of the task was higher than A5 or A6, and the user corresponding to A7 had ventured into many sub-topics. A new structure was constructed by A1 with the inputs from A5, A6, and A7. However, when A1 sought assistance for the second time, as the user ventured into third distinct keyword, the responses were received from A5 and A6 only. This was because the new structure in A1, made the structure in A7 irrelevant. Also, an agent will respond only if it has additional keywords or resources that are not present in the requesting agent's task structure, though it may be qualifying with high relevance. Because of these, during subsequent requests made by an agent, as the user progresses in his task, the number of responses received becomes lesser and lesser. This pattern was noticed in the input request of all agents.

Table 3 summarizes the task structures constructed by the agents at the end of round 2. As it can be observed, there is a reduction in the number of resources used by the users, for performing their respective WKG tasks. The agents also suggest the keywords to the users, to explore further, when required.

Table 3. Summary of Task Structures in Round II

Agent	Task	Complexity	No. of Keywords	No. of Resources
A1	T3	Medium	3	6
A2	T3	Medium	3	6
A3	T4	High	4	14
A4	T4	High	4	11
A5	T1	Medium	2	5
A6	T1	Medium	2	6
A7	T2	High	4	11
A8	T2	High	4	13

6. CONCLUSION

In this paper, we address the challenge of assisting users involved in Web-based Knowledge Gathering (WKG) tasks, by proposing an environment consisting of intelligent agents that build contextual structures based on the observable behavior of the users involved in such tasks. The contextual information is the only source of implicit information that could be gathered from the user, and we utilized it for describing the contextual background of a WKG task. We chose the semantic link networks for representing the contextual information, as it gives a mechanism for organizing and structuring it, and also the BDI-style agents that tracked the contextual information were able to directly use them for communication, learning and sharing in a cooperative environment. It also gave the ability for the agents to understand the intricacies of the task through the user's actions.

The agents were modeled to share their contextual task structures, and seek assistance from other agents in a cooperative environment that might have knowledge of same or similar kind of tasks. The process of determining the task similarity was achieved by finding the contextual relevancy between two given contextual task structures. Also the new information gathered from other agent's similar task structures was used to build a new task structure that could be used at appropriate instances for guiding and enriching the knowledge of the user. The assistance was provided by the user's agent in the form of useful resources for that particular task, and keyword suggestions for better query formulation.

The user evaluation of the system, carried out through various experiments was able to assist users in achieving their goals effectively. Though our experiments were related to a small number of tasks, on a large-scale, the system would be able to assist many users involved in WKG tasks with wide-ranging contextual structures, and the application of socio-contextual filters [18] would allow us to narrow down on a few similar tasks from a large set of tasks for providing assistance. This kind of intelligent agent-based technique for assistance coupled with a supportive environment for WKG will provide a better platform for the users to carry out knowledge-intensive tasks on the Web using today's search engines.

7. REFERENCES

[1] Birukov, E., Blanzieri. E., Giorgini P. 2005. Implicit: An agent-based recommendation system for web search. In Proceedings of the 4th International Conference on Autonomous Agents and Multi-Agent Systems, 2005, pp.618-624.

[2] Broder, A. 2002. A Taxonomy of Web search. SIGIR Forum, Vol. 36, 2002, pp. 3-10.

[3] Chen, L., Sycara, K. P. 1998. WebMate: A Personal Agent for Browsing and Searching. In Proceedings of the Second International Conference on Autonomous Agents, 1998, pp.132-139.

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

[5] Google AJAX Search API http://code.google.com/apis/ajaxsearch/

[6] Hoeber, O. 2008.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 (WIIAT '08), vol. 3, 2008, pp. 29-32.

[7] Hoeber, O., Yang X. D., 2008. Evaluating WordBars in exploratory Web search scenarios**,** Information Processing and Management: an International Journal. Volume 44 , Issue 2 (March 2008) Pages 485-510, 2008 ISSN:0306-4573.

[8] Ingwersen, P., Belkin, N.2004. Information retrieval in context - IRiX. SIGIR Forum, 38(2), 2004

[9] Kellar, M., Watters, C., Shepherd, M. 2006. A Goalbased Classification of Web Information Tasks. In Proceedings of the Annual Meeting of the American Society for Information Science and Technology (ASIS&T 2006), Austin, TX.

[10] Kim, H. R., Chan, P. K.2003. Learning implicit user interest hierarchy for context in personalization. In Proceedings of the 8th international conference on Intelligent user interfaces (ACM, New York, NY, USA, 2003) pp. 101–108.

[11] Kobayashi, M., Koichi, T.2000. Information retrieval on the web ACM Computing Surveys (CSUR), Volume 32 , Issue 2. June 2000

[12] Lee, T. B., Hendler, J., Lassila, O. 2001. The Semantic Web. Scientific American, May 2001.

[13] Li, H.F., Lee,S.Y., Shan, M.K.2004. On mining webclick streams for path traversal patterns. In Proceedings of the 13th international World Wide Web conference(New York, USA, 2004) pp. 404–405.

[14] Lingling, Z., Li, J., Jingui ,P. 1999. An intelligent interface agent for Web-based information retrieval. In Proceedings of IEEE Workshop on Internet Applications, Aug 1999, pp.38-43.

[15] Marchioni, G. (2006) Exploratory search: from finding to understanding. Communications of the ACM, 49, 4, 41-46.

[16] Pirolli, P., Card, S. K. 1999.Information Foraging Psychological Review, Vol. 106, No. 4, 1999, pp: 643-675.

[17] Rajendran, B. 2008. 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.

[18] Rajendran, B., Iyakutti, K.2009. Socio-contextual Model of Knowledge Sharing among User Assisting Agents. In Proceedings of the IEEE ISI 2009 international workshop on Social Computing (SOCO), 2009.

[19] Rajendran, B., Iyakutti, K.2009. Cognitive Agents for Understanding the Complexities Involved in Web-based Knowledge-gathering Tasks", Proceedings of the IEEE System Man, Cybernetics Conference, SMC 09, 2009

[20] Rao, A.S., and Georgeff, M. 1995. BDI Agents: From Theory to Practice. In Proceedings of the first international conference on Multi-agent systems, ICMAS-95.(San Francisco, CA, 1995) pp.312-319

[21] Rose, D.E., Levinson, D.2004 .Understanding user goals in Web search. Proceedings of the WWW 2004, 13-19.

[22] Smith, R. G.1980. The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers, vol. 29, no. 12, 1980, pp. 1104-1113

[23] Tian, X., Du, X., Hu, H., Li, H.2007.Modeling Users Cognitive Structure in Contextual Information Retrieval. Fuzzy Systems and Knowledge Discovery, In Proceedings of Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007) Vol.4, 2007, pp. 349-353.

[24] White, R.W., Kules, B., Drucker, S.M., Schraefel, M.C. 2006. Supporting exploratory search. Communications of the ACM, 49, 4, 37-39.

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