6Y−2

# A Study on a Realization of Multi-agent System on a Distributed Environment *

Roberto Okada† Eun-Seok Lee‡ Norio Shiratori†
†Research Institute of Electrical Communication, Tohoku Univ.
‡Dept. Information Engineering,Sung-Kyun-Kwan Univ., Korea

## 1 Introduction

In modern societies, information is becoming very valuable and locating then when required constitutes a very hot issue. However, this task put a high load on the user's due to the explosion in the amount of electronic information dispersed over a world-wide large scale information network - like Internet - where traditional retrieval techniques are insufficient. Some agent-based solutions were proposed, but many drawbacks remains still unsolved. In this paper, we introduce a completely agent-based approach for Information Gathering, where several agents come together; we explain how this "community of agents" - by cooperation and negotiation - decrease user load and gain efficiency in the retrieval process.
The proposed multiagent-based architecture has the following merits: (a) *sharability*: many users can share these expert agents; (b) *complexity hiding*: often IG involves quite complex coordination of many different agents. Having the user interact only with his User Agent hides the underlying complexity, (c) *modularity and reusability*: these software agents - pieces of code - can be copied from one user to another with little modification or adaptation to take into account particular user's preferences, (d) *flexibility*: these software agents can interact in new configurations "on-demand" depending on the information requirements of a particular task.

## 2 Architecture for Completely agent based Approach

As one agent alone cannot cope with all tasks, our aim is to integrate the existing heterogeneous agents to build a *Cooperative Agent Society (CAS)* and we propose a model for this CAS, detailed in [1]. Then, we applied the proposed CAS model to one important domain of application - Information Gathering (IG) ; named "Completely-agent based approach for IG".

**Design concepts:**
1. To satisfy user requirements, "multiple agents" actually perform live search at heterogeneous remote sites. Domain experts determines what sites to search and the paths to the best solution quality with lowest search cost.
2. This approach should "abstract" from the user the low-level details, i.e the user gives to the agents high-level user goals and they dynamically synthesizes the appropriate sequence of steps to satisfy those goals.
3. The key point for cooperative behaviour is that "when an agent faces an unfamiliar situation, it consult its peers who may have the necessary experiences to solve it". Over time, these agents build up a trust relationship with each of its peers

**Architecture:**
The Agent Community for IG will be composed of the set of agents necessary for IG and their environment. In the set of necessary agents for IG, we consider three types of agents: (1) User Agent (UA): one per user, it takes care of user preferences and helps him to formulate proper query, (2) Machine Agent (MA) attached to the information sources, it control access to the information they provide. By doing so, negotiation capability, security and consistency of data can be ensured, (3) Manager (Man): have Domain-specific knowledge and plan how to satisfy user requirements in their domain of expertise. Also, they can generate the "envoys or replicas (rep)" of themselves and send these rep to interact at the remote site with the MA of information servers. The rep allow parallel search from heterogeneous sources.
Each agent is composed of : (a) communicator, (b) negotiator, (c) coordinator, (d) learner, (e) executor, (f) knowledge base; as described in [1].

**Interaction between agents:**
Assume: (i) there are several domain-specific Man, (ii) each agent knows at least its own ability i.e when it is asked by other agents "how much ability has to answer about a given topic $\tau$ - represented by $K(\tau)$,", the requested agent can answer with the tuple given by $< \tau, quantity, quality, cost >$; where *quantity* is how much it knows about $\tau$, *quality* is how good is its knowledge about $\tau$ and *cost* is how much will cost to make use of its ability.
(1). The user - by interacting with his UA - give his request in an abstract way (i.e. not giving the low-level details). This UA first should find the proper Man/s with the ability to satisfy at least some portions of the request. This UA - when it lack knowledge to take this decision - will ask for help to other known agents. This UA learns also the user preferences and will help him - in cooperation with the selected "domain-specific" Man - in proper query formulation (which is difficult for users in unfamiliar domains); (2) selected Man will "plan" how to respond to user request i.e convert the high level goal into a sequence of specific actions to take. To do this, the Man ask to several MA for $K(\tau)$, the requested MA respond with $< \tau, quantity, quality, cost >$ and based on these parameters (representing cost, network resource

*分散環境上でのマルチエージェントシステムの実現に関する一考察

usage, speed) Man will select the set of MA to work with. Negotiation for information between Man and MA will be included at this step, (3) Man receives the responses from the different MA, process it (e.g compose, resolve conflicts, reorder) and return to the UA, (4) the user will provide feedback about how good is the solution presented. This feedback is used as relevance feedback. This learning will change the "trust" relationship between the agents.

# 3 Evaluation

**Existing solutions and their drawbacks:**
The existing solutions for locating information on Internet are based on two paradigms: *browsing* and *searching*. We classified them from the agent point of view into non-agent and "partially-agent" based approaches.
Regarding the partially-agent based - named because the existing agents cover only some portion of the necessary functions between the user and the information sources - according to the two paradigms mentioned before, we have:
* **Agents helping browsing**: Systems like WebWatcher, Letizia, interactively advises Web users - in the browsing process - about which hyperlink to follow next; it "*learns*" by observing the user's reaction to its advice.
* **Agents helping searching**: "Indexing agents" are the most popular type of agents on the Web (e.g Lycos, WebCrawler, InfoSeek). Indexing agents carry out a massive, autonomous search of the Web and store an index of words from document titles and document texts. The user can query the agent by asking for documents containing certain keywords.

**Our Completely-agent based solution:**
The details of the implementation is described in [2] Our proposal can overcome the weak points of existing solutions:
**(1) User need to know** where to locate and then select the most suitable servers. We abstract this from the user, his UA - by cooperating with other agents - will find the suitable Man;
**(2) Help to formulate query**: is needed, especially for users facing unfamiliar domains. The proper "domain-expert" Man will dialogue with the user in "domain-specific" terms;
**(3) Maintenance**: is needed to ensure consistency of data. The MA attached to the information servers will communicate any important changes occurred in the associated information sources to the proper agents;
**(4) Combined service**: with existing approach is not possible; e.g planning a trip, which includes flight reservation, hotel reservation, touristic information is handled separately. In our proposal, Man 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, parallel searching is possible;
**(5) Security and negotiation**: 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".

| | NON-AGENT based (WWW) | PARTIALLY AGENT-based (INDEX) | Proposed COMPLETELY AGENT-BASED (CAS_IG) |
|---|---|---|---|
| USER LOAD | | | |
| Decrease necessary user Knowledge (WHERE, HOW) | △ | ○ | ◎ |
| Help user in formulating query | × | △ | ○ |
| Facilitate location of information | △ | ○ | ○ |
| QUALITY OF INFORMATION | | | |
| Security and Negotiation | × | △ | ○ |
| Avoid garbage | △ | △ | ○ |
| Maintenance | △ | △ | ○ |
| Search related topics in parallel | × | × | △ |

Figure 1: Comparison of approaches

# 4 Conclusion

"Computer for everybody" seems to be the trend, where the type of users will expand to non-experts. Agent-based paradigm is the solution which will abstract the users for the low-level tasks i.e he would give to the agents his requirements and the proper set of agents come together and plan how to satisfy it. In this paper, we propose a completely agent based approach for information gathering, where the agents supports the users, represents the user to the system and handle complex interactions with other cooperating agents and system resources.

# References

[1] R. Okada, E. Lee and N. Shiratori: "A Society of Cooperative Agents on the Information Network: Towards Intelligent Information Gathering" in Proc. 1995 Int'l Conference on Network Protocols

[2] 木原 英人, 岡田 ロベルト, 李 殷碩, 白鳥 則郎: "マルチエージェントシステムの実装方法に関する一考察", 情報処理学会第 52 回全国体会講演論文集