Information Gathering System: Internet Navigation

Elhadi Shakshuki
Systems Design Engineering
University of Waterloo
Waterloo, Canada, N2L 3G1

Hamada Ghenniwa
Mackenzie Financial Corporation
Toronto, Canada, M5S 3B5

Mohamed Kamel
Systems Design Engineering
University of Waterloo
Waterloo, Canada, N2L3G1

Abstract

The World Wide Web (WWW) is a very vast, dynamic distributed source of information. A significant amount of time and effort are commonly required to locate, retrieve and integrate information. Also, different users may have different views and interest for the same subject matter. We have developed a multi-agent cooperative information gathering system (CIGS) that assist different users to locate, retrieve and integrate information on the WWW. The system utilizes the Internet search engines to locate and retrieve information resources based on keyword terms. It utilizes the user interests and preferences to identify the relevant document. Both the user's topic of interest and the information in a document are represented in terms of topic and quality. The system constantly modifies the relevance of the retrieved information to the user's topic of interest and preferences by learning from the user's feedback. A prototype of CIGS was implemented using the IBM Agent Building Environment Toolkit. In this implementation, we demonstrated how CIGS can be used to improve and enhance the Internet search engines' functionality, performance, and quality.

1 Introduction

The Internet is an important distribution medium for a variety of information. Information can be available on the Internet (known as the WWW) before even it is officially announced.

Although, the Internet provides the infrastructure platform to access the available information sources, there are several critical issues that need to be addressed in order to provide value for it. Firstly, the user is no longer able to easily locate the relevant information to his/her interest. Secondly, most of the information resources are dynamic. Finally, the information system environment is open, i.e. not only information might be updated, rather, new heterogeneous information sources are constantly being added or removed.

With these issues in mind, the problem of how to gather information that might be distributed becomes very critical. Currently, Internet search engines are one of the commonly used tools for information gathering. Although these engines are of valuable service, they provide primitive services with several disadvantages. The search for information is limited and typically biased towards indexing more `popular' information, and each search engine covers a small portion of information resources on the Internet [Lawrence and Giles, 1998].

An information gathering system in an open environment is a system that is capable of locating, retrieving and integrating information. The design principles for such systems are as follows. The system should pro-actively search for information from different distributed sources and avoiding repetitive users intervention to satisfy their interest. The system should provide information that is relevant to the user's interest. In addition, it should monitor and update the changes of the information resources periodically. Finally, the system should provide information to the user within the specified time limit defined by the user.

As a result of the progress in the fields of distributed databases and distributed artificial intelligence, it becomes possible to design information gathering systems that provide users with the illusion of a single information system. This approach is taken by this work, in which each information resource is modeled as an independent agent that has the ability to make decisions on its actions without external interference. Additionally, the agents are allowed to interact with each other intelligently at the goal level. These agents are cooperative in the sense that they are able to help each other to fulfill the users' requirements.

The proposed architecture of Cooperative Information Gathering System (CIGS) is based on two types of agents, namely, User Agent and Resource Agent. A User Agent provides flexible mediation services between the user and the information environment. Resource Agents act as information providers for the User Agents. The user submits a query to the User Agent. The query represents the user's desire which includes the topic of interest and the related constraints. Then the User Agent contracts with several Resource Agents. Resource Agents may utilize the Internet search engines to locate and retrieve documents that are relevant to the user's interest.

This paper is organized as follows. Section 2 presents a view for information and how it is used to model documents and user's topic of interest. The proposed architecture of CIGS with detailed description of the agents' design is given in

Section 3. Section 4 describes the implementation of CIGS. Section 5 provides some results with a discussion. Related work is discussed in section 6, and section 7 presents the conclusions.

2 A view of Information

The terms 'data' and 'information' are usually used interchangeably in a number of different senses. In this paper, *Data*, refers to a collection of electronic signals (bits) that can be used as a basis for computation and reasoning; *Information*, refers to a collection of data that has a 'meaningful' pattern. This distinction facilitates how information can be viewed in a computer based environment.

We use this view to represent the characteristics of physical entities known as information resources. One of these characteristics is *topic* (or the context) of the information resource. It is modeled as a specific pattern of concepts. A topic, such as weather, may be represented in terms of several concepts, like temperature and pressure. Thus, a topic can be represented as a set of words, each of which represents a concept. In addition, we use the notion of *quality*, to quantify the degree of the relevance of the information source content with respect to a topic or a concept. Modeling information as a pattern of concepts allows the agents to effectively reason about the content of the information resources. This view of information is used to model documents as well as the user's interest, as described in sections 2.1 and 2.2.

2.1 Document Representation

The content of information resource might be of different structures such as flat files, semi-structured or structured. In this work the focus will be on semi-structured information, in particular documents.

A widely used method [Salton and McGill, 1983] for determining the topic (context) of a document is based on identifying individual words, eliminating the commonly used words, such as *for* and *the*. Then, using the remaining words for concepts identification of a document. Words may also be truncated to word-stems instead of considering complete words. Porter [1980] has introduced a stemming algorithm that transforms and returns one common stem for a word that has different forms. For example, the word `travel'' is a stem for: `travel', `traveled' and `travels'. Then, a document, say d, can be represented using the vector space model [Salton and McGill, 1983] as: $d = < (t1, w_{d,t1}),\ldots, (tj, w_{d,tj})>$, where $w_{d,tj}$ represents the weight of word tj in document d.

In this model, $w_{d,tj} = tf_{d,tj} \times idf_{tj}$ where $tf_{d,tj}$ is the number of occurrences of word tj in document d, and idf_{tj} is the inverse document occurrence of the word tj in the collection of documents (idf_{tj}). One commonly used measure of the inverse document occurrence is given by $idf_{tj} = \log(N_D/n_{tj})$. Where, n_{tj} is the number of documents containing word tj, and N_D is the total number of documents in the collection.

2.2 The User's Interest

The user's topic of interest can be represented similar to that of the document. Then the user's interest in a specific concept within the topic is represented as a weight associated with the concept. Formally, the user's topic can be represented as: $u = < (c1, w_{c1}),\ldots, (cj, w_{cj})>$. The calculation of the weights (w_{cjs}) can be based on the weights of the user's interest that may change. Using this representation, the similarity between a document u and the user's topic of interest u can be measured in terms of the scalar product of their weight vectors.

To determine the weights of the user's topic of interest (or to learn about user's preferences) the user is assumed to provide a positive or negative feedback about the information retrieved. Accordingly, the agent modifies the weights of the concepts constituting a topic of interest by moving the vector of interest closer to the vector representing a document that received positive feedback and away from vector that received negative feedback. Also, the user can selectively update a specific weight.

3 Cooperative Information Gathering System (CIGS)

CIGS is a multi-agent system, in which each agent cooperatively interacts with other agents to fulfill the users' needs. There are two types of agents, User Agents and Resource Agents, as shown in see Figure 1. The User Agent facilitates the interaction between the user and the information environment. The Resource Agent provides the relevant information to the User Agent.

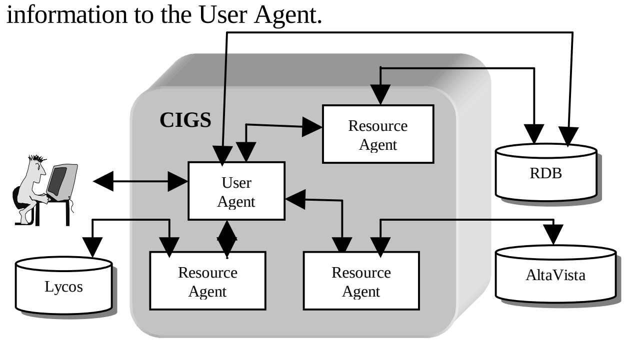

Figure 1: Cooperative Information gathering System Environment.

The design of the agents is based on the CIR-Agent model [Ghenniwa, 1996]. This model consists of the following components:

- Problem solver –provides the agent with the capability of reasoning about its knowledge to generate the appropriate solution that to satisfy a goal.
- Interaction devices –provide the agent with the capability of interacting with the user and other agents, through the communication component. These devices include

- Assignment device –provides the agent with the capability of delegating goal(s) that cannot be achieved locally to other agents.
- Knowledge update device –provides the agent with up-to-date information about the environment and the world.

- Learning –provides the agent with the capability of learning about the user and the other agent's capabilities and interests.
- Knowledge –includes the information that an agent has in its memory about the environment, itself, the current and the expected world.
- Communication –provides the agent with the capability of sending and receiving messages to the elements of the environment.

3.1 User Agent

The User Agent can be viewed as a facility that allows the user to interact with the information system environment. The User Agent responsibility is to receive queries submitted from the user and provides the user back with the relevant information. This information might be retrieved from local and/or remote resources through Resource Agents. The main components of the User Agent are shown in Figure 2.

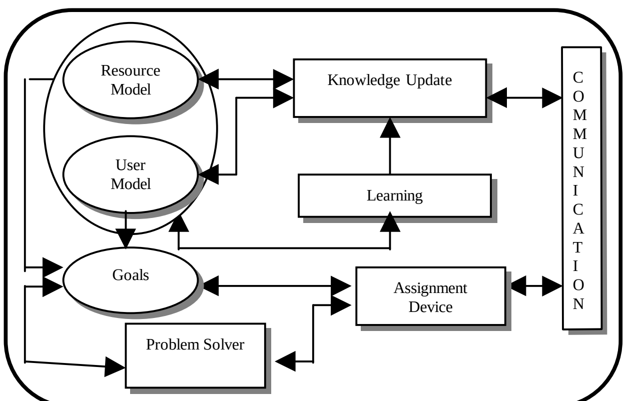

Figure 2: User Agent Architecture.

The User Agent accepts a query in a form described by a set of words that include the user's topic of interest and the associated constraints. The user's interest includes information quality and the response time. The problem solver then generates a solution for achieving this goal. The goal might be decomposed into sub-goals. A goal or a sub-goal might be locally achievable; otherwise, the User Agent interacts with the user and/or other agents.

The User Agent interacts with the user, to learn about the user's interest. The User Agent also learns about other agents' capabilities, in terms of topic and quality of information that they can provide, as well as the time required to deliver the information.

3.2 Resource Agent

The Resource Agent acts as an information provider that has a direct access to the information resources. The responsibility of the Resource Agent is to receive queries submitted by the User Agent and return the relevant information. The main components of the Resource Agent are shown in Figure 3.

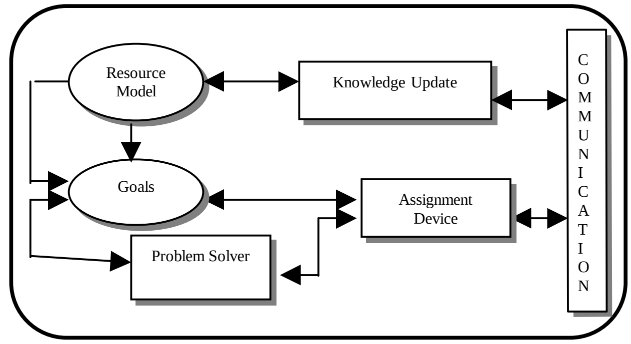

Figure 3: Resource Agent Architecture.

Due to the dynamic nature of the environment, information resources are constantly changing by, adding new information, deleting old information or modifying existing information. Hence, through the knowledge update component, the Resource Agent periodically monitors the information resources based on a predefined time frame. This capability allows the Resource Agent to provide the User Agent with an appropriate set of information resources and consequently the User Agent ensures the retrieval of an up-to-date information.

The Resource Agent utilizes the search engines as a meta-data for the relevant information. At this stage the relevance of the documents is based on a set of words. Among those documents the agent then selects those that match the user's topic of interest. The Resource Agent accomplishes this process by identifying the documents, downloading and then extracts the semantic features i.e. the topics and their qualities. Therefore, the architecture provides context extractor as one of its components. The Resource Agent saves the extracted semantic features into relational database. The agent local database contains the following:

- Concept –contains all the locally available information i.e., topic and quality, and other fields such as the address of the information.
- DocInformation –contains all the information needed about documents referenced in concept table such as words, words frequency and weights.

4 Agents Implementation

The implementation of the CIGS is based on IBM Agent Builder Environment (ABE) [1997]. The ABE supports a rule based, forward chaining inference engine that can be used as the reasoning mechanism. The knowledge base representation is based on KIF [Genesereth *et al.*, 1992], that facilitates knowledge interchange independent of the implementation technology.

The agent knowledge is constructed as classes. Models of the user and the resources are instances of these classes. This approach provides flexible use of the system by different users and allows changes to agent knowledge without affecting the functionalities of other components. The facts are of three different types. Long-term facts are build and stored persistently. Examples of this type are related to the definitions of concepts. Short-term facts are originated from adapters through trigger events or sensors such as user's current action or the time allowed for search. And the third type is derived facts, which are generated from a successfully fired rule such as the topic of a document or its quality.

The user interacts with the User Agent of the CIGS through a set of dialogs. These dialogs are implemented using Java's Abstract Windowing Toolkit (AWT).

The agents of CIGS are implemented using Java and provide access to relational databases via the Java Database Connectivity (JDBC) and Microsoft's Open Database Connectivity (ODBC) interfaces.

The agents can communicate in two ways: (1) through TCP/IP stream sockets or through shared memory. In the former, the agents may reside in different computation environments. In the latter, the agents may reside on different process spaces, but at the same computation environment.

All the agents in CIGS are implemented as a stand-alone Java application. To achieve our goal in providing a high performance system, explicit thread management is used to support concurrency of the agents' components on the same process space. Thus, the User Agent can perform other activity, for example accepting another queries from the user, while waiting for the results of the queries assigned to the Resource Agent.

5 Results

Three different scenarios were constructed to demonstrate the functionality of CIGS for gathering information from the WWW using the Internet search engines. In the first scenario, we utilize CIGS without its full features. This reduces it to a direct use of a search engine with two added features. The first is that the user is allowed to request a limit on response time. The second feature is the system guarantees that all of the retrieved addresses point to existing documents. In this scenario, a user submits a query to the User Agent in the form `Weather at Waterloo Ontario'. The User Agent then assigns this query as a goal to be achieved by the Resource Agent, which utilizes AltaVista search engine. This Resource Agent is constraint to provide information within a specified time. The Resource Agent then submits this query to AltaVista search engine to access and retrieve only the existing address among the top ranked addresses by the search engine. The number of addresses returned by the Resource Agent depends on the time constraint. Finally, those addresses are returned to the User Agent and displayed to the user. The first scenario demonstrates that the User Agent returns only the addresses of information ranked by the search engine that is pointing to existing documents.

In the second scenario, the utilization of the user's topic of interest and preferences for different users are demonstrated. In this case, the CIGS allows a user to define concepts based on his/her view of interest. In this scenario, two different users submit the same query to their User Agents. Each user has different interest for the same topic `weather'. The first user describes the `weather' in terms of `humidity and pressure', while the second user describes it in terms of `temperature, wind and visibility'. When the user submits the query, the User Agent identifies the type of the goal. If it is locally achievable then the information is presented to the user. Otherwise, the goal is assigned to the Resource Agents. In this scenario two Resource Agents are utilized, Agent-A utilizing AltaVista search engine and Agent-L utilizing Lycos search engine. The User Agent assigns the same goal to both Resource Agents along with the constraints. The constraints include the time allowed for search by each agent, the quality of information and the user's topic of interest. Each Resource Agent submits the query to their respective search engines. When the Resource Agents receive the addresses from the search engines, they access the documents and perform the ranking based on the user's topic of interest and preferences. When the Resource Agents return the results to the User Agent, the User Agent integrates the information received, eliminates the duplicate and ranks them based on their qualities. In this scenario, the most relevant document for each user is different. This scenario demonstrates the ability of CIGS to customize the results for users with different views of the same topic.

In the third scenario, we run CIGS with its full features. In addition to the features mentioned in scenario one, the system provides other features. The first is that, the user is allowed to choose the quality of information. This feature has an impact on the User Agent' response time i.e., high information quality might result into a low response time, when the information is remotely achievable. However, the objective of the User Agent is to provide the user with at least the needed quality given the time constraint. The second feature is the ability of the system to cover large portion of information resources on the Internet because of utilizing more than one search engine. The third feature is that the system provides more relevant information, overtime, by learning from the user's feedback. In this scenario, a user submits a query and he/she had no definition for the topic of interest. Also, the User Agent has no knowledge about this topic. In this case the User Agent interacts with both Resource Agents to assign the query. It is similar to the behavior of the second scenario with the exception of the constraints imposed. Upon receiving answers from the Resource Agents, the User Agent presents the information and interacts with the user for a feedback. At this time the User Agent builds a model of the user's interest toward the requested topic and modifies it during future interactions.

6 Related Work

There has been much work recently in the area of software agents to assist the user to navigate the WWW. Most of this work is aimed at making agents performing some actions for achieving goals on behalf of the user, making the interaction easy and efficient.

Many attempts, such as [Ackerman *et al*., 1997; Balabanovic, 1997; Menczer, 1997; Moukas, 1996], have developed WWW based agents to help the user finding interesting and relevant pages. These agents may learn from user feedback using vector model. All of these attempts utilize word frequency to measure the quality of the documents.

Bollaker *et al*. [Bollacker *et al.*, 1998] have developed an Internet based information agent, CiteSeer, that assists the user in the process of performing a scientific literature search. Although it is of valuable use, it does not incorporate the users view or preferences.

The CIGS is different from the other attempts in that it retrieves and virtually organizes the information on the WWW that fits the user's interest and definitions of topics and concepts. Additionally, CIGS provides answers within a constraint time defined by the user with up-to-date information.

7 Conclusions

This paper describes a CIGS based on multi-agent approach that assists the user in the process of locating, retrieving and integrating information. The retrieved documents are parsed to extract the information that match user's topic of interest. The information are placed into a local relational database with the address of each retrieved document to be used as a meta-information. The system can then use this database for future queries and updates.

Monitoring the information resources by the Resource Agents deliberately has made the CIGS aware of the dynamic nature of the information resources, and thus coping well with an open environment.

The use of the user's view and interest for defining topics allowed the CIGS to retrieve and integrate more relevant information to the user.

This system provides a number of important advantages. The User Agent can run continuously, accepting queries and planning for them while the Resource Agents achieving contracted goals. The User Agent transparently provides the information for the user. The User Agent hides the heterogeneity of the Resource Agents by interacting with them at the goal level. Finally, the User Agent manages the time constraint imposed by the user.

References

[Ackerman *et al*., 1997] Ackerman, M. S., Starr, B., and Pazzani, M. ''The Do-I-Care Agent: Effective Social Discovery and Filtering on the Web", In *Proceedings of RIAO Computer-Assisted Information Searching on the Internet*, pp. 17-31, 1997.

[Balabanovic,1997] Balabanovic, M. ''An adaptive Web Page Recommendation Service", In Proceedings of the First International Conference on Autonomous Agents, pp. 378-385, February 1997.

[Bigus, 1997b] Joseph P. Bigus and Jennifer Bigus, *IBM Agent Building Environment Developer's Toolkit*, User's Guide, Level 6, June 1997.

[Bollacker *et al.*, 1998] Bollacker, K., Lawrence, S. and Giles, L., ''CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications", Proceedings of the 2nd International ACM Conference on Autonomous Agents, pp.116-123, 1998.

[Ghenniwa, 1996] Ghenniwa Hamada, Coordination in Cooperative Distributed Systems, *A Ph.D. Thesis*, Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada, 1996.

[Genesereth *et al.*, 1992] Genesereth, M. R., Fikes, R.E., et al. ''Knowledge Interchange Format Version 3 reference Manaual", Logic-92-1, Stanford University Logic Group, 1992.

[IBM, 1997a] IBM Agent Building Environment Developer's Toolkit, http://www.networking.IBM.com/iag/iaghome.html, 1997.

[Lawrence and Giles, 1998] Steve Lawrence and C. Lee Giles, ``Searching the World Wide Web", Science, Volume 280, Number 5360, pp. 98, April 3, 1998.

[Menczer, 1997] Menczer, F., ``ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery", In Machine Learning: Proceedings of the fourteenth International Conference, pp. 227-235, July 1997.

[Moukas, 1996] Moukas, A., ``Amalthaea: Information Discovery and Filtering Using a Multiagent Evolving Ecosystem", In Proceedings of the Conference on Practical Applications of Agents and Multiagent Technology, April 1996.

[Porter, 1980] Porter, M. F., ``an algorithm for suffix stripping". Program I4 (3), pp. 130-137, 1980

[Salton and McGill, 1983] Salton, G. and McGill, M., Introduction to Modern Information Retrieval, McGraw-Hill, c1983.