Using Agent technology to access the Semantic Web

Sanjaya Ratnayake, Ruvindee Rupasinghe, Anuruddha Ranatunga, Shalinda Adikari,
Sajayahan de Zoysa, Kamala Tennakoon, and Asoka Karunananda
Faculty of Information Technology, University of Moratuwa, Katubedda, Sri Lanka
Email:sanjurat@yahoo.com

Abstract- **Semantic web consists of heterogeneous sources of knowledge including texts, graphics, blogs, animations, audios, and videos. However, there have been limited researches conducted to present semantic web information in the form suitable to individuals. This paper reports on the design and implementation of 'Divon', a swarm of agents that emulate a user profile driven approach to present semantic web information in the forms suitable to individuals. Divon has been structured with four agents, namely, Message Agent, Query Handler Agent, Presentation Agent, and User Profiler Agent. The Message Agent plays a key role in guiding the search process and displaying the information in a suitable form for the user whereas User Profiler Agent creates individual user profiles according to individual preferences. Divon has been developed on JADE environment and can run on any computer in connection with an arbitrary search engine.**

1. Introduction

Internet has been founded by the American authorities in the early 90's as an infrastructure for the defense and scientific work [3]. However, after identifying its immense potential as a facilitator for a myriad of human activities it has grown on a massive scale. An essential result of this unprecedented growth is that the amount of information residing in the web has also grown to a huge amount. Nowadays, Internet has grown into a multifaceted source of knowledge and postulated what is known as the Semantic Web [13]. Within this complex web environment, it is not easy for the web users to locate a particular piece of information they need. The traditional solution for this problem was that search engines came into operation. However, the results returned by these search engines were too general and did not match the exact requirement of the user. Hence, the need for personalized web searching mechanisms became apparent. Many researchers have proposed Agent based solutions for information retrieval [9].

User modeling techniques together with machine learning have been generally used to build Intelligent Agents [5]. Starting from a small knowledge base, Agent enhances its knowledge base through the machine learning techniques. Furthermore, there are also some agents specially designed for the purpose of information retrieval [9]. In addition, swarm of agents or multi agent systems have been experimented with to facilitate information retrieval on the semantic web [2].

However, all these approaches have a serious limitation when it comes to the user modeling. That is the agent's knowledge about the user is mostly static. The reason for this is that the knowledge parameters which are considered by the agent remain static over time. As a result, the user profile available to the agent does not reflect the actual user requirement at a particular time. More importantly, none of these approaches have handled the need for presenting the retrieved information in a format suitable to respective individuals.

In responding to these issues, we have developed a swarm of agents, Divon, that can dynamically modify the user requirements and guide the searching and presentation of information accordingly. Divon is designed to be run totally at the client end and hence would not add any burden to the network traffic. Basically it would maintain a dynamic user profile. User's current context would be identified by the attributes like location, time, and searching habits etc. Divon has been developed on the Java Agent Development Environment (JADE), and can be run on any client computer.

The rest of the paper has been organized as follows. Section 2 briefly describes the current approaches to information retrieval on the Internet. The swarm intelligence based approach is described in section 3. Section 4 describes the design of the Divon. Implementation details of the Divon are described in section 5. Work flow order of the Divon is described in section 6. Finally, section 7 contains the conclusion.

2. Current Approaches To Information Retrieval

Over the last many decades, researchers have introduced various approaches to information retrieval on the Internet and the semantic web. Undoubtedly, the Search Engines are the most known and the very first technology for information retrieval on the Internet. Subsequently,

Meta search engines [1], Distributed information retrieval systems [11], Agent-based information retrieval [20] have also been introduced by various researchers.

2.1 Search engines

Search engine is a web site that collects and organizes content from all over the Internet [17]-[19]. Those who wish to locate something would enter a query about what they would like to find and the engine provides links to content that matches what they want [15]. Among these, Web search engines are used to retrieve information from World Wide Web (WWW). Using a set of algorithms, search engines built different methods to find information that are required by users [14]. For instance, S. Asadi and H.R. Jamali (2004) have analyzed the shift in search engine development and identified eight aspects, namely, general aspects of web search, query formulation, search process, reference feedback, ranking, and retrieval of models, information filtering, clustering, and selection process [18]. According to this study, during the period 2003-2004, the general aspects of web search have been reduced, whereas information filtering and search process have been increased. This shows that users are more interested in information retrieval in a more personalized manner. In line with these requirements, in today's context, even the general purpose search engines such as Yahoo and Google have improved with basic natural language processing abilities.

Meta search engine operate based on the virtual databases. However, it does not compile a physical database of the web. As an alternative first they take a user request and pass it to several other various databases and then compile the results in a standardized manner based on a specific algorithm [7]. All Search engines have been primarily implemented as traditional software systems.

Distributed information retrieval techniques have been used to address the problem of long duration of the update intervals. The conventional architectures are centralized, nevertheless these techniques are capable of shortening the update intervals [22]. Distributed information retrieval systems can be created with the contribution of multiple local Meta search engines that corporate with each other [16].

2.2 Semantic Web approach

Another way of distributed information retrieval is the semantic web method. Semantic web provides the means to use metadata that help determine which documents are relevant. Simply semantic web means data with meaning. Semantic web is a technology which describes things in a way that computer's applications can understand [6]. Unlike normal web search engines semantic web does not consider links between pages; it is based on the relationship between things and properties of things. Semantic web approach considers not only the total amount of data but also the differences among the local metadata vocabularies [21].

The Semantic web is governed by a three step framework, where the first step is *resource selection*, in which for any given query the full Web has to be queried. The second step is *query reformulation and ontology alignment*, which deals with the differences in the vocabularies used by the user and the selected information resources. *Aggregation and data* fusion is the third step, which integrates the ranked results from the individual resources into a single list. The three step framework of semantic web has generally been implemented as a piece of traditional software that does not use any intelligence for information retrieval.

2.3 Agent-based information retrieval

There are verities of research projects that have adapted the agent technology for information retrieval [12]–[4]–[20]. It is quite natural to use agent technology for information retrieval, since agents are autonomous and can work to a large extent without user intervention all the time. These agent solutions work as a front end for standard search engines. Some of these agents also enable personalized searching on the Internet [10]. However, pre-structured agents with static user profile may restrict the effective, personalized searching on the Internet. In particular, effective agent must be sensitive not only to user profile, but also to the capability and the choice of search engines, resources on the semantic web, presentation format of resources, etc. Obviously, all these aspects are too much to handle by a predefined single agent.

3. Approach to Using swarm of agents

As we discussed, the existing approaches to information retrieval have been primarily based on technologies that improve performance of search engine. Further, agent-based approach to information retrieval has also faced with serious limitations as per personalizing the search process through the dimensions of user profile, resources, location, nature of the Internet connectivity, time of the day, etc. whereas the agent technology would be one of the most effective approaches to information retrieval; it is too ambitious for a

single agent to assign all the tasks related to search process.

Therefore, we propose an approach to effective searching on the semantic web, which composes a group of agents (swarm of agents). Using this approach, the communication among agents produces the best result and allows the evolution of the performance of the swarm of agents. More importantly, this approach ensures effective information retrieval through the collaboration of four agents, namely, Message Agent, Query Handler Agent, User Profiler Agent, and Presentation Agent. The overall approach comes out as intelligent software, called Divon, which can run on a client machine.

Key features offered by Divon are autonomous, personalized, reactive, adoptable, proactive, and collaborative and facilitate effective information retrieval on the Semantic web. Divon is necessarily autonomous as it runs all the time with minimum user intervention. In particular, agents in the swarm communicate without bothering the user, yet come up with solutions through their interaction. Altogether the system would carry the following main benefits for the end user.

1. Highly personalized content
2. Sensitiveness to the user's current context
3. Adaptability to the changes in information requirements of the user
4. Proactive search on behalf of the user
5. Reduced search time
6. Enhanced quality of the search results

High personalized content is supported by the User Profile Agent. Divon identifies user by the windows authentication and develops the profile based on criteria such as location, time, search query, viewed sites, time spend on particular sites, and repeat views. Based on the profile, user is categorized in to the relevant domain by the User Profiler Agent. By default Divon consists of ten domains, technology, entertainment, business, news, information, online transaction, education, research, children, and general. Divon is capable of adding new domains while killing isolated domains.

Divon is reactive to be sensitive to the user's current context. As such, Divon is capable of representing semantic web in the form of blogs, news, videos, books, web, local sources, images, and general. As Divon is continuously updating user profile, the system is reactive to the changes in information requirements of the user. Due to its reactive nature, although the Divon is autonomous, the user can access the Divon at any time.

Adaptable nature of Divon enables us to install Divon for any client machine that uses an arbitrary search engine. Divon is also adaptable of incorporating new requirements, the changes of previous interests, etc., of a user into the system.

Proactive search on behalf of the user takes place when the user logs in to the system. The user can view the previously visited, preferred sites before he starts a search. Divon reduces the searching time at the user level by proactive search, query optimization, result filtering and ranking. At the process level, search is improved by caching and accessing local data sources. Divon enhances the quality of search results by initially accessing local data sources. If the search is done outside local sources, it still filters results based on the user profile. When swarm is not busy it analyzes the user details and patterns. So with the same time the user will be classified into several domains and data representation styles.

Divon produces results through the collaboration of four Agents. The collaboration has addressed the problem of not being able to handle the entire workload of effective searching by a single Agent. This is the key contribution in our research work.

4. Design of Divon

Fundamentally, Divon can be seen as a swarm of agents that deal with messaging, request handling, and resource allocation. Fig. 1 shows the functional overview among these processors. In order to implement the concept, we have defined Query Handler Agent that has access to semantic web and the repository of URLs which are collected during execution. The Message Agent is responsible for overall communication among the user and the agents in the system. The User Profiler Agent plays a central role by communicating with all other agents whereas Presentation Agent is responsible for dynamic data representation.

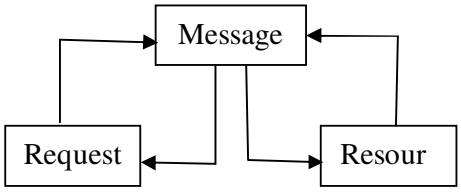

Fig. 1. Functional overview-Divon deals with accepting requests displaying messages and resource/agent allocation to perform the request

The top level architecture of the Divon is shown in Fig. 2. It comprises four agents, Local repository, and the connection to the Semantic Web. Next we shall briefly discuss the role of four agents and Local Repository in Divon.

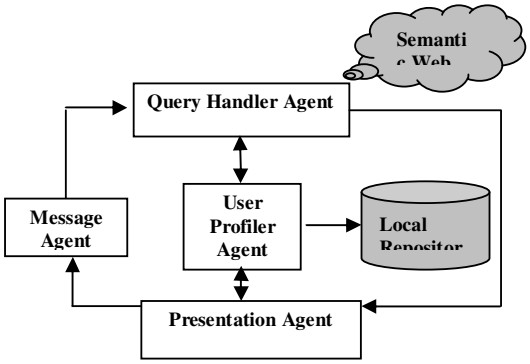

Fig. 2. High level modular architecture of Divon – Four Agents with Local repository.

4.1 Message Agent

Message Agent is responsible for handling the messages over the system. This works as the communicator between all four agents of the system and the user. Blocking the messages, prioritizing them, storing the messages, and invoking the right agent at the right time are major functions of the agent. At the search request, the Message Agent checks the status of the system agents. If the agents are in process Message Agent will store the data in its local data sources and later utilize swarm of agents when they are free. Message Agent is also capable of providing advanced warnings. For example if the user tries to do a bulky download, first the Message Agent analyzes the network traffic and if the traffic is high it informs the user about the alternatives available.

4.2 User Profiler Agent

User profiler Agent handles the user specific data for each individual users of Divon. The complex architecture of the agent monitors the complex behaviors of each user and identifies their key features. Based on those features users are categorized into different domains by the User Profiler Agent. Each domain consists of domain specific keywords and those are matched with the search query. In keyword identification the agent works with a complex algorithm which maps the parameters such as, the web sites each user visits, the number of clicks performed and the time spent on them. The keywords identified by the agent are more personalized to the user and are able to map the user behavior in an appropriate manner. More importantly the changes of the user behavior can be identified by the User Profiler Agent and the user's profile can be customized accordingly. When the user is not using the machine over a period more than six months , the User Profiler deletes the profile of the specific user. Once the agent identifies the behaviors of the user, the agent itself searches through the web to retrieve the related URLs for the user automatically. Every improvement done to the users are stored in the repository.

TABLE I
KEY WORDS WITH SIMILAR MEANING

Key Word	Meaning	Domain
SOAP	Protocol	Technology
	Cleaning Utility	General
MOUSE	Animal	Animal
	Computer peripheral	Computer
OFFICE	Working Environment	Job
	MS Office Software	Computer
BOW	To bend forward at the waist in respect	Culture
	Front of the ship	Traveling
	Weapon which shoots arrows	History

4.3 Query Handler Agent

Once the Message Agent sends the query to the Query Handler Agent, it tries to personalize the query according to the feedback given by the User Profiler Agent. The user's domain has a huge importance here as different users can use the same word to interpret different meanings. Table I depicts such words and the domains that they can be categorized.

Thus the keywords given by the User Profiler are prioritized by the Query Handler and mapped with the given query to create a meaningful implementation to it. More importantly with the experiences gathered by the Query Handler Agent, it can improve its ability of creating the suitable query. Query optimization is also done within the Query Handler Agent to make it more specific to search the given query. The optimized query is sent to the search engine by the Query Handler Agent in order to retrieve the search results. The Query Handler Agent communicates with the Presentation Agent for further processing of the query.

4.4 Presentation Agent

Presentation Agent is responsible for representing data in an understandable manner to the user. More importantly the Presentation Agent can analyze, filter, sort, and rank the search results according to the user preferences. In case of a scenario where the given results do not match the user profile, the Presentation Agent communicates with the Query Handler Agent through the Message Agent and requests Query handler to process again. So this goes in a cyclic manner until the best search results are retrieved. The direct communication between the User Profiler Agent makes the Presentation Agent aware of the user preferences, so that the Presentation Agent can present the results in a user preferred format, for example in pictures, audios, videos, blogs, images, news, books etc.

4.5 Local Repository

Local repository stores all the information transmitted through the system. The repository is updated with the feedback of the User Profiler Agent. So in case of a failure the local repository can be used to recover the system from the failed state.

5. Divon Implementation

Divon has been implemented using JADE, which provides the critical features of agent oriented systems such as a distributed, fault tolerant, highly secured, and semantic framework [8]. The Google search engine has been used as the search engine for development purposes. The middleware technologies of the agents such as agent management, agent communication, and agent software interaction have been implemented according to the specifications of The Foundation for Intelligent, Physical Agents (FIPA) [8]. The basic functionalities of the agents such as birth, registration, location, communication, migration, and operation of agents were implemented according to the FIPA specifications in agent management.

The main function of Divon, which is the agent communication, was done using Agent Communication Language (ACL). The communication among agents is an asynchronous message communication and being autonomous and proactive are the key features of the agent. It has its own thread of execution and has the knowledge to be invoked at the right time and to perform the correct action.

The actual tasks that the agents should perform are defined within the behaviors of the agents. Therefore each agent who has been described above have their own behaviors. The behaviors can be invoked concurrently. Table II describes the dedicated behaviors of each and every agent.

TABLE II
THE BEHAVIORS OF AGENTS

Agent	Behaviors
User Profiler	createDomain , deleteDomain, addKeywords, matchUser, createUser, getDomain, getKeyword, createUser, dropUser
Query Handler	getQuery, optimize, connectSemanticWeb
Presentation	getPreferences, presentation, searchResults, sortResults, rankResults, checkValidity, askQueryHandler
Message	display, checkBuzyUsers

The Message Agent implements cyclic behaviors in order to achieve its main feature, which is the continuous checking for resources. Here the agent communication is done in an asynchronous manner in which all the messages passed over the system are stored in a queue and for selecting the messages from the message queue, the Message Agent has to check the status of the other agents periodically. Fig. 3. depicts the state and transmission diagram of Message agent and its communication among other agents.

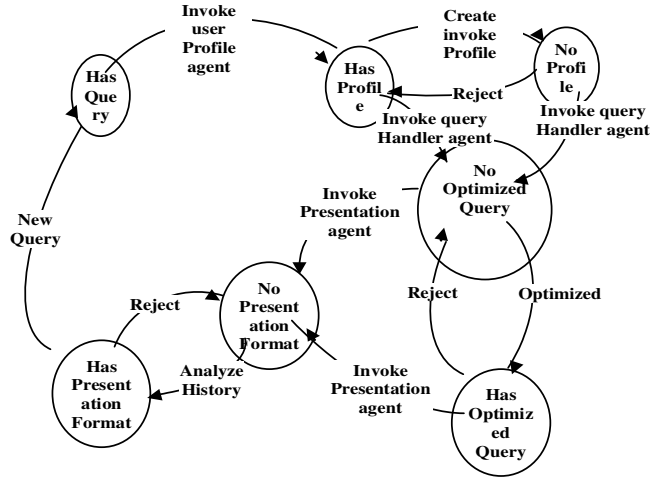

Fig. 3. state and transition diagram of the Message Agent

The dynamic data representation is handled by the Presentation Agent using its presentation behavior. Presentation Agent will communicate with the User Profiler Agent and negotiate about the user preferred data format. Once the User Profiler Agent responds, the Presentation Agent starts its process accordingly. Meanwhile checkValidity behavior checks the relevance of the search results to the user profile and if there is any mismatch Presentation Agent's askQueryHandler behavior communicates with the Query Handler Agent through Message Agent.

User Profiler agent is able to categorize users based on his location. Here it considers the user's location specific requirements for a more accurate searching strategy. createDomain behavior in User Profiler Agent is able to generate domains for specific users. The keywords relevant to those domains are categorized and listed by addKeywords behavior. Here createUser behavior creates new users whereas dropUser deletes frequently not logged in users.

Query Handler Agent is able to expand and optimize the search query with specific keywords using its expand behavior. The data retrieval from the semantic web is performed by the

connectSemanticWeb behaviors. The query optimization is based on the combination of different searching strategies, namely; click history based personalization, ontology based personalization, personalization based on long term search history and Group Based Personalization.

6. Divon Functionalities

When a user logs in, Divon presents the user with the result of the most recent search session. This information and preferred sites are filtered by the Message Agent based on user's domain and recent searches. For example, a researcher who searched on "Agent Theory" recently will be getting some agent theory related links at the time of logging. The links consist of sites that a researcher has not visited previously but related to the "Agent Theory".

At the time of browsing the user may also experience the effect of Divon. For example, lets assume a house wife and a researcher have explored the search queries in Table III. These are based on past searches and now stored in the Local Repository.

TABLE III
QUERIES BY DIFFERENT USERS

House Wife	Researcher
Washing machine powder (images)	Protocols(tutorials)
Sun Light (images)	Computer (tutorials)
How to clean clothes(video)	XML (tutorial)
Laundries in Texas(images)	Remote Procedure (thesis)

Now according to the information available, Divon will classify the researcher who prefers to search computer related tutorials into "Computer" category and house wife, who is interested in images into "General" category. Let us assume that both make a search on "SOAP". At this point, Divon generates different results for the house wife and researcher, because they belong to different categories. Furthermore the presentation criterion also changes according to the user interest, by providing the researcher more tutorials and the house wife more images. Table IV shows the output generated by Divon for the above queries. The researcher will be getting the results from tutorials and thesis and house wife will be getting the results from general web or as images.

TABLE IV
ANSWERS BY DIVON

House Wife	Researcher
Washing Soap(images)	Simple Object Access Protocol (SOAP) 1.1 (tutorial)
Effect of washing hands with soap(images)	SOAP - Wikipedia, the free encyclopedia (tutorial)
SOAP POWER(images)	SOAP Tutorial
Getting Clothes Clean(video)	Simple Object Access Protocol – SOAP (thesis)

Let's consider another situation where the user retrieves different search queries according to the user domains they belong. Assume two users of the Divon who belong to computer and animal domains respectively are searching for the term "mouse". If they use a normal search strategy the first three search results they would obtained are listed in table V.

TABLE V
QUERIES BY DIFFERENT USERS IN A GENERAL SEARCH

Index	Normal serach engine
1	Mouse (computing) - Wikipedia, the free encyclopedia en.wikipedia.org
2	Apple – Mighty Mouse www.apple.com/mightymouse
3	What is mouse? - a definition from Whatis.com Searchexchange.techtarget.com

Nevertheless when they use the personalized strategy they can directly go for their desired solution quickly. As depicted in table VI, through Divon, users interested in computers get computer related information while users interested in animals can view animal related data. Therefore the personalized search could provide users relevant search results with a high precedence.

TABLE VI
QUERIES RETRIEVED BY DIFFERENT DOMAIN USERS

Index	For a computer domain user	For an animal domain user
1	Mouse (computing) - Wikipedia, the free encyclopedia en.wikipedia.org/wiki/Computer_mouse	Pet Mouse, Fancy Mice, Pet Mice, Mus musculus animal-world.com/encyclo
2	What's Your Favorite Computer Mouse? www.youtube.com	Jungle Mouse www.junglemouse.ne
3	What is mouse? - A Word Definition From the Webopedia Computer www.webopedia.com/TERM	Biomethodology of the Mouse - Animal Research - The University of Iowa Research.uiowa.edu/animal

7. Conclusion

Since the number of internet users is growing rapidly, the interests of internet users vary rapidly, making it hard to find relevant information according to individual user interests. Therefore personalized search engines, agent based search engines, and semantic web approaches have come into existence. Nevertheless those approaches have a serious limitation in user modeling and data presentation due to static user modeling methodologies.

Agent based systems are the newest tools which replace other web searching technologies. In this paper we have presented Divon, a swarm of agents who learn the user behaviors by observing the users searching habits and activities, and to present the

search results in a way appropriate to individuals. Our solution is implemented one step ahead of the existing agent based designs as it has the ability to search results from the semantic web depending on the users' profile. Divon handles requests, responses, and resources in a more meaningful manner to achieve maximum resource utilization. The four agents, namely; Message Agent, User Profiler Agent, Query Handler Agent, and Presentation Agent collaboratively work together to provide highly accurate information to the user from the semantic web. User Profiler Agent who creates separate user profiles for every user communicates with Presentation Agent and Query Handler Agent to provide required user information and keywords to the personalized search. Query Handler Agent creates meaningful queries using the user entered queries and user specific keywords and retrieve search results from the semantic web. Message Agent communicates with the user, get their requirement and present him the results with his preferred format.

In the future we hope to expand the number of agents in our system to take more complex behaviors into consideration. In addition to the currently available user profiling criterions, we can consider users email information and messenger information. Furthermore we hope to integrate the system with the web browser so that we can monitor the users scrolling behavior also. We expect to generate more sophisticated learning, filtering, and ranking algorithms to further increase our system performances.

Acknowledgements

All the faculty members and our colleagues deserve to be appreciated for contributing to the success of this project in various ways. Also the authors of the references we have used throughout this project are highly appreciated.

References

[1] A. E. Howe, D. Dreilinger, "SAVVYSEARCH: A Metasearch Engine That Learns Which Search Engines to Query", *Jour:* AI Magazine, 1997.

[2] A. M Fard, M. Kahani, R. Ghaemi and H. Tabatabaee, "Multi-agent data fusion architecture for intelligent web information retrieval", *Proceedings of World Academy of Science*, Engineering and Technology, vol. 22, 2007, ISSN 1307-68.

[3] B. M. Leiner, V. G. Cerf, D. D. Clark , R. E. K. Leonard, K. Daniel, C. L. Jonathan, B. P. Lawrence, G. Roberts and S. Wolf,(1999), "A Brief History of the Internet", The Computing Research Repository.

[4] D. Bogdan, C. Ruth, M. Malcolm, C. Taylor , and M. Rusinkiewicz, "Distributed Processing of Queries for XML Documents in an Agent Based Information Retrieval System, "Kyoto International Conference on Digital Libraries, 2000.

[5] D. Riecken, Agents that reduce work and information overload, Communications of the ACM, Vol.37,1994, No 7

[6] Dr. B. Matthews, Semantic Web Technologies, CCLRC Rutherford Appleton Laboratory.

[7] E. J. Glover, S. Lawrence, W. P. Birmingham, and C. L. Giles, "Architecture of a Metasearch Engine that Supports User Information Needs," *Conference on Information and Knowledge Management*, Kansas City, Missouri, United States, 1999.

[8] F. Bellifemine, G. Caire, and D. Greenwood, "*developing multy agent systems with JADE*", Wiley series in Agent Technology, 2007, pp. 20-45.

[9] G. Graart, Agent based information retrieval supported by information markets, Master's thesis, University of Maastricht, 2003.

[10] Giovanni, S. Marco, D. Pasquale, L. U. Thiel , and M. L'Abbate, "A Personalized Information Search Process Based on Dialoguing Agents and User Profiling", *European Colloquium on IR Research – ECIR,* 2003.

[11] I. A. Macleod, P. Martin, B. Nordin, J. R.Phillips, "Strategies for building distributed information retrieval systems", Information Processing and Management, 1987.

[12] J. B. Odubiyi, D. J. Kocur, S. M. Weinstein, N. Wakim, S. Srivastava, C. Gokey, and J. Graham, "SAIRE - A Scalable Agent-Based Information Retrieval Engine", Autonomous

Agents & Multiagent Systems/International Conference on Autonomous Agents, 1997.

[13] J. Hendler, "Agents and the Semantic Web", IEEE Expert / IEEE Intelligent Systems, vol.16, March 2001.

[14] Massimo," Hyper Search Engines",journal: Computer Networks [and ISDN Systems].Marchiori," The Quest for Correct Information on the Web, 1997.

[15] M. Peshave, How search engines work and a web crawler application, Department of computer science, university of Illinois at Springfield Springfield, Midwest instructions and computing symposium, April 2005.

[16] N. E. Craswell, Methods for Distributed Information Retrieval, The Australian National University, May 2000.

[17] P. Thomas, An introduction to search engines, Search & Delivery; CSIRO ICT Centre, May. 2008

[18] S. Asadi and H. R. Jamali,(2004), "Shifts in Search Engine Development: A Review of Past, Present and Future Trends in Research on Search Engines, " *Webology*, vol. 1, December 2004.

[19] S. Brin and L. Page, The anatomy of a large scale hyper textual web search engine, Doctorial thesis, University of Stanford, 2001.

[20] S. Ugurlu , N. Erdogan, "An Agent-Based Information Retrieval System", Advances in Information Systems, 2000.

[21] T. Schreiter; An Introduction into Semantic Web Services; Hasso-Plattner-Institute for IT Systems Engineering, University of Potsdam, Germany.

[22] U. Straccia, R. Troncy, towards distributed information retrieval in the Semantic Web: Query reformulation using the oMAP Framework, *3rd European Semantic Web Conference*, June 2006.