協調的エージェントのモデル化と情報検索への応用

岡田ロベルト†　李殷碩‡　白鳥則郎†

†東北大学電気通信研究所
情報科学研究科

‡韓国成均館大学工科大学
情報工学科

ネットワーク上に混在する利用可能な膨大な情報から、実際にユーザが必要とする有効情報を探し出すことは容易ではない。初期の情報検索システムにおいては、ユーザは自分が必要とする情報がどこにあるのかなどの位置情報はもちろん、その情報のサーバとのインタラクション方法などについても熟知しなければならない。 本稿では、エージェント集団による協調機構を考案しそのモデル化を行うと共に、情報検索の領域に適用しその有効性を確認する。

A Model of Cooperative Agents and its Application to Information Retrieval

Roberto OKADA†, Eun-Seok LEE‡ , Norio SHIRATORI†

†Research Inst. of Electrical Communication
Graduate School of Information Sciences
Tohoku University
980, Sendai

‡Department of Information Engineering
Faculty of Engineering
Sung-Kyun-Kwan University
Seoul, Korea

With the explosion in the amount of electronically available information on the network, it is becoming more and more difficult for a user to find his desired information. The set of data that represents the best response to a query may be the aggregation of data acquired from distributed, heterogeneous sources. Also, in early Information Retrieval systems the user needed to know where to access to get the desired data and how to interact properly with the information server. In order to overcome the mentioned drawbacks, some non-agent and other **partially** agent-based approaches were proposed. In contrast, in this paper we propose a **completely** agent-based approach by introducing a **community of cooperative agents** for information retrieval, where the different type of agents cooperate to decrease user load and gain efficiency in the retrieval process.

1 Introduction

The amount of information available in electronic form is growing very fast and this put additional burden on the designers of information systems as well as for the people who use them.

The set of data that represents the best response to a query may be the aggregation of data acquired from distributed, heterogeneous sources [6]. It is challenging for designers of information systems to help people find the information that is of interest, especially in a network like Internet with millions of servers.

In early Information Retrieval (IR) systems, the user needed to know which set of information servers is most likely to have the desired information, how to interact properly with them and how to combine the obtained information from possibly distributed and heterogeneous sources.

In order to decrease the mentioned user load, some non-agent-based and agent-based solutions were proposed. Even for the second approach, they are only "partially" agent-based. Our aim is to show how a "Society or Community of Agents" can work cooperatively for efficient Information Retrieval.

There are a lot of researches regarding agents

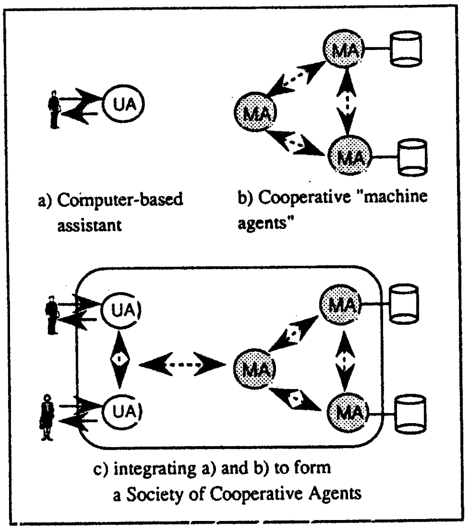

Figure 1: a,b show existing agent researches, c shows Society built with heterogeneous agents

which tries to provide support to humans by doing repetitive, tedious or complex tasks on their behalf. Some of them concentrate on providing "computer-based assistant" to users, helping them perform tedious, repetitive, or time-consuming tasks more easily and efficiently [5, 2] (Fig.1 a). Other researches focus on how agents attached to providers of information or services can work "cooperatively" in order to facilitate access to what they provide [6, 9, 4, 10] (Fig. 1b).

As is not possible to have an all-mighty agent, it needs to work in cooperation with other agents. In this work, our aim is to integrate the existing heterogeneous agents to build a *Society of Cooperative Agents* (Fig. 1c). We propose the CAS model as a means of modelling such a Society.

In section 2, we review the Information Retrieval problem. In section 3, the formalization for the proposed CAS model is given. In section 4, we explain how to map the CAS model to the Information Retrieval domain. In section 5, in order to show the merits of our proposal, we review the characteristic features of the existing approaches for IR on the Internet. In order to overcome the drawbacks of the existing solutions, we introduce a community of agents - Cooperative Agent Society for IR or CAS_{IR} - and by comparing with the existing solutions, we build a table of characteristic features for qualitative evaluation.

2 Information Retrieval

With the explosion of electronically available information, an additional burden has been placed on the designers of Information Systems, as well as for the people who use them. The burden for user are that he has to supply the query and the corpus where to search. Even user having access to multiple corporas, the user (rather than the Information Retrieval Systems) should know which one is most likely to contain the correct answer. In the case of a huge number of corpora available (like Internet), is not possible for a user:

- to know which of the possible sources is most relevant to a query
- to manually submit a query to more than a very small fraction of those sources

From the designer's point of view, the challenge for the designers of Information Systems is to help the user to find the information that is of interest. Basically, there are two ways of helping the user to find information:

a)Browsing : user-guided activity of exploring the content of a resource space, the user has to navigate by himself through the data space.

b) Searching :automated process, where the user provides some information about the resources being sought, and the system locates some appropriate matches.

We can summarize the existing problems as follows:

1. There are many different Systems providing Information, and the user need to have knowledge about the differences to access properly.
2. In case that the user need to search for some information in several sites, connecting and accessing one by one to each of them is boring, time consuming and may imply high cost of communication
3. Some database systems do not allow user easy access to the information they record, and have complex and clumsy interface that insists the user to produce unambiguous request for information.
4. The networkable capabilities of many current systems mean that the datasets to be examined may exist on many machines, in a variety of forms. This means that a variety of access protocols have to be used in order to obtain data in the first place, and then it has to be converted into a suitable format for integration with the other datasets currently under investigation.

3 Modelling Society of Cooperative Agents

In this section, we will introduce our general model for a Society of Cooperative Agents - the CAS model (CAS stand for Cooperative Agent Society). We view the Agent Society as composed of 2 domains: one domain for the set of agents and their environments and the other domain for the task to be assigned to the agents. For a given task T in the domain of TASK, the most suitable agents to execute the given task form the Agent_Community_T in the domain of agents.
At the top level, the CAS model is defined in terms of the TASK to be done and the Agent Community AC most suitable to execute the TASK.

$$\mathsf{CAS} = < \mathsf{TASK}, \mathsf{AC} >$$

Both components of CAS are further decomposed, where the Agent Community AC is expressed in terms of the set AGENT and their environment Env.

3.1 TASK model

TASK (which consists of a set of elements task) is defined in terms of requirement req and the associated set of constraints CONSTR, where each element constraint is denoted by cntr. Each constraint is a $< attribute, value >$ pair and can express hard or soft bounds on resource usage, level of user involvement expected, etc.

$$\mathsf{task} = < \mathsf{req}, \mathsf{CONSTR} >$$

Ex:CONSTR = < Time_limit = 1 hour, Monetary_limit = 10,Usr_involv = None >.

3.2 AC (Agent Community) model

Agent Community AC (which consists of a set of elements Agent_Comm) is defined in terms of a set AGENT and the associated environment Env.

$$\mathsf{Agent\_Comm} = < \mathsf{AGENT}, \mathsf{Env} >$$

3.2.1 Env model

One of the component of the Agent Community - the environment for the agents - is defined in terms of resources and links. The table above shows the detailed definition of each part.
The descriptor descr defined in resource rsc is used to describe the utility of a given entity with respect to a topic. quality can be a vector describing

■ ENV = < RESOURCE, LINK >

- **RESOURCE = { rsc_r }**

  where rsc = < entity, descr, position >

  entity = < network|info_source|server >
  descr = < topic, quality, quantity, cost >
  position = < relative ,absolute >

- **LINK = { $link_l$ }**

  link = { origin, dest, cost }

some features of quality, like completeness, certitude,precision, etc. quantity can be expressed as percentage for entities like info-source and as (low, medium, high) for entities like network. cost can be expressed as cost/Kb of data transmitted for network and link, cost/document searched or retrieved for info-source, cost/CPU for server.

3.2.2 Agent model

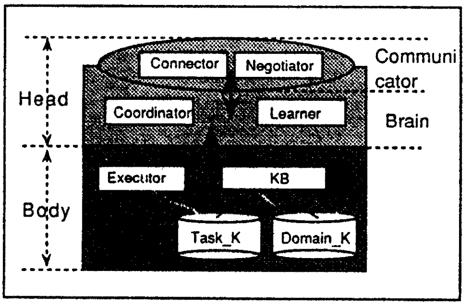

Figure 2: CAS agent

The general model proposed here - by making analogy with human - clearly separates units requiring high level of intelligence of an agent (which we will include in the HEAD) from its executive capability (included in the BODY).

agent BODY: contains relatively static and passive elements. They wait to receive the proper message in order to execute a pre-defined method to produce an action, but they cannot become self-activated. They should wait until an instruction from the HEAD (intelligent part of agent) comes.

$$\mathsf{Body} = < \mathsf{Executor}, \mathsf{KB} >$$

- Executor:This is the module that actually executes the task , each action corresponds to methods in the object-oriented paradigm. The actions are passive and wait for a message to act.

$$\text{Executor} = \{ \text{actions} \}$$

- KB (Knowledge Base)

$$\text{KB} = \langle \text{Task\_K}, \text{Domain\_K} \rangle$$

agent HEAD: this is the portion that contains the necessary elements for intelligent behavior and cooperation with other agents. Also here, the necessary MetaKnowledge is included.

$$\text{Head} = \langle \text{Communicator}, \text{Brain} \rangle$$

*** Communicator** This facilitates the communication with other agents and is also responsible for Conflict Resolution during task performing and resource sharing.

$$\text{Communicator} = \langle \text{connector}, \text{negotiator} \rangle$$

i. connector: which provide

- access to appropriate telecommunication channels and
- network information about the agents (such as their addresses)

ii. negotiator: This module is responsible for Conflict Resolution (CR) during task performing and resource sharing.

$$\text{Negotiator} = \{ \text{CR\_strategies} \}$$

*** Brain** As the name suggests, this part is responsible for the "intelligent behavior" of the agent. Brain make possible to work cooperatively with other agents.

$$\text{Brain} = \langle \text{coordinator}, \text{learner} \rangle$$

i. coordinator: responsible for cooperation and recovery for problematic situations.

ii. learner: this module is responsible for one important aspect of intelligent behavior, i.e learning to adapt to new situations to try to improve efficiency.

4 Information Retrieval System based on CAS model

In this section, our aim is to apply our general CAS model proposed in the previous section to the important application domain of Information Retrieval. We named the Information Retrieval System based on our CAS model as ***Cooperative Agent Society for Information Retrieval*** or CAS_{IR} in short.
In order to proceed with the mapping, we start from the top-level definitions.

a) mapping TASK to the IR is simply to express the task as a query and the associated constraints.

b) the Agent Community for IR will be composed of the set of agents necessary for IR and their environment. In the set of necessary agents for IR, we consider three types of agents: user agent, task agent and machine agent.

The justification for introducing the three types of agents mentioned before are as follows:

User Agent (UA) takes care of the user preferences and helps him to formulate proper query.

Machine Agent (MA) control access to the information they provide. By doing so, negotiation capability, security and consistency of data can be gained.

Task Agent (TaskA) composed of Manager and replicas. Manager have Domain-specific knowledge and plan to satisfy user requirements in the corresponding domain. The replicas makes possible the parallel search and from heterogeneous sources.

Lets see how the different agents interact in order to satisfy the user request. Assume one user agent per user, which learn its user preferences and help him to interact with the system in a transparent way. With the introduction of this agent, we can decrease the load from the user about knowing where to access to search for information and how to interact with the accessed information source. The user agent "dialogue" with its user in order to get the request , then analyze it and tries to determine which Domain-specific Manager(s) can handle the user request properly. Once the suitable Manager(s) has been found, this Manager (which is expert in his domain and knows the specific terminology and how to execute tasks) will start the Domain-specific dialogue with the user in order to get the necessary domain-specific details. Once the Manager get the user request, start planning to determine how to satisfy the request and which Information Sources can contribute to the solution. After finishing the analysis, decomposition (if possible) and allocation of tasks, Manager will generate replicas or "envoys" which will carry the assigned task , go to the Information Source site and interact at the remote places. Some research on this kind of "migrating or mobile agents" are being considered as a means to decrease the network load of having permanent links from the beginning to the end of the interaction between the requester and the server. Additional possible features include the possibility of having parallel search (by sending different replicas to different places) and to deal with heterogeneous servers (by generating the replicas understanding the specific server protocol). The "replicas" reach the remote place and start interacting with Machine Agent responsible for controlling the access to the information they provide. The candidate solutions found are sent to the Manager which compose the sub-solutions or resolve conflicts when necessary, and the final set of solutions are given to the user. Addition of Machine Agents to the Information Server make possible of having negotiation capabilities, security and also consistency of data (the agents will ensure the consistency).

5 Implementation and Evaluations

As a means for evaluating our proposal, we consider qualitative and quantitative evaluation. The Qualitative evaluation is done by comparing the characteristic features of the existing approaches for Information Retrieval on the Internet against our proposed system.

5.1 Non-Agent Based Approach on Internet

With the Internet having seen an explosive growth in recent years, a number of services have arisen on the Internet to help users search and retrieve documents from servers around the world. Examples of these applications are WAIS, Gopher and WWW. The problem with these systems is that although they allow user to search through a large number of information sources, they provide very limited capabilities for locating, combining and processing information. The load of finding information is still on the user, who has to follow links and this hyperspatial navigation is complicated.

5.2 Partially Agent Based Approach

Several searching tools have been developed, basically in two types [1]:

1. a client-based search tool e.g:Fish [1] that does automated navigation, thereby working more or less like a browsing user, but much faster and following an optimized strategy. The disadvantages of this approach [1] are :
   - retrieving documents through the Internet can be time-consuming.
   - the use of network resources is sometimes considered unacceptably high.
   - the fish-search can only search documents for which links are found in other documents.
   - waste of resources by transferring redundant information. If two users from the same site perform individual searches there is a possibility of the search spaces overlapping and the same information being returned to each user. [3]
   - wasting system resources.
2. a gateway, offering (limited) search operations on small or large parts of the WWW, using a "pre-compiled database". The database (index or catalog) is often built by an automated Web scanner (a "robot"). The disadvantages of this solution are:
   (a) User need to know: With various indexes available, users must choose which one to use. While interfaces (such as SUSI, and the CUI meta-index) can help by removing the need to locate an index first (user load of finding the proper interface by himself!) , they do not provide users with information about which services the indexes offer.
   (b) Maintenance: While a centralized index allows users to perform *flat searches* (i.e without regard to how the indexed information is organized), it can suffer consistancy problems as the amount of resource data increases. This problem led "Archie" (one index service on Internet for FTP) to settle on a compromise of allowing any piece of directory information to be as long as 30 days old. This inconsistency may be acceptable for data changing relatively slowly, but for quickly changing data, a centralized index is difficult to manage [7] .
   (c) Security: no security matters.

(d) Not possible to search for a set of related keywords ,sequentially or in parallel. So a *combnined service* is not possible (like planning a trip, which includes flight reservation, hotels reservation, etc.).

This is true in general for the existing searching systems: although search is much more flexible and general than browsing, it may still be hard for the user; forming good queries can be a difficult task, especially in information space unfamiliar to the user.

5.3 How our proposed CAS_{IR} overcomes the problems

The problems we found in the study of partially-agent based systems presented in the previous section are summarized as:

1. User need to know
2. Help to formulate query
3. Maintenance
4. Security
5. Combined service

For (1), we found that the user need to first locate and select the most suitable index, in our case the task of finding the proper Manager will be done by the User Agent. For (2) , it is difficult for the user to formulate the proper query, especially in unfamiliar domains. In our case, once the UA have selected the most suitable "Domain Specific" Manager, according to the user request , the Manager and the User Agent will help the user in formulating the proper query. For (3), by attaching a Machine Agent (MA) to the information source. This MA will take the responsibility to communicate any important changes occurred to the associated information sources to the proper agents. For (4), by having this MA controlling the access to the information they provide, it is possible to solve the security problem and also is possible to add the capability of "negotiating for information". For (5), the Domain-specific Manager can divide the task into related sub-tasks, plan which MA can satisfy the sub-tasks , generate replicas holding the sub-tasks and send them to the selected MA. With the introduction of replicas it is possible to gain the capability of parallel search.

6 Conclusions

The existing agent researches focus on some specific portion of the whole problem and tries to introduce agents to solve them. We identify this group of works as Partially-Agent based. In contrast, in this research we proposed a model for a *Society of Cooperative Agents*, the CAS model, which is Completely-Agent based approach. In order to study the applicability of our proposed model, we map the model into one very important domain of application ,the Information Retrieval domain. To prove the effectiveness of the CAS_{IR}, we need to evaluate it, and we provide a qualitative evaluation by comparing existing approaches for Information Retrieval and our proposed model mapped to IR. The future works include completing the implementation and then work on the quantitative evaluation.

References

[1] De Bra and Post. Information retrieval in the world-wide web: Making client-based searching feasible. In *First International Conference on the WWW*, 1994.

[2] Palaniappan etal. The envoy framework: An open architecture for agents. *ACM Transactions on Information Systems*, Vol. 10, No. 3, pp. 233–264, 1992.

[3] R. Fielding. Maintaining distributed hypertext infostructures: Welcome to momspider's web. In *First International Conference on the WWW*, 1994.

[4] C. Knoblock, Y. Arens, and Ch. Hsu. Cooperative agents for information retrieval. In *Proc. of the 2nd. Intl.Conference on Cooperative Information Systems*, 1994.

[5] P. Maes and R. Kozierok. Learning interface agents. In *Proc. 11th National Conference on Artificial Intelligence*, 1993.

[6] T. Oates, N. Prasad, and V. Lesser. Cooperative information gathering: A distributed problem solving approach. Technical Report UMASS 94-66, Dept. of Computer Science, Univ. of Massachusetts, 1994.

[7] M. Schwartz, A. Emtage, B. Kahle, and B. Neuman. A comparison of internet resource discovery approaches. *Computing Systems*, Vol. 5, No. 5, 1992.

[8] N. Shiratori, K. Sugawara, T. Kinoshita, and G. Chakraborty. Flexible networks: Basic concept and architecture. *IEICE Transactions*, Vol. E00-A, No. 1, 1993.

[9] D. Steiner, D. Mahling, and H. Haugeneder. Collaboration of knowledge bases via knowledge based coordination. In *CKBS '90 Proc. of the Int'l Working Conf. Cooperating Knowledge Based Systems*, 1990.

[10] Ch. Yahata and M. Takizawa. Negotiation strategy for multiple autonomous agents. マルチメディア通信と分散処理ワークショップ, 1993.