SEA: SEARCH ENGINE AGENTS

Constantin B. ZAMFIRESCU[*] [**], Marius STAICU[*], Mihai LUCA[*]

**AI Research Group*
2A Reconstrucţiei Street 2400 Sibiu, ROMANIA
Phone/ Fax: +40-69-224168 E-mail: trident@sibnet.ro
***"Lucian Blaga" University of Sibiu*
Dept. of Computer Science and Automatic Control, 4 Emil Cioran Sreet 2400 Sibiu, ROMANIA
Phone: +40-69-422424 Fax: +40-69-212716 E-mail: zbc@acm.org

Abstract: The Web is a disorganized place, and it is growing more disorganized every day. Even with the state-of-the-art indexing systems, web catalogs, and soft-bots, World-Wide Web users are finding it increasingly difficult to gather information relevant to their interests without considerable and often fruitless searching. Following an agent-oriented approach, the research in this paper aims at addressing such circumstances in a more comprehensive framework, able to extend our results to other interrelated challenges. Usually settled in the **information retrieval** (IR) research field, as a continuously confrontation with today's globalisation trends, some relevant issues together with some effective methods to address them are identified and discussed. How our application will tackle them, is the main question that this paper will tring to answer.

Keywords: agent-orientation, adaptivity, user-friendliness, information management, search engine, WWW.

1. Introduction

When, at the begining of 1950's, Calvin Moers, one of the information science pioneers, coined the term **information retrieval** (IR) he also defined the problems addressed by the activity: (1) *How to represent and organize information intellectually?* (2) *How to specify a search intellectualy?* and (3) *What systems and techniques to use for those processes?*[1]. The problem underlying all of theoretical, experimental, and empirical activities in user modeling revolves around the classic and most dofficult question [2]: What it is important to know about the user in order to support the user in interaction with the IR system? Accordingly to Maglio and Barret [3], developing an explicite model of a user's information need addresses the folowing issues:

1) What kind of support should this model give?

a) Improving precision (the system can add other terms in the query from the user to cover the context of its meaning). This can improve the percentage of relevant retrieved documents.

b) Improving information need coverage. The concepts conveyed by a user query express a vague information need. Expanding this concepts will make it more likely that every aspects of this information need is captured.

c) Pointing the user to relevant information. The system may expand and search for these expansions autonomously.

2) What aspects of an information need should be represented? A distinction can be made between *topics of interest* and *situational factors*. The first term refers to the concepts which are part of the information need. The latter provides a context for a specific information need, for instance, the type of knowledge requested or the background knowledge of the user.

3) How to infer aspects of an information need? If not provided directly by the user, these aspects should be estimated from other sources. Some considered sources of information are:

a) Document read by the user. The system can monitor the user's browsing through the World-Wide Web. Acceptance of a document by the user can be measured explicitely, by an option in the interface, or implicitely by measuring reading time.

b) Clicking behaviour. This can be used, for instance, to estimate the user's browsing strategy or reading capacity.

4) How to deal with the ambiguity of the actions of the user in regard to the estimation of the information need? Because of this ambiguity, the system should

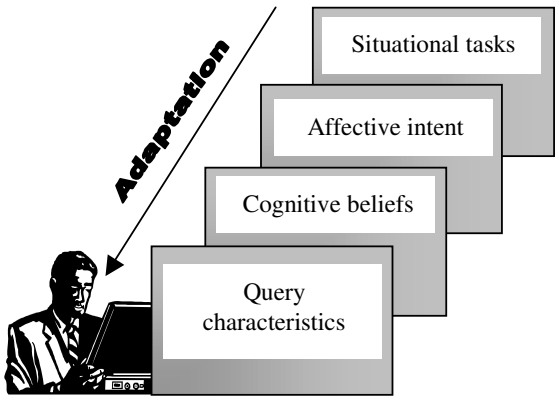

Figure1. IR interaction

have a way to deal with conflicting hypotheses. As possible formalism, fuzzy logic and Bayesian networks can be considered.

Consequently, to undertake all kinds of such questions our implementation (SEA) should support the following issues: (1) assist the user in the diagnosis process and question reformulation; (2) select appropriate search engine for efficient searching accordingly to their profiles; (3) translate the question into one or more queries and search strategies acceptable to the given search engine; (4) manage searching strategy; (5) support the user in the results assessment; (6) provide the user with the appropriate outputs in a suitable structure; and, way not, (7) advice he or she in the follow-up activity.

The remainder of this paper is organised as follows. Section 2 summarises the rationale of SEA architecture. The third section will present some detailed implementation issues regarding SEA agent-oriented approach. Some related works in the IR field are given in the section 4. Finally, our remarks and future path will establish the framework that will be here to stay, if not in intention, at least in today's manner of dealing with information tension.

2. Architectural issues

Over the last few years, there has been increasing interest in intelligent agents, distributed artificial intelligence and distributed systems. Links this with the increasing focus on IR systems and co-operative work patterns, raises the issues of how these "distributed cognition" [4] capabilities can be integrated to create intelligent tools.

The MAS paradigm represents one of the most promising approaches to build complex and flexible new architectures required for next generation of intelligent tools offering a new dimension for large-scale integration. MAS are software systems composed of several autonomous software agents running in a distributed environment. Beside the local goals of each agent, global objectives are established committing all or some group of agents to their completion. Some advantage of this approaches are: it is a natural way for controlling the complexity of large, highly distributed systems; it allows the construction of scalable systems since the addition of more agents is a easy task; MAS are potentially more robust and fault-tolerant than centralised systems.

An important role for agents may be the delegation of tasks. Agents interact and negotiate with each other to determine a suitable contracting agent. The contract net model [5] provides a suitable general protocol to design and implement this negotiation process.

The MAS provides a platform for co-ordination and co-operation, within which its agents can work collectively to solve specific problems. Clusters or teams of agents are identified [6] to perform specific reasoning for a given task and decision-making responsibilities are delegated to co-ordination groups made up of these agents. After all, if we don't expect people to be omnipotent, why should we expect this from agents?

A more detailed description of the agentcy features are given in [7, 8] and in a related paper A3CKM: Anthropocentrical Agent Architecture for Complex Knowledge Management.

2.1. User modelling

Accordingly to (Saracevic, Spink and Wu, 1997), approaches to user modelling in IR can be divided in two main category: *system-centered* and *human-centered*. While the first put emphasis on *relevance feedback* (users are modeled through texts or clusters of texts) and *query expansion* (the initial or modified query is used as a basis for user modeling), the second take into account *question shape* (user modeling is accomplished through various interview and analysis techiques) (Harter, 1992; Redford, 1996) and user's *cognitive aspects* (Allen, 1996). Another method is to build into the system ways and means by which users can on their own model articulate their problem with the system's assistance.

On the user side we can model cognitive, affective and situational levels. Saracevic et al [17] suggest that user modeling is an interactive process that proceeds in a dynamic way at different levels trying to capture user's *cognitive*, *situational*, *affective* and possible other elements that bear upon efectiveness of retrieval (See Figure 1).

In such a framework the request must be scrutinized through all its related steps: from request formulation to answer acceptance. So, in a searching process can be delineated three key stages: *request formulation*, *retrieved pages selection* and *locating the intended information.*

2.1.1. Profile enhancement

Keywords occurring in a particular searching process (e.g. source description, contextual links) will be clustered using a similarity matrix for the keywords stored in the user profile, very similar with the approach

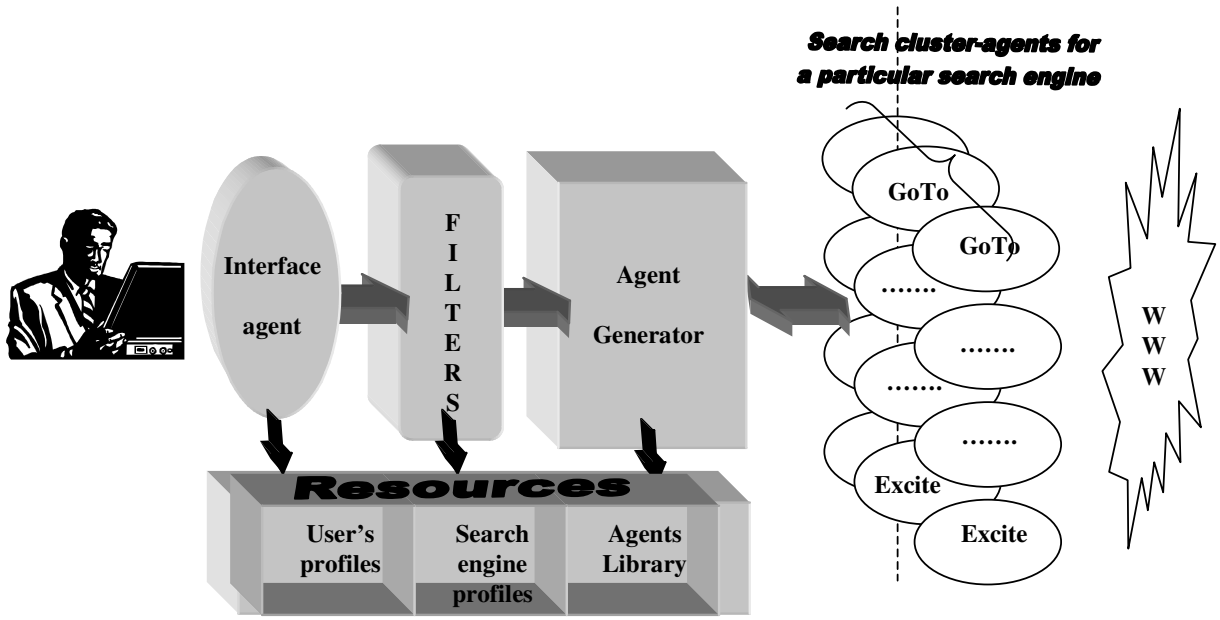

Figure 2. SEA overall architecture

followed by Davis, Weeks and Revett [9] in their Jasper implementation. Contrasting with them, we look at the search process as a whole, instead of the pages stored in the ultimate part of the user's request. We capture the user's choice, the rational behind each of them, the open questions related to the request, the assumption behind it, and any related supporting information. The matrix used, will give us a measure of the 'similarity' of keywords in the user's profile. For two keywords K_i and K_j, the Dice coefficient is given by:

$$2 \text{ X } | K_i \cap K_j | \ / \ | K_i | \ + \ | K_j |$$

Once the similarity matrix is calculated it is exploited in two ways: (1) *profile enhancement* (adding those keywords most similar to the keywords explicitly represented in the user's profile in similar way of query reformulation techniques) and (2) *proactive searching* (search proactively for new WWW pages relevant to user's interest). The algorithm is straightforward – given an initial starting keyword, find the *n*. Link this *n* to the original word and repeat the process for each *n* new words a number of *m* times.

If complete-link clustering is used [9], whereby the similarity between the least similar pair of items from two clusters is taken as the similarity between the clusters, the cluster dendogram is obtained. A similarity threshold can be set to provide the similarity degree between the clusters.

2.1.2. Adaptive recommendation

Accordingly to Balanovic [10] for the content-based approach, there are four essential requirements:

w – A representation of a Web page.

m – A representation of the user's interests.

p(w, m) – A function to determine the pertinence of a Web page given a user's interest

u(w, m, s) – A function returning an updated user profile given the user's feedback ***s*** on a page ***w***.

The assumption underlying content-based systems is that the content of a page is what establish the user's interest. Going on, we make the further assumption that we can represent the content of a page purely by considering the words contained in the text and also by its description.

Considering the vector-space model of IR [11] as a suitable mechanism for documents based representation, documents and queries are represented as vectors. This model has been used and studied extensively, forms that basis for commercial Web search systems and has been shown to be competitive with alternative IR methods [12].

In this model, we assume some dictionary vector ***d***, where each element d_i is a word. Each document then has a vector ***w***, where element w_i is the weight of a word d_i for that document. If the document does not contain d_i, then w_i=0.

As in Fab implementation [10], in our formulation we reduce words to their steams using the Porter algorithm [13]. That will ignore words from a standard stop list of 571 words, and calculate a TFIDF weight: the weight w_i of a word d_i in a document ***W*** is given by:

$$w_i = (0.5 + 0.5\, tf(i) / tf_{max}) (\log (n / df(i)))$$

where $tf(i)$ is the number of times word d_i appears in

document ***W*** (the term frequency), ***df(i)*** is the number of documents in the collection which contain d_i (the document frequency), ***n*** is the number of documents in the collection and tf_{max} is the maximum term frequency over all words ***W***.

To avoid over-frittering, accordingly to experiments described in [14], the optimum of used words is between 30 and 100. In our implementation, because the algorithm is used especially for recommendation based on the page's description and not for the content of the page itself, a range between 20 and 50 is sufficient. Once the top approximately 30 words have been picked we normalize ***w*** to be of unit length, to allow comparisons between documents of different lengths.

The vector representation is then used both for pages recommendation, and for the model of the user's interests. In order to measure how well a page ***w*** matches a profile ***m***, we use a variant of standard IR cosine measure:

$$p(w, m) = q(w) * m$$

where the function ***q(w)*** return the similarity measure accordingly to the profile described in the above section.

Updating ***m*** also corresponds to a normal operation in retrospective IR. We use a simple update rule:

$$u(w, m, s) = m + z(t)\, w$$

where ***z(t)*** is the user's implicit score for page ***w*** in the selecting process

2.2. Information retrieval system

The chief intent of HTML and HTTP is to assist user-level presentation and navigation of the Internet. Automated search or sophisticated knowledge gathering has been a much lower priority. Given this emphasis, relatively few mechanisms have been established to mark up documents with useful semantics information beyond document-oriented information like "abstract" or "table of contents". As a result, most common indexing mechanism and agents robots for the WWW have generally fallen into one of three categories: (1) text-indexing engines; (2) catalogs painstakingly built by hand; and (3) private robots using ad-hoc methods to gather limited semantic information about page.

Each approach has disadvantage. Text indices suffer because they associate the semantic meaning of web pages with actual lexical or syntactic content. Although text indices are improving, the amount of information on the Web is also growing rapidly. A major disadvantage of hand-build catalogs is the man-hours required to construct them. Given the size of the WWW, and the rate at which it is growing, cataloging even a modest procentage of web pages is a difficult task. Ad-hoc robots that attempt to gather semantic information from the web typically gather only the limited semantic information inferable from existing HTML tags. The current state of natural language processing technology

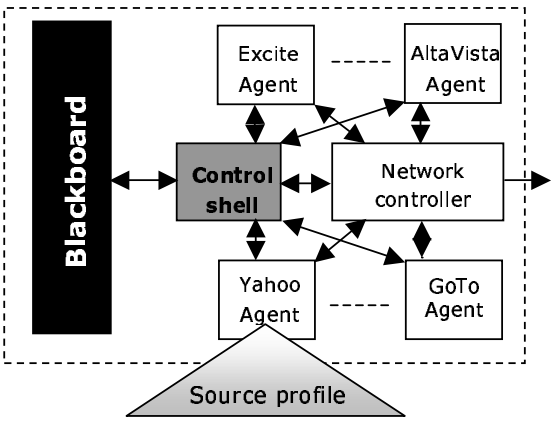

Figure 3. The blackboard architecture

makes it difficult to infer much semantic meaning from the body text itself at resonable rate.

In our implementation, the agent generator is based on the agent-based blackboard approaches (see Figure 3). As Vranej and Stanojevic [15] observes, the blackboard framework adopted here, is a promising choice for the co-ordination mechanism of multiple knowledge representations and reason techniques in multiparadigm system. It bridge the knowledge gap among those agents whose capabilities are restricted to a local area of expertise, facilitating the gathering of search-agents into collaborative groups or clusters and coordinating their decision-making. On the other hand, agent generator operate as a meta-level agent to increase cooperation between search-engine agents that are somehow replications of a general architecture for the search engine agents class (clone agents). Taken into account the resources profiles (serach engines), the clone agents reprezent the corresponding resource agent in concurrent IR activities.

2.3. SEA overall architecture

In the Figure 2, are represented the most important elements of the SEA architecture. As can be seen, to achieve its goals, it use significantly an agent-oriented approach.

Interface agent determines, constructs and maintains the user's profile (initially built on some relevant question relevant to the user interests). It also take the request from the user and sent it in a propper shape via filters module to agent generator who realize the interface with the information sources. Moreover, interface agent will continuously monitor the user actions and try to gather more dates about him/her (interest domains, frequently accessed pages, process in finding the required information) so that on a new demand to be able to personalize better the response (the order in which will present gathered information based on how frequently they are visited, new addresses who may be of some interest and so on).

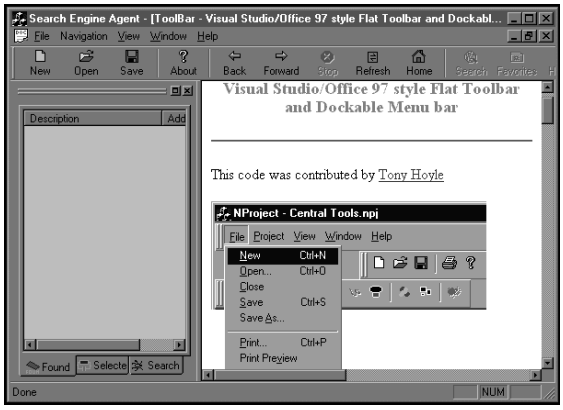

Figure 4. SEA Interface agent

Filters make an intelligent filtering (deleting duplicate links, local links, advertising link, dead links etc.) on the received data and try to adapt it accoordingly to the user 's profile and technological environment, respectively.

Agent generator module control the search agents activation. For each search engine it will launch a specific search agent who send the query and receive the answers from the particular search engine. These answers are filtered and found addresses are passed to the filtration module who accordingly to the user's profile will select the desired information.

For each available *search engine* implement there is a dedicated *agent* who know how to format the query and which are the possible answers that it will can obtain. The answers are bring to a canonical form and are send to the agent generator. It gather all the obtained response and pass them to the filtering module.

Resources are made up from "traditional" categories: data base, knowledge base, model base and agents library, providing the capability to create, update, store, recall, operate and control component units.

3. Implementation issues

At this time, the program is divided into two modules (one who realize the interface with the user and another who handle the data) [16]. These modules communicating through TCP/IP, allow an easy transformation, an independence on the graphical platform used (Windows, Xwindow) and make possible to exploit the search module from another program if this implements the established communication protocol.

The user interface module take over the demands and show the answers. This module is available in two forms : as a stand-alone program or as a Java applet which may be integrated in a Web browser (and providing WBI - Web Browser Intelligence). The demands are forwarded to the search module together with the addresses of the search engines on which the search must occurs. The user has the possibility to set the engines that ought to be used to retrieve the needed information. For each known engine in the search module there is a specialized agent. If it is necessary to add a new engine you need simply to clone a specific agent. Because the search module is written in Java it is easy to add new agents which will be loaded dynamically. In the GUI it is integrated an interface agent with multiple functions. Firstly it keep and develop the user's profile in order to take better decision regarding the quality of the information presented. For each user, the interface agent will observe how, when and in what context the information are used (which are the first address visited, which are the addresses visited frequently and so on). The interface agent can use a direct feedback from the user : he can say if the search was good and how interesting are the founded information. All these information will be used in order to enhance the user's profile.

4. Related works

At this time SEA is an example of a IR MAS, helping the user manage the "information overload" problem often wencounterd when using a WWW. The services provided by existing information search tools on the Internet can be devided into four main functions: search, storage, access aand organisation. There are many systems which offer some or all of these to the WWW user, including WAIS, Archie, the Harvest system and Jasper [9]. Divergent with them, we look at the search process as a whole, instead of the pages stored in the ultimate part of the user's request. We capture the user's choice, the rational behind each of them, the open questions related to the request, the assumption behind it, and any related supporting information.

As Gori et al [17] in their implementation, to provide a usable interface, we considered both a vocal device based on the simplest sounds that can be emitted clearly and a system to aid user interaction by means of prediction. Contrasting with them, our predictions are process-based and context-situated, not merely a statistical one. Moreover, predictions are somehow meta-informative and not situated in the on hand appealing page.

5. Conclusions

Although the separated features of the SEA have been treated separately by the current approaches, the trends of economical environment impose the need of an osmotic approach able to deal with heterogeneous resources. Compared with traditional search engines, SEA promotes a more anthropocentric orientation, improve data access capabilities and communication ability. Compared with older approaches, it greatly enhances the IR effectiveness on the Web, reaches more extensive problem domains, more component problem-solving capability. To carry out these functions, three kinds of knowledge are identified that SEA have to deal with: user's *cognitive*, *situational*, *affective* and possible other elements that bear upon efectiveness of retrieval [18].

7. References

1. Moers, C. (1951). Zatocoding applied to mechanical organization of knowledge. *American Documentation*, **2**, p. 20-32.
2. Belkin, N.J., C. Cool, Acest Stein and U. Thiel (1995). Cases, scripts, and information seeking strategies: On the design of intercative information retrieval system. *Expert Systems with Applications*, **9**, p. 379-395.
3. Maglio, P.P. and R. Barret (1997). How to buil Modeling Agents to Support Web Searchers. In A. Jameson, C. Paris and C. Tasso (Eds.), User *Modeling: Proceedings of the Sixth International Conference*, UM97, Vienna, New York: Springer Wien New York.
4. Hutchins, E. (1990). The Technology of Team Navigation. In J. Galagher, R. Kraut, and C. Egido (Eds.). *Intellectual Teamwork*, Erlbaum, NewJersey.
5. Maturana, F.P., D.H. Norrie (1997). Distributed decision-making using the contract net within mediator architecture. *Decision Support Systems***,** Vol. 20, p. 53-64.
6. Carley, K.M., Z. Lin (1995). Organisational designs suited to high performance under stress. *IEEE Transations on Systems, Man and Cybernetics*, **25**(2), p. 221-230.
7. Zamfirescu, C.B. and F.G. Filip (1999). An agent-oriented approach to team-based manufacturing systems. In: H. Van Brussel and P. Valckenaers (Eds.). *Proceedings of the Second International Workshop on Intelligent Manufacturing Systems*, Katholieke Universiteit Leuven Press, Belgium, p. 651-658.
8. Stone, P. and M. Veloso (1997). *Multiagent Systems: A Survey from a Machine Learning Perspective*. Carnegie Mellon University, Pittsburgh, PA
9. Davis, N.J., R. Weeks and M.C. Revett (1997). Information Agents for the World-Wide Web. In H.S. Nwana and N. Azarmi (Eds.), *Software Agents and Soft Computing. Towards Enhancing Machine Intelligence*, Springer Verlag, Berlin, p. 81-99.
10. Balanovic, M. (1997). An adaptive Web Page Recomandation Service. In *Proceedings of Agent Autonomous Agents*, ACM Press, Marina Del Rey, California USA, p. 378-385.
11. Salton, G. And M.J. McGill (1983). A*n Introduction to Modern Information Retrieval*, McGraw-Hill.
12. Harman, D. (1994). Overview of the third Text Retrieval Conference (TREC-3). In *Proceedings of the 3rd Text Retrieval Conference***,** Gaithersburg, MD.
13. Porter, M. (1980). An algorithm for suffix stripping. *Program,* **14**(3), proces 130-137.
14. Pazzani, M., J. Muramatsu, and D. Billus (1996). Syskill & Webert: Identifying interesting web sites. In *Proceedings of the 13th National Conference on Artificial Intelligence*, Portland, OR.
15. Vranej, S., M. Stanojevic (1995). Integrating multiple paradigms within the blackboard framework. *IEEE Transactions on Software Engineering*, **21**(3), pp. 244-262.
16. Pallmann, D. (1999). *Programming Bots, Spiders, and Intelligent Agents in Microsoft Visual C++*. Microsoft Press, Redmond, WA.
17. Gori, M, M. Maggini and E. Martinelli (1997). Web-Browser Through Voice Input and Page Interest Prediction. In A. Jameson, C. Paris and C. Tasso (Eds.), User Modeling: Proceedings of the Sixth International Conference, UM97, Vienna, New York: Springer Wien New York.
18. Saracevic, T., A. Spink and M.M. Wu (1997). Users and Intermediaries in Information Retrieval: What Are They Talking About? In A. Jameson, C. Paris and C. Tasso (Eds.), User Modeling: Proceedings of the Sixth International Conference, UM97, Vienna, New York: Springer Wien New York.