Intelligent Agents Serving Based On The Society Information

Sanem SARIEL
Istanbul Technical University, Computer
Engineering Department, Istanbul, TURKEY
sariel@cs.itu.edu.tr

B. Tevfik AKGUN
Yildiz Technical University, Communication
Design Department, Istanbul, TURKEY
akgunbt@yildiz.edu.tr

Abstract

In this paper, we propose a serving system consisting intelligent agents processing society information in a multi-user domain. The agents use the similarity information on the user preferences for providing services with the best quality. This similarity information is formed by using the clustering techniques. The agents are allowed to interact with others and exchange information based on this information. The agents' beliefs are formed based on the clustering process results and the user feedback for the information gained by the interactions. These beliefs are used to provide a continuous service for the user. The natural division of tasks among the agents reduces the workload for each agent. The cluster structure of the society may be changed according to the satisfaction levels of the users. Some general society parameters of the system are learned by a reinforcement learning method. We designed the agents by using our proposed social interactive agent model. The system model can be used for many of the serving systems by changing the service type.

Keywords
Socially intelligent interactive agents, user modeling, clustering, learning.

1. Introduction

While designing any kind of serving system for users, if the user preferences examination is added to the system, the service provided to the user will be more satisfactory. Moreover if the user satisfaction is tested and the service is changed based on this feedback, the serving system will become more intelligent. Therefore considering user profiles and interacting with the user should be implemented for intelligent service systems. However serving based on the user's directives as in supervised learning makes the system having lack of intelligence, and also it can burden the user. For an effective work, interaction among service providers of different users can be implemented. For implementing such an interactive work, intelligent agent systems [10][15] can be used. If the serving agents are designed with the capability of interacting and processing society information for providing the best service, the agents become more intelligent. The agents in this kind of processing learn from both the user and the other agents. This kind of serving systems such as mail or web document filtering [16][7], information broadcasting and management [3][4], and recommendation [1] systems can provide the service with the best quality.

In our proposed system, we designed our agents based on our social interactive agent model. The agents interact with others to provide the best service for users. The interaction partner choice decision is made based on the information about the groups of the society. These groups are formed by clustering techniques. The provided service will be similar for the users in the same group. Therefore the individual tasks for user agents in the same group can be divided among them because of having similar user preferences. Our agents act autonomously to increase the service quality, and they can change interaction partners based on the user feedback. The overall society group structure may change over time. Some general society parameters such as components of common dictionaries about services are learned by the reinforcement learning method by assigning critic role to each user in the society.

2. The Proposed Serving System

The proposed system consists of User Interface Units (UIUs) consisting service units (SU), and communicating with users, User Interface Agents (UIAs), a Clustering Agent (CA), and existing JATLite [9] components. All of the system components can be seen in Figure 1.

The main contribution of the system is to provide the best service for the users. This necessitates a very huge workload for individual service systems. Moreover, the user wants to give less information while getting the best service. Therefore a real social interaction model is required in a multi-user domain. To overcome this difficulty, a model analyzing multi users solutions should be used. So the system was designed as consisting intelligent agents interacting with both the user and the others. Every user is assigned to an intelligent agent, called UIA. These agents are designed based on our proposed social interactive model that will be explained in the next section. While agents serve the users, some tasks can be naturally divided in real time. This natural division means that, at a time while an agent needs for some service parameters and tries to find the necessary parameters for this service, another agent may still have no requirement for these parameters. The parameters may be found for the first agent, and it may be later used by the second agent and vice versa. The ready-made solution of the first agent may be satisfactory for the second agent, and therefore some redundant tasks are eliminated for some agents. In a live system in which there are many working agents of active users, the utility factor of being in a multi-user environment will be very high.

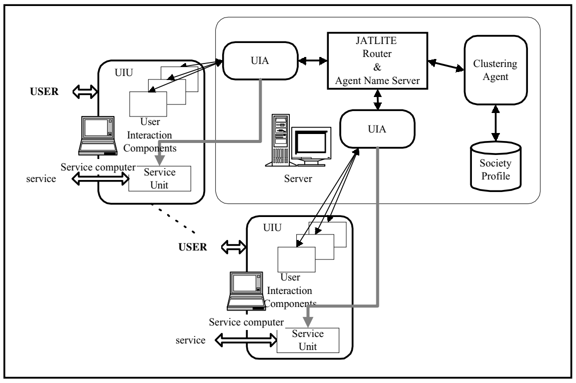

Figure 1. The Overall System Diagram

The agents are designed to provide the best service continuously. To provide the best service, UIAs should autonomously decide which action to take. This decision is based on the social interaction model. The UIA should ensure that user's satisfaction level is high after serving. Consequently the user satisfaction is always examined in real time. The agent arranges its actions based on these feedbacks and formed beliefs. The UIA should be informed about the other agents which can provide the necessary parameters that the user is most satisfied. This information comes from the clustering results. The clustering process is implemented by the CA. The CA takes some information from all of the users via their UIAs to form clusters. This information type should be previously determined. It can be information about some general characteristics of users (ie, age, nation, sexuality, etc.), answers of some questionnaires, or a combination of both of them. This information should be much enough to determine the cluster structure of the users and little enough to avoid burdening the user. While a UIA interacts with its user, it gets some feedback from the user via the corresponding UIU. UIU has the necessary components for the interactions. The feedback is used for the agent to act adaptively to provide the best service. We modeled the UIA as parameter generator unit for the required service. These parameters are fed to the specialized SUs and the user is served according to them. The SU may provide related services for the users according to the selected application domain. The agent interactions based on the society cluster information and the adaptive service model were mainly focused in this work. The service of a SU is out of scope of this model. SU should be designed according to the desired service. We have chosen the story telling agents domain to apply the designed system. In this domain, the service parameters are graphical facial muscle parameters. The users are served with personal graphical facial expressions [12].

The user in the system can get the service from any service computer. Because the modeled system components were written in Java, there are few limitations on the platform of the service computer. Although a user is not currently connected to the system, its corresponding UIA is alive because it can help the others about formerly formed service information of its own and it can get real updates on society information continuously. The UIAs live on the server machines. System components on the server part shown in Figure 1 can be distributed in the network.

To implement such a system, we needed an agent construction and communication tool. We just needed a tool implementing communications. The JATLite [9] is appropriate for our system. Because we implemented our own interactive social agent model, there is no need that the chosen tool supports the social behavior of agents. The JATLite Router Unit in the system implements agent registration processes and there is also a Naming Server (ANS) connected to the Router. The KQML [6] layer provides the agent communication is implemented according to the speech-act theory.

3. The Interactive Social Agent Model

In our proposed social agent model, every agent has to serve according to the own user's profile and decide autonomously to which action to take. The profile generation and updating process is made continuously. The agent has to adapt itself according to the user's demands that may change over time.

In the social context, if the interactions among people are analyzed, it can be seen that one person's everyday life decisions may be affected by the others' experiences. This affectivity occurs according to the person's reliability degree to the others. This social situation is modeled in our system by using an influence constant. The social agent interacts with the user and the other agents in its life cycle. It helps the society information to be collected and it behaves as a real social entity having some behavior similar to human. This approach is designed to simulate real interaction dynamics among individuals in a society. The interactions among agents are firstly random. The influence constants are also random and determined by the system. Thus the modeled system is very close to the real world.

It is necessary to assign users into clusters, to serve them effectively. The social classes of users can also be changed over time. This affects the general cluster structure of the system. Therefore the problem turns to a problem with dynamically changing interactions. The service quality based on the gained knowledge of the agent should be appropriate to the user's demands.

3.1 The Proposed Model

In our proposed interactive social model, the agent is responsible for providing the best service for its user. This is implemented by the autonomous and intelligent behavior of the agent. The agent has to choose which action to perform. This choice depends on the user's profile, the collected information from the other agents and feedbacks taken from the user. The Interactive Social Agent Model can be seen in Figure 2.

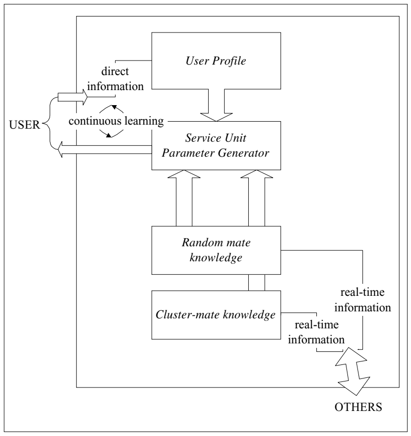

Figure 2. The Interactive Social Agent Model

The information for an agent to serve can be from four different types of sources: the user, an external human expert, the other agent from the same cluster, or the other random agent.

The agent generates its user's profile and the beliefs based on the taken information from these different sources. The direct information is provided by the user or a human expert. The beliefs are updated based on the information from cluster-mate agents or the information from others, and the corresponding user feedbacks.

It is useful to use the direct information to provide service for users. However to collect all of the information from the user is almost impossible. In most application domains the user wants to give less information and to get the best service. Therefore a real social interaction model is required in a multi user domain. The agent should use previously gained information and beliefs to choose the appropriate information source. According to the chosen source, the service is provided to the user. After serving the user, a feedback is taken from the user. By using the feedback, the agents' beliefs are updated. According to the chosen source and taken feedback, the society general cluster form may need to be changed. Thus the general society concepts are updated continuously.

One agent's beliefs are formed firstly after determining the user's social cluster membership information. This information comes from the CA. According to this information, the agent knows the cluster-mate agents and relies upon them to get help for unknown service parameters for the user. There are *n* agents in the multi agent system. If the total number of cluster-mate agents having the necessary service parameters is *m*, an agent's newly formed knowledge K_i (corresponds to the necessary parameters and it can be implemented as an array of parameters) is determined by the Eq. (1) as an abstraction. In this equation K_l is the gained knowledge based on the information from the l^{th} cluster-mate agent and the user's feedback.

$$K_i = \frac{\sum_{l=1}^{m} K_l}{m} \qquad (1)$$

After serving the user with this newly formed knowledge for a while, the user satisfaction is examined. If the user satisfaction is not enough, the agent can change its interaction partners and choose some other random agents to get desired parameters. The agent initiates a random interaction process. The beliefs about the other agents from different clusters are formed based on the information from them, the influence constant and the user satisfaction level. The influence constant provides the heterogeneity of the system and also determines one agent's reliability degree upon others. Therefore the social dynamics will be different for each agent. This constant is adapted from the real life as a human personal characteristic property. The Eq. (2) determines K_i^+, the i^{th} agent's newly formed knowledge to provide the required service in a random interaction process. c_i is the influence constant of the agent (0! c_i ! 1). K_i, is the knowledge of the i^{th} agent gained by taken information from the cluster-mate agents (Eq.(1)). K_j is the gained knowledge about the taken information from a random agent. This agent is not from the same cluster.

$$K_i^+ = \frac{2K_i + (K_j - K_i)c_i}{2} \qquad (2)$$

In this equation, if c_i =0, the agent is not affected by the random information. In this case, to increase the satisfaction level of the user, direct information should be provided from either the user or an external expert. If c_i=1, the user is directly affected by the random information. In this case, the cluster-mate agents information and the random information is considered equally. The c_i constant can be determined according to the own user's personal characteristics by some questionnaires. Therefore each agent in the system has a different view of the society and beliefs as in nature. Because of the distributed nature of the environment, the interaction among agents will be random. The agent does not have to interact with all the agents as in real societies.

3.2 The Continuous Learning Model

Before generating clusters, some standard information is taken from each user via the assigned UIA. This information is processed to determine the user's cluster in the society. After CA generates the clusters of the society, it can provide cluster-mate agents information for all of the agents. This information is used by the agent to get the new service parameters which are not directly provided by the own user. The same cluster-mate agent's information can be useful and the user may be satisfied with the service. Then the cluster information of the agents need not be changed. If the user is not satisfied with the service, random selection among the other agents is implemented. The agent's knowledge is formed again for the required service based on the information comes from interactions, and it is presented to the user. User satisfaction is examined again. If the user satisfaction level is good enough, a new clustering schema may be formed, because of the similar profiles from different clusters. The agent beliefs are updated and a request for changing cluster information is sent to the CA. In this case, the new satisfaction information and old clustering information should be considered by the CA to form updated clusters of society. If the user is not satisfied with the all of the service provided by the agent based on formed knowledge or there is no sufficient information from the other agents in the system for the required service parameters, some direct information either from the user or an external expert should be requested. This situation may occur initially in the system. Some sample scenarios can be summarized in the continuous learning model as:

- New direct information is taken from the user for a new service. It can provide solution for a new service for the other agents.
- Static information from human experts can first be presented to the user. After this process user satisfaction is examined.
- Interaction is implemented with different users who have the requested solution from the same cluster. After this process user satisfaction is examined.
- If the information taken by interaction from the same cluster does not provide a satisfactory solution, random agent interaction is ensured with the agents having the required service parameters. In this process the cluster contents can be changed, and the users representations in the clusters can be changed.

4. The Clustering Agent Model

The Clustering Agent (CA) is the central unit in the system. It collects all of the users information and generates society profile by the clustering techniques. It can also implement a learning process to determine dominant parameters of the society. The learning process is implemented by getting feedbacks from the users by means of UIAs as a reinforcement learning method.

The main objective of the CA is to form user clusters and to update clusters according to the received requests from UIAs. The clusters are initially generated by using information from all of the agents in the system. After determining the first clustering structure of the users, the CA sends inform messages to UIAs to indicate their cluster-mate agent information. Thus cluster-mate agents can interact with each other.

The important discussion arises from the question at which times will the clustering process be implemented. The solution for dynamic clustering called incremental clustering, by Fazli [5] can be adapted. However, this solution has some problems like order-dependence. Because the order-dependence is still an open question, the clustering process is implemented on some certain number of user information in our system. Although this approach is suitable for small number (i.e. hundreds) of users, it is not an efficient solution when the number (i.e. thousands) of users increases. There should be found more effective solutions for implementing the clustering process dynamically independent from the order.

The CA also determines some application parameters, general concepts about society in the learning process, if necessary. It can implement this process in a reinforcement learning process, by Q-learning algorithm [14]. It uses all of the UIAs as critic for the learning process. It chooses a random critic at a time. This randomness makes the environment non-deterministic because different users assign different feedbacks for the same states.

After generating the society clusters, some requests from individual agents may come about a change on the cluster structure. Because each UIA interacts with its user and examines the satisfaction levels, they can decide that cluster-mate agent information is not appropriate for the own user. This is a reason for changing the clustering structure of the system. The UIA can indicate its belief about similarities with some random mate agent. Consequently the clustering schema can be changed according to the provided information from this individual agent. The individual agent acts based on its beliefs. However, the CA decides if the all of the clustering schema should be changed or not. This is determined by the coming requests from many agents.

4.1 The Clustering Process

All users' information is considered in the clustering process. The incoming information from each user arrives in the CA at a different time step. The clustering process is implemented for certain number of data samples. The feature weights are selected according to the learning process outputs. Euclidian distance is used in the distance calculation.

Different clustering algorithms produce different cluster outputs for the same input data [8]. Cluster analysis on the given data should be made by using different algorithms. For that reason, Fuzzy c-means (FCM) [2], Complete-link (Clink) and Single-link (Slink) [8] algorithms were implemented on the sample data set. Partition coefficient criterion [11] is used with the FCM algorithm. The partition coefficient criterion considers the membership values. To find the optimum number of clusters, the dendogram is cut at a desired level for the Clink and Slink algorithms. The CA produces three different types of outputs for these algorithms. According to the application domain, one of these outputs can be chosen.

4.2 The Converted Q Learning Process For Non-Deterministic Environments

The Q learning [13][14] algorithm is suitable for learning some general concepts of the society. This algorithm searches over a state space, tries to determine Q values, estimated action-values, for the states. In our proposed system, this algorithm is converted as distributing the critic role among users. The learning process is used for determining which parameters are effective for a specified service. The states are the corresponding parameter effectiveness arrays. The learned result state will be the desired parameter effectiveness array for the whole society. The "1" bit in the state representation corresponds to the effectiveness of one parameter in the service. The hamming distance between binary representations of two neighbor states is 1. The neighbors are chosen from the possible states to have greater state value than the current state. There is one bit change on the state transition and the change will be from 0 to 1. Thus the number of neighbors of each state will be different according to its state representation. The Q values of states are stored in a look-up table. The CA holds the state graph for traversing the states and applying Q-learning updates on them. Convergence criterion is met, when a state's Q value increases more than the others.

The environment is non-deterministic, because feedbacks taken from the users will be different for the same situations or states. Because the environment is non-deterministic, to learn the final array requires more steps than the deterministic case. The primary aim of the learning process is to provide additional information for the clustering process. Moreover, some global parameters are being learned. The CA as a learner chooses a different agent to get feedback at each step. There is no importance on which agent criticizing because of the non-deterministic property of the environment. Therefore the scalability property of the system does not change.

In the learning process, the CA visits a state at each step. A service parameter array is mapped to the current state and the service parameters are sent to a random UIA. The related user of this random agent assigns a grade for the provided service. The CA takes the grade from the corresponding user. The reward in our approach is calculated as a function of the grade and the belief of the agent on the grade. The complete algorithm for the converted learning process can be seen in Fig. 3.

```
Clear for all table entries, Q values
Assign the initial state, s
While a convergence criteria is not met
{
Chose a transition from the current state
Generate the service according to the selected action
Update the belief
/* At certain time steps take a list of connected agents. (Connection to the Router & Agent Name Server)*/
Send the corresponding service parameters to a random agent
Wait for the grade
..
Calculate the reward as a function of the belief of the agent and the taken grade
Update Q value of the table entry according to the Q-learning updating rule for non-deterministic environments [14]
Find the new s' state
}
```

Figure 3. Converted Q Learning Algorithm

The belief of the agent indicates a change on the feedbacks between neighbor states. While going from one state to another, the related parameter (the different one) changes in the mapped service. If the subsequent grades are the same and the belief indicates a change on the feedback, a negative reward is taken; otherwise a positive reward is taken, and vice versa for the prediction indicating no change. Fig. 4. presents a sample reward calculation while going from one state to its neighbor. In the sample scenario, the change is on the 4th parameter for the first transition. The corresponding parameter value in the firstly sent service will be different from the next one. The prediction indicates a change on the two subsequent grades. Assuming the feedback taken at the first step from one user is + and – for the next step. As a result, a positive reward is calculated, because the prediction is consistent with the taken grades.

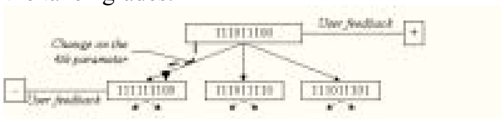

Figure 4. State transition in the learning process

5. Experimental Results

The learning process results were observed in a simulation. The simulation results can be seen in Fig. 5. In this figure, convergence steps for each state can be found. Because different states have different number of neighbors, the convergence steps are different related to the convergence threshold. The parameter array length is a very important value for the convergence step. When the array length increases, the grading process of each user increases. In this case, the parameter reduction and dividing approaches can be used.

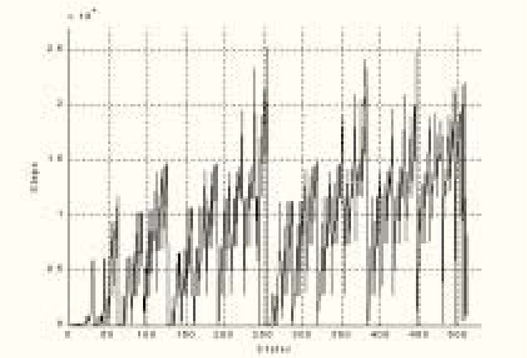

Figure 5. Learning process simulation results

6. Conclusion

In this study, an interactive agents system processing society information and providing the best service for users according to their demands is proposed. To model the specified society, the samples should come from the society itself. The proposed interactive social agent model presents a model for providing the best service by interacting with either the own user or the other agents in the system. The service for one user is determined based on the interaction information in real-time.

The influence constant may be determined according to the characteristics of the user. For instance, the age may be an effective value for determining the influence constant. The specific user groups or societies may be formed to consider their data separately. The societies may be formed by using information about sexuality, age, etc. Therefore differences among groups may be determined. The overall system model is suitable for the systems providing any kind of services according to the users' specific demands. The SU may be any kind of serving unit working for the user. The system generates necessary parameters for this unit.

The learning process is implemented on a distributed user group that forms a scalable environment. The reason for using distributed data in the learning process of the CA is to model all of the society and to ensure just one critic per user for determining parameter affectivity arrays. The simulation of all the grades generated by a small group causes this facility be non-effective because many trials are needed. The non-determinist property of the environment is coped with the Q learning based algorithm in the learning process.

The CA should use dynamic clustering techniques for huge user groups. But results of this process should not be order-dependent. This is still an open question. According to the application domain, the appropriate clustering algorithm should be chosen. An adaptive mechanism choosing the most appropriate clustering algorithm according to the application domain can be added to the system as a future work.

7. References

[1] Active Recommendation System For The Library Without Walls. http://www.c3.lanl.gov/~rocha/lww/

[2] Bezdek J.C.,*Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[3] David E., and Kraus S., "Agents for Information Broadcasting", *ATAL'99*, USA, 1999.

[4] Dignum F. Information, "Management at A Bank Using Agents: Theory and Practice", *Applied Artificial Intelligence*, Vol. 14, 2000, pp. 677-696.

[5] Fazli C., "Incremental Clustering for Dynamic Information Processing", *ACM Transactions on Information Systems*, Vol.11, 1993, No. 2.

[6] Finin T., and Fritzon R., "KQML- A Language and Protocol for Knowledge and Information Exchange", *Proc. of 19th Intl. DAI WorkShop*, 1997, pp.127-136.

[7] Fürnkranz J., Holzbaur C., and Temel R., "User Profiling For The Melvil Knowledge Retrieval System", *Applied Artificial Intelligence*, vol. 16, 2002, pp. 243-281.

[8] Jain A. K., Murty M. N., and Flynn P. J., "Data Clustering: A Review", *ACM Comp. Surveys*, Vol. 31 1999, pp. 264-323.

[9] JATLITE, http://java.stanford.edu

[10] Maes P., "Agents that Reduce Work and Information Overload", *Comm. of the ACM*, Vol. 37, 1994 pp. 31-40.

[11] Ross, T. *Fuzzy Logic With Engineering Applications,* McGraw-Hill Companies Inc, 1995.

[12] Sariel, S., Akgun, B. T., "A Personal Facial Expression Generation System with Interacting Agents", *TIDSE Proc, Computer Graphik Ed*, 2003 pp. 82-93.

[13] Sutton R. S., and Barto Andrew G., *Reinforcement Learning, An Introduction*, The MIT Press, England, 1999.

[14] Watkins C.J.C.H., *Learning from delayed rewards* PhD Thesis, University of Cambridge, England, 1989.

[15] Wooldridge M., and Jennings N.R., "Intelligent Agents: Theory and Practice", *The Knowledge Engineering Review*, Vol. 10, 1995, pp.115-152

[16] Zhank B-T., and Seo Y-W., "Personalized Web-Document Filtering Using Reinforcement Learning", *Applied Artificial Intelligence*, Vol.15, 2001, pp. 665-685.