Coalition among Information Agent Based on Cost Model

4Y－9

Saeyor SANTI Mitsuru ISHIZUKA
Dept. of Information and Communication Eng.,
Faculty of Engineering, University of Tokyo
e-mail: {santi,ishizuka}@miv.t.u-tokyo.ac.jp

1 Introduction

The Web repositories are exposed in form of a huge amount of heterogeneous information sources. The information is overwhelming the users. Besides the target information itself, the changes upon the previously released information are significant and worth being notified to those who perceived the out of date information as soon as possible. The changes made in Web repositories occur at unpredictable rates. Unfortunately, stock type information source has no means to inform its prospective users about the changes While the stock type information source occupies a large percentage of sources on the Web, it is necessary to have a system that monitors changes on the Web, and provides comprehensive presentation to the prospective users. In this paper, we propose a mechanism that provides change monitoring and presentation service for a large group of users by coalition among service agents. The service agents keep improving the overall utilization factor by several schemes based on the decision made by game analysis. We apply a cost model to the service mechanism in order to study the cooperative behavior of the service agents. The reduction of cost is designed to comply with the level of cooperation among service agents. This paper presents a paradigm for the service and the formation of cooperative behavior of the agents in the community.

1 Matchmaking among service agents

Besides the self-organization in each service unit, the distributed service units work in coordination with the Matchmaking Agent as shown previously in Fig. ??. The coalition among Service Agents is made possible by the Matchmaking Agent. The agent has 3 main functions as follows:

1. **Request matching**: The agent tries to match nearly identical requests and dispatches them to appropriate Service Agent in the community.
2. **Issuing suggestions to users and Service Agents**: The agent has opportunity to evaluate the resource sharing efficiency because it holds the service profiles of the Service Agents in the community. It tries to adjust the load balancing and utilization factor of the overall system. The utilization factor can be improved if the agent can find new users who request nearly identical requests to the existing ones. The Matchmaking Agent also implements the same game analysis policy described in previous section to issue suggestions to the users.
3. **Providing information of services on demand**: Service Agents in the community may query the service profiles from the Matchmaking Agent. This profiles is necessary when a Service Agent deals with the users according to the access behavior of others.

The Matchmaking Agent cooperates with the Service Agents in the community by interpretation of incoming request objects as follows:

1 Request Object

Request Object:

Function:	*Request_function*
Input:	*Description, Constraint Input_obj*
Output:	*Description, Constraint Output_obj*
Owner:	*Service_Agent*
Time:	*Issued_time*

Figure 1: *The template of a request object.*

The request object is a common template for requests posted to the Matchmaking Agent. A request object consists of requested function, input, output, owner, and time stamp. The functions currently implemented are listed as follows:

- **Declaration of service profiles**: In order to made the existing requests available to other service units, the Service Agent declares its service profiles to the Matchmaking Agent.
- **Load request**: When the Resource Manager considers that the load in the unit is still low

compared to the available resources, it may give charity to the community by issuing a load request to the Matchmaking Agent. The constraint of the request object indicates the acceptable number of requests. The agent assigns requests to the service unit if available. The Matchmaking Agent returns the request object that the output was assigned a set of requests back to the service unit.

- **Load distribution request**: In some cases such as a service unit become overloaded or needs to be temporarily closed, the service unit has to distribute its services to others. It issues one or more request objects to the Matchmaking Agent in order to distribute the requests to available service units.

- **Effectively identical requests matching**: When a user request a service, the in charged Service Agent consults with the Resource Manager in the unit about an effectively identical request. In the case of no effectively identical request available, the Service Agent sends this request object to be served by the Matchmaking agent. In the worst case, there is no effectively identical request in the community, the Matchmaking Agent relays the request to an appropriate service unit. This can be both the service unit that issued a load request or the site which the Matchmaking Agent considers that the load is still low. However, the elected service unit has right to refuse the request base upon constraint on the location. If the request is not accepted by any service site, the in charged Service Agent takes the request into its own service unit.

2 Decision making in matching process

A cost model is applied to the service mechanism to investigate cooperative behavior of service agents in the community. The invited users under the invitation of preference adjustment requests and the new users gather together to get assigned to appropriate service groups. This implies that they desire to play in an assignment game in order to improve profits. Right in this process, the requests will be clustered into groups of prospectively identical requests. These groups are applied to a cost model for finding acceptable matches. The cost for a service is defined on the concept that the more effective identical requests, the lower the cost of each service in the identical group. We define C_i, the cost per service for each user in group i^{th} base on F, the full cost per user, as follows:

$$C_i = F(1 - k\frac{e^{\frac{g_i}{N}} - 1}{e - 1}) \quad where \;\; 0 \leq k \leq 1 \qquad (1)$$

The cost reduction increases exponentially according to the number of identical requests in the group. This create a persuasion for the users to join a large group even the request for that group is not exactly match what they want since this can reduce their costs. Another effect of the cost model is that the service costs in the community become lower remarkably as the number of identical requests increases. As a result, the competitiveness of the community becomes higher in case that we consider the service in multiple communities. The above is the evaluation of value of the services on the service provider side. We define the appreciation value of the user j^{th} against the service of the group i^{th} by u_{ij}. Then, the difference value for the service of group i^{th} viewed by the user j^{th} is $a_{ij} = u_{ij} - C_i$. If we have a set of service groups $G = \{g_1, g_2, ...g_n\}$, and a set of users $H = \{h_1, h_2, ..., h_m\}$, then we can find the maximum profit combination by finding the maximum value of $\sum a_{ij}$ among the combination space of G and H.

3 Conclusion

In this paper, we presented the mechanism of Web repositories change monitoring service that notifies the users about changes in Web repositories and creates comprehensive summary pages of the changes. The improvement of overall utilization factor is derived from the resource management within each service unit and the cooperation among service units. The Matchmaking Agent is the key of coalition. The coalition among service units brings about a broader scope for request matching. Moreover, the Matchmaking Agent has potential to balance the services among service units. At the same time, the mechanism of load transfer based on coalition among the service units strengthen the robustness of the service.

Decision making process of both Matchmaking Agent and Service Agent in each service unit relies on the game analysis. The expected payoff values based on experience in the past have direct impacts to the decision. The cost model promotes the degree of cooperation by compromising the user needs based upon the maximum profits. As a result, this increases the identical requests dramatically compared to matching only by similarity of incoming requests.

References

1) Santi Saeyor, Ishizuka Mitsuru *WebBeholder: A Revolution in Tracking and Viewing Changes on the Web by Agent Community* Webnet98 3rd World Conference on Internet Technology, Nov. 7-12, 1998, Orlando, Florida, USA

2) Santi Saeyor, Ishizuka Mitsuru *Longest Common Tag Sequence Algorithm for precise reviewing of changes in the WWW* The 56th Annual Convention, Information Processing Society of Japan