

# Agent-Based Peer-to-Peer Service Networks: A Study of Effectiveness and Structure Evolution\*

Yathiraj B. Udipi  
Dept of Computer Science  
North Carolina State U  
Raleigh, NC 27695, USA  
ybudupi@ncsu.edu

Pinar Yolum  
Dept of Artificial Intelligence  
Vrije U, Amsterdam  
The Netherlands  
pyolum@few.vu.nl

Munindar P. Singh  
Dept of Computer Science  
North Carolina State U  
Raleigh, NC 27695, USA  
singh@ncsu.edu

## Abstract

*We study peer-to-peer service networks consisting of autonomous agents who seek and provide information services. Agents potentially help each other by giving referrals to guide the search. Each agent autonomously decides whom to contact for a service and whom to provide a service or a referral. Service networks evolve as the agents change their neighbors to improve how their needs are fulfilled. If an agent autonomously decides to, it may cache some responses from other (information) services. We observe that even a small cache improves agents' success in discovering needed services and enables a few initial service providers to serve the information needs of many. Caching induces clustering of agents based on interest.*

## 1. Trustworthy Service Networks

The purpose of information access is to fulfill a need. In conventional systems, information access is mapped to database lookup and an emphasis is placed on looking for correct (as from relations) or relevant (as from text) results. By contrast, in open settings, correctness and relevance may not be as clearly independent of users, so we must look for authoritative sources that can provide correct and relevant results. The results may be available from multiple sources and hence are not necessarily unique. The need for finding authoritative sources leads us to develop a decentralized approach wherein peers can find peers who are trustworthy and offer high quality services.

**Basic Communication Protocol** A trustworthy service network consists of *principals* that are computationally rep-

resented via *agents*. When an agent is in need of a service, it begins to look for a trustworthy provider for the specified service. The agent queries some other agents from among its *neighbors*, which are typically a small subset of the agent's acquaintances. A queried agent may perform the specified service or may provide *referrals* to other agents. The querying agent may accept a service offer or may follow referrals that it receives. Hence an agent may receive the required service from its neighbors or through a series of referrals. Agents thus keep track of the trustworthiness of other agents by querying services or by flexibly taking referrals.

**Service Caching** With the basic communication protocol, the agents need not always get an answer to their query, but whenever they get an answer it is generated from scratch by some provider. Not all agents are service providers and hence all the agents depend on the few service providers present. This causes a potential bottleneck for three reasons. One, because of the incomplete connectivity of the agents in the network, many times not all consumers can reach the service providers [3]. Two, if the providers are reachable by all, then the providers can become overloaded with queries. Three, if the agents do not learn the services offered and if their principal requests that service again, they will have to repeat the process of service location.

These shortcomings motivate us to find a way by which we can reuse some of the services already offered by some providers. Information services can be easily cached by the agent, so that the providers need not generate them again. Caching aids the search for information since an agent that is looking for information can find it in some cache—its own cache or the cache of another agent. The caching of a service by an agent takes a dual role: first to satisfy its principal's needs and second to serve the other agents who are in need of that service.

**Representation** Each agent models its acquaintances based on their *expertise* (i.e., the quality of the services they

\* This research was supported by the National Science Foundation under grant ITR-0081742. This is a revised version of a paper appearing in the AAMAS 2003 Workshop on Agent Oriented Information Systems.

provide), and *sociability* (i.e., the quality of the referrals they provide). An agent may have varying levels of interest and different grades of expertise in multiple domains. The interests and expertise of the agents are represented as term vectors from the vector space model (VSM)[1], each term corresponding to a different domain. Each of the dimensions of the vector can have values ranging from zero to one. The higher the value, the closer is the match to that particular domain. Sociability is captured as a scalar. An agent generates a query based on its interests.

The expertise and sociability of the acquaintances are adapted based on the service ratings given by its principal. When an answer comes in, the modeled expertise of the answering agent and the sociability of the agents that helped locate the answerer (through referrals) are updated based on the quality of the received answer. At certain intervals, each agent can modify its choice of neighbors based on these acquaintance models.

A cache consists of a set of entries that contain the answer as well as some information about the quality or appropriateness of the given answer. A cache entry is hence a  $\langle query, answer, quality \rangle$  triple. The quality of the cache entry is modeled based on how appropriate it is to the owner of the cache based on its interest.

### An Example Search

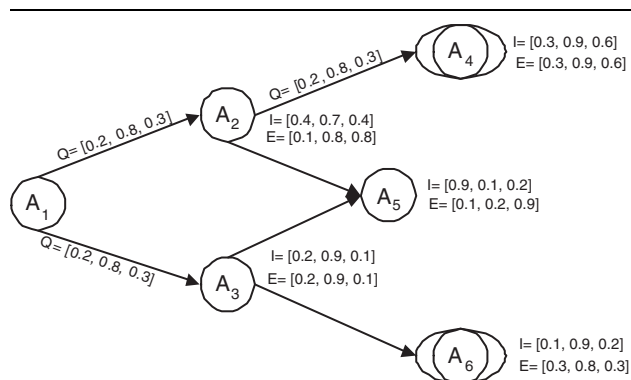


Figure 1. An example search

Let us describe the protocol using the example network of Figure 1. Here,  $A_1, A_2, A_3, A_4, A_5,$  and  $A_6$  are agents. An edge from agent  $A_i$  to agent  $A_j$  means that  $A_j$  is a neighbor of  $A_i$ . For each agent, the vectors  $I$  and  $E$  represent interest and expertise respectively. Agent  $A_1$  is interested in an item that is cast into a query vector of  $[0.2, 0.8, 0.3]$ . Agents  $A_4$  and  $A_6$  have the desired item. When  $A_1$  does not find an answer in its cache, it sends the query to  $A_2$  and  $A_3$ .  $A_2$  receives the query and checks its cache to see if any matching answer exists there. Not finding the answer in its cache,  $A_2$  decides to send the query to its neigh-

bors, since its interest is similar to the query.  $A_2$  sends the query only to  $A_4$  because  $A_4$ 's expertise as modeled by  $A_2$  matches the query. After  $A_2$  receives the item from  $A_4$ , it evaluates the answer and if useful, puts the item into its cache and forwards it to  $A_1$ . Meanwhile, since  $A_3$  is not interested in the query, it just sends a referral ( $A_5$  and  $A_6$ ) to  $A_1$ .  $A_1$  sends out queries to agents for which it received referrals and the procedure is repeated. After evaluation,  $A_1$  caches only the useful answers.  $A_1$  increases the modeled expertise of  $A_2$ , as  $A_2$  provides the answer (returned from  $A_4$ ) and increases the sociability of  $A_3$  for having given a good referral ( $A_6$ ).  $A_1$  decreases the sociability of  $A_3$  for having given a wrong referral of  $A_5$  and the expertise of  $A_5$  is also decreased for not having given any answer. During a neighbor change,  $A_1$  may decide to drop  $A_3$  in favor of  $A_4$  or  $A_6$ .

## 2. Evaluation

We study the impact of caching information services in service networks. Our simulations contain 400 agents and 4 domains in the vector model. There are 20 or 40 *experts* (5 or 10 in each domain) who provide services—these are the agents who originate the information that others may cache. Service providers do not generate queries and therefore do not have any neighbors. All the service consumers have four randomly chosen neighbors. During the course of the simulation, after every three queries each agent has a chance of modifying its choice of neighbors, but the number of neighbors for each agent remains the same. Each agent ranks its acquaintances based on an equal weight of both sociability and expertise. The top-ranked acquaintances become the neighbors. As the agents change neighbors, they find neighbors that better suit their interests. The simulations are run for a duration corresponding to 10 neighbor changes.

Our preliminary results are that even with small caches and fewer number of service providers, the agents can find answers successfully. If the agents are selective in the answers they cache or the answers they serve from their caches, then the chances of circulating poor answers are reduced. Further, caching results in the agents with similar interests to group together. The above study supports design rules to help decide when and in what situations to apply caching to maximize success while discovering services.

## References

- [1] G. Salton and M. J. McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
- [2] M. P. Singh, B. Yu, and M. Venkatraman. Community-based service location. *Comm. ACM*, 44(4):49–54, Apr. 2001.
- [3] P. Yolum and M. P. Singh. Emergent properties of referral systems. *Proc. AAMAS*, pp. 592–599. 2003.