Turk J Elec Engin, VOL.13, NO.3 2005, © TÜBİTAK

A Model for User Profiling Systems with Interacting Agents

Sanem SARIEL[1], **Tevfik AKGÜN**[2]
[1]*İstanbul Technical University, Electrical and Electronics Faculty,*
Department of Computer Engineering, İstanbul-TURKEY
e-mail: sariel@cs.itu.edu.tr
[2]*Yıldız Technical University, Art and Design Faculty,*
Department of Communication Design, İstanbul-TURKEY
e-mail: akgunbt@yildiz.edu.tr

Abstract

Service systems according to users' personal demands and preferences are on their way to provide required services in many application domains. Emulation of interactive model of human societies produces valuable outcomes for such systems. In this work, a system model with interactive agents for multi user service systems is proposed. A novel social interactive agent model using different interaction forms is also included in this proposal. In this model, agents decide how to serve to their users by considering their users' profiles and information from other agents. User clusters are formed by clustering techniques. The Q-learning algorithm is used for learning some general parameters for the society of the users. The model is evaluated in the personal story teller agent domain as a case study. Facial expressions are generated based on users' demands by these agents in this case study.

Key Words: *Socially intelligent agents, user profiling, clustering, learning, multimedia applications, facial expressions.*

1. Introduction

Agents are autonomous systems that react to the environment in which they live, behave proactively according to own goals, and interact socially by communicating with others [1]. If they can learn and adapt to the environment they are classified as intelligent. There are many types of agent systems in literature [2]. Here in this research, the focus is on User Assistant Agents serving users based on the personal interests, needs and demands. Possible applications of user assistant systems are mail or web document filtering [3][4], information broadcasting and management [5], and recommendation [6] systems. In these systems, agents learn from users in either supervised or unsupervised manner. Users play teacher role in supervised learning cases, whereas critic role in unsupervised learning cases.

Since the main motivation behind the systems with multiple agents is the real world and live societies, usually systems are modeled as multi-agent systems. System agents behave personally, in corporation or as opponents against each other based on the problem domain as in real life [7]. Particularly, in user assistant systems the primary goal is to serve users based on the personal demands [8] meanwhile keeping users' satisfaction level high. In these systems, agent interactions may be used to increase the utility of the system because of the similarities among users and their behaviors. However the interactions among agents should be made appropriately. To achieve suitable interactions, user groups and interaction partners should be defined. Then the information exchange among partners can turn the individual serving process into an effective and collective work.

Forming user groups or determining interaction partners can be implemented centrally or distributed. In the central case, the clustering methods [9] are appropriate to form user groups by considering information related to defined features. In the distributed case, interaction partners can be determined by direct communication among agents related to similarities on knowledge-bases of the agents.

In this paper, a generic type user profiling system model design suitable for multi user environments is proposed. The overall model contains a novel interactive social agent model. In our proposed interactive social model, each agent is responsible for providing the best quality service for its user. User preferences are mainly considered to provide the appropriate services for users. In the model, agents are allowed to communicate with each other for social information exchange. Interaction partners of each agent may be changed dynamically by the agent to serve user's demands (using feedback of the corresponding user). A central unit accessed by all agents process user group information to provide effective interactions among agents for the service quality to be improved.

Since the proposed system model is generic, it can be used for different kinds of service systems by assigning appropriate tasks for the service module in the model. The proposed model is evaluated in personal story telling agent domain as a case study. In this application domain, the service module has capabilities for presenting facial expressions while reading stories. The text recognition and speech generation processes are out of scope of this work but can easily be integrated with the system in the service unit. Each user preference differentiates because of the preferred facial expressions related to different emotional states. Darwin conducted the early works on facial expressions [10]. According to his theories, facial expressions are universal, and social cultures do not affect them to have major differences. There are still many arguments on facial expression generation processes of the human brain [11]. The importance of the cultural effects on facial expressions is still a part of the ongoing discussion. Our designed system may be useful for finding answers to these questions by making experiments on different societies of people. The motivation behind the selection of such an interactive domain is the increasing demand on the systems containing friendly human like interfaces. The possible applications can be advertising agents, news telling agents or intelligent systems forming user groups for different kinds of services.

The organization of this paper is follows: In Section 2, the proposed user profiling model is presented. Section 3 introduces the proposed interactive social agent model, and Section 4 presents the central unit. The application system as a case study and the implementation details are presented in Section 5 and Section 6. Section 7 and Section 8 explain the modified learning process and the clustering process details, respectively. Section 9 discusses the results of the experiments, and Section 10 concludes the paper with possible directions for future work.

2. The Proposed User Profiling System Model

2.1. Requirements

A user assistant system should be capable of providing the desired service while leaving little effort to the user. Therefore the system should maintain the usage of different users' information to reduce the workload of each user. Interactions among entities are necessary in the system. Users' demands may change over time. Such a system should adaptively improve itself based on the user's preferences.

Assumptions:

- The environment is a multi user environment.
- There is a communication infrastructure for interactions among agents.
- There are no privacy issues regarding the shared service information among users.
- The agents are benevolent and can share the correct information related to the users to increase the effectiveness and utility of the system.
- Even users are not connected to the system, representative agents still live in the system.

2.2. Design

The proposed system [12] consists of User Interface Units (UIUs) communicating with users, User Interface Agents, the Central Profiling Agent (CA) responsible for generating society profile, and a Router unit for information exchange and naming services. All of the system components can be seen in Figure 1.

Each interface unit (UIU) is assigned to a specific user. The UIU has special sub-units to provide the service for the selected application domain. User interaction components serve for user I/O. The service unit (SU) is the final service generator unit of the UIU. SUs may provide related services for users according to the selected application domain. Each user interaction component of a UIU is directly connected to the corresponding user interface agent (UIA). UIAs interact with both the user by means of the interaction components and the other agents by means of the communication network managed by the Router unit. These agents are designed based on our socially interactive agent model explained in the next section. While UIAs act autonomously for serving to their users, they may also be helpful for their interaction partners to achieve their own goals.

The Router unit manages effective messaging processes among agents. Each agent is given a name by this unit. The agent connection states are known by the Router. Off-line and on-line communication primitives are supported.

The Central Profiling Agent (CA) is responsible for processing user information to form user groups or determining interaction partners. Therefore it communicates with all agents in the system. It implements clustering processes on the incoming users' information.

2.3. Interactions in the system

The overall goal of the multi agent system is providing an effective interaction network for the individual agents to improve their service qualities. The main goal of the system is increasing the service quality with less effort. This is the reason why the interactions are the key points in the system design. While agents serve the users, some tasks can be naturally divided in real time. This natural division means that, at a time step while an agent needs for a service and tries to find a solution for this service, another agent may still have no requirement for this service. The solution for this service may be found for the first agent and later it may be used by the second agent when it is necessary 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. In a live system in which there are many working agents of active users, utility of being in a multi user environment becomes very high. However the success of the system is highly dependent on the established interactions among agents. Therefore an effective social interaction model is required to effectively use multi user information.

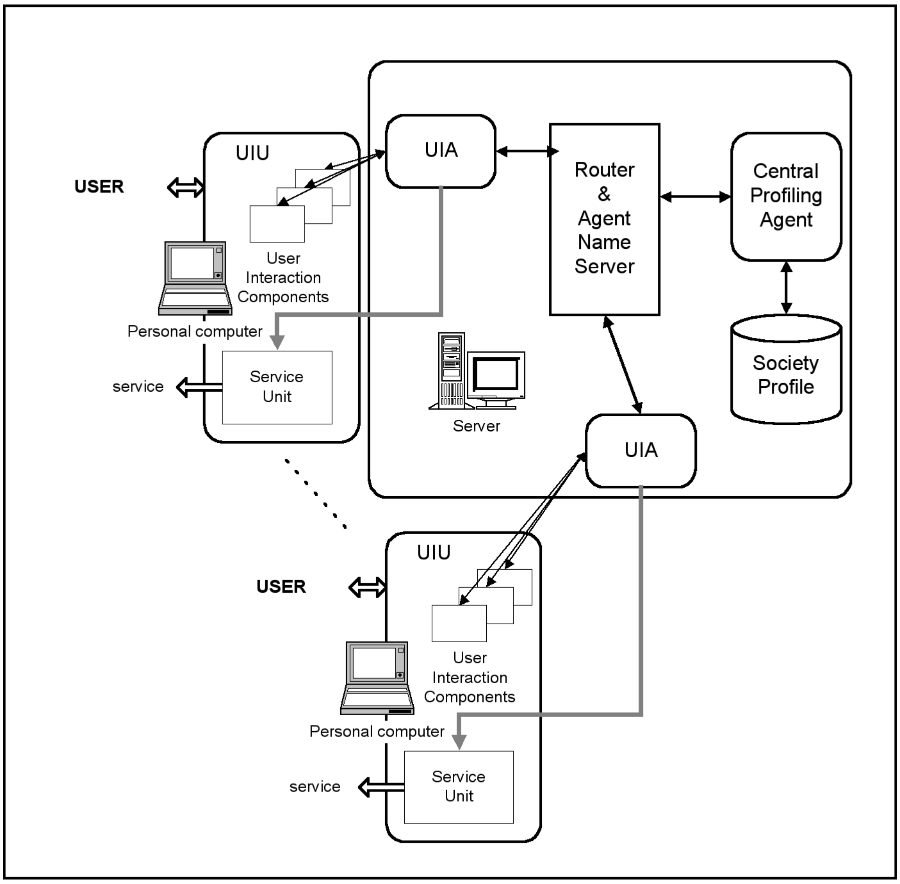

Figure 1. The overall system diagram.

The interaction partners of the agents are determined based on the users' information. Similar behaviors of users are determined and their corresponding agents are assigned as cluster-mates to allow interactions. This strategy is based on the assumption that the past behaviors may be useful for future predictions. In the proposed interactive social agent model the interaction partner selection is initially implemented based on the user similarities. Then the agents are allowed to change their interaction partners based on their users' demands.

The CA forms clusters of users by using information from all users via their UIAs. This information type should be determined previously. 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. In our application system, answers for some special questions are used to determine user similarities. This information should be much enough to determine the clusters of the users and little enough to avoid burdening the user. The CA takes this information from all of the agents and processes it by using clustering techniques. After forming user clusters, the CA can inform each agent about its cluster-mate agents that can be identified as friendly information providers. According to this information, the agents can have beliefs of the agents having similar preferences. The cluster-mate agents' information is used by the agents and may be classified as satisfactory or not by the user. According to feedback, each UIA can request for a change in the cluster structure and it sends this request to the CA. The agents may also interact randomly in the system. Sometimes results of this random interaction process is more satisfactory then the estimated beliefs. The cluster structure of the users is dynamic and UIAs get updates dynamically.

Users can get the service from any service computer in the system. Although a user is not currently connected to the system, its corresponding UIA is alive to help the others about formerly generated service information of its own and it can get real updates on society information continuously. Therefore UIAs live on the server machines. System components on the server part shown in Figure1 can be distributed in the network.

3. The Interactive Social Agent Model

In our proposed social agent model [13], each agent has to serve according to the own user's profile and decide autonomously on 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 which may change over time. Therefore the agent changes its beliefs 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 with similar to human behavior. This approach is used to simulate real interaction dynamics among persons in a society. The interactions among agents are firstly random. Influence constants are also random and determined by the system. Therefore the modeled system is very close to the real world. Each agent can be informed by the central unit about general society concepts.

It is necessary to assign users into clusters, to serve them effectively. Social classes of users (and the general structure of the system appropriately) can also change over time. Therefore the problem turns into a problem with dynamically changing interactions. Service quality based on the gained knowledge of each agent should satisfy corresponding 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 take. This choice depends on the user's profile, collected information from other agents and feedback taken from the user. The Interactive Social Agent Model can be seen in Figure 2.

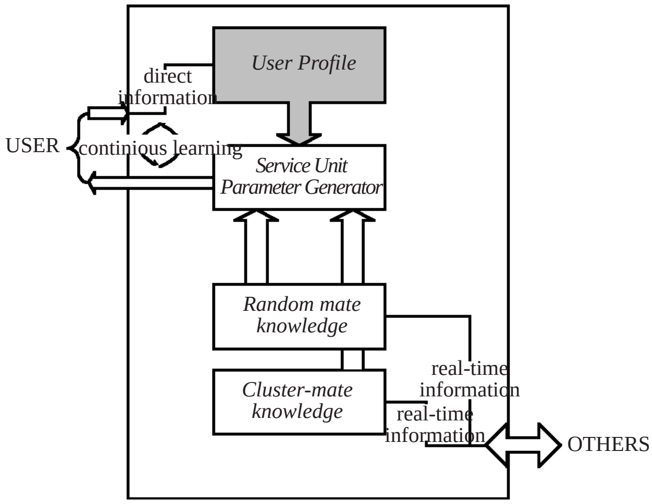

Figure 2. The interactive social agent model.

Information needed for service can be from four different types of sources:

- The user (directly formed information)
- The information from an external human expert (directly formed information)
- Other agents' information from the same cluster (real-time information)
- Any other agent information (real-time information)

Each agent generates its user's profile and the beliefs based on the incoming information from these different sources. Direct information is provided by the user or a human expert. According to the application domain, this information consists of parameters for different subjects. Beliefs are updated based on the information from cluster-mate agents or the information from others if necessary and the corresponding user feedback.

It is useful to use the direct information to provide service for users. However to collect all the information from the user is almost impossible. In most application domains, users desire the best service with little information. Therefore a real social interaction model is required in a multi user domain. Agents should use previously gained information and beliefs to choose suitable information sources. Service is given based on the chosen source. By using feedback for the service, agents' beliefs are updated. According to the chosen source and taken feedback, society general cluster structure may need to be changed.

Each agent forms its beliefs after determining the user's social cluster membership information initially. This information is given by the CA. According to this information the agent learns the cluster-mate agents and rely upon them to get help for unknown services for the corresponding user. There are n agents in the multi agent system. If the total number of cluster-mate agents who has provided the necessary information for the required service is m, an agent's newly formed knowledge, K_i, is conceptually determined by Eq.1. In this equation K_l is the gained knowledge based on the information from the 1^{th} cluster-mate agent and the user's feedback.

$$K_i = 1/m * \sum_{l=1}^{m} K_l \tag{1}$$

Agent beliefs can be changed based on taken feedback from the user for the cluster-mate solutions. An agent can change its interaction mates and choose other random agents to get requested information. Beliefs about the others are formed based on the information from others, the influence constant and the user satisfaction level. The influence constant ensures 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 adopted from the real life as a human personal characteristic property. Eq.2 determines the ith agent's newly formed knowledge to provide the required service. c_i, is the influence constant of the agent $(0 \leq c_i \leq 1)$. K_i^+, is knowledge of the ith agent gained by information from the cluster-mate agents. K_j is the knowledge about the taken information from the jth agent. This jth agent, which is not from the same cluster, is randomly chosen from the environment.

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

In this equation, if $c_i = 0$, the agent is not affected by the random information. Therefore to increase the satisfaction level of the user, direct information should be provided from either the user or an external expert. If $c_i = 1$, knowledge is affected by the random information. Therefore the cluster-mate agents' information and the random information are considered equally. 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 agents as in real societies.

3.2. Continuous learning model

In the standard profiling phase, information related to previously determined questions is taken from all of the users via their agents. This information is processed to determine the user clusters in the society. After finding the clusters of the society, the CA can send the cluster-mate agent information to all agents. This information is used by each agent to get new services for which information is not directly provided by the own user. The cluster-mate agent's information can be useful and the user may be satisfied with the service. Then the cluster-mate information can be used for further processes. 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 taken from interactions, and it is presented to the user. User satisfaction is also examined after this interaction. If the user satisfaction level is good enough, a new clustering structure may be needed, because of the exception in similar demands from different clusters. The agent beliefs are updated and a request for changing cluster structure of the society is sent to the CA.

If the user is not satisfied with all 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, 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 to the other agents.
- Static information from human experts can be presented to the user initially. After this process, user satisfaction is examined.

- Interaction is implemented with different users who have the requested solution from the cluster-mate agent. After this process user satisfaction is examined.
- If information taken by interaction from the cluster-mate agent does not provide a satisfactory solution, random agent interactions may take form between agents having the required service information. After this interaction, the cluster structure of the society may need to be changed.

4. The Central Profiling Agent

The Central Profiling Agent (CA) collects information from all users and generates society profile. The CA can also implement a learning process to determine dominant parameters of the society. The society profile (ie. clusters of users) is generated by using all of the individual user parameters. The learning process is implemented by getting feedback from the users by means of UIAs for the randomly generated service parameters by the CA. Processes of the CA and interactions with the environment are modeled in Figure 3.

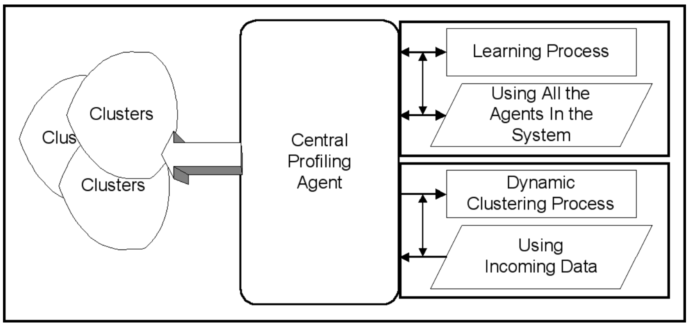

Figure 3. The central profiling agent processes and its interaction with the environment.

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

After generating the society clusters, individual agents may send requests messages for changes on the cluster structure. Because each UIA interacts with its user and examines the satisfaction levels, it 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 (ie. from outside of the cluster). Therefore the clustering structure may be changed according to the provided information from individual agents. The individual agents act based on their beliefs. However, the CA decides if the all clustering structure should be changed or not. This is determined by the coming requests from many agents.

The CA can determine some application parameters, general concepts about society in the learning process if necessary. It can implement this process by using a reinforcement learning method. The Q-learning algorithm [14] is a kind of reinforcement learning process, suitable for this objective providing successful results. Each UIA acts as a critic for the learning process. The CA chooses a random critic (ie. agent) at a time. This randomness makes the environment non-deterministic. Different users may assign different feedback for same states. The CA has to run the learning process to determine the dominance degrees of the service parameters. According to the application domain, to overcome the long learning time of the CA some state simplifications should be made.

The dynamic clustering approach called incremental clustering by Fazli [15] can be adopted for clustering processes. However performance of this process is highly dependent of the data input order. Since this is still an open question, implementing a dynamic process for clustering may result in different cluster structures for different orderings of the user data. To avoid uncertainties, the clustering process is implemented on some certain number of user input in our system. Although this approach is suitable for small number (ie. hundreds) of users, it is not an efficient solution when the number of users increases (ie. thousands). More effective solutions should be found for implementing the clustering process dynamically independent from order dependence.

There is another difficulty with the clustering algorithm choice. Different clustering algorithms can produce different clustering structures from the same given data [9]. Therefore according to the application, a suitable algorithm should be chosen. An adaptive mechanism choosing convenient clustering algorithm according to the application domain can be added to the system.

5. Case Study: Facial Expression Generation Domain for Personal Story Telling Agents

Analysis of facial expressions as an important tool of human communication is one of the most attractive fields of social sciences. It is also important for computational efforts. Facial expressions play an important role for nonverbal communication [16]. Nonverbal communication for autonomous characters such as synthetic agents, virtual actors and avatars is as important as text based communication. Some environments consisting of real and artificial agents or characters may require nonverbal communication by facial expressions. Facial expressions for artificial characters are especially very important for natural and believable visual interaction. This work necessitates social analysis on facial expressions. Some researchers in the fields of computer graphics and pattern recognition have proposed many works analyzing and synthesizing facial image sequences [17], talking facial display generation by analyzing visemes [18], and facial expression analysis by using Ekman [11] and Fiersen's FACS coding system [19]. Pixar Animation Studios animators work carefully on the facial expression generation of artificial characters, and this part is the most important and the final phase of the character generation process [20]. Maes' work on the effects of personification of agents [21] discusses the importance of facial representations of agents in a poker game. The results of this work show that while the caricature character of a dog is more likable than the other representations, the realistic agent's facial expressions are more tractable, and this agent is considered as more intelligent than the other subjects. Some applications statically determine some facial expressions such as, happiness, sadness, fear, surprise, and anger. Rea and Gandalf [22] as conversational characters interact with users by presenting some facial expressions. A sociable humanoid robot Kismet [23] communicates with people by using sensory information and represents some facial expressions related to her emotions. Such a robot, Feelix, has static facial expression information [24], [25].

General characteristics are determined statically in design time to produce facial expressions for the systems consisting of artificial characters in previous works. User preferences are not considered in designing user specific applications such as news or story telling characters, advertising agents, and assistant programs with graphical facial interfaces. These services can be improved by generating facial expressions according to the information taken from users. The way that they are served may be determined by the user's personal preferences. However if the provided service is similar, assigning systems working individually is not an effective solution. Because requesting all of the facial expressions from the user is almost impossible. This service should be provided by grouping users according to their preferences or requests (some additional properties can be used as parameters to form groups). This is ensured by clustering users with similar preferences. Thus, people belonging to the same group may be served similarly. Our proposed general model is capable of filling these gaps and effectively serving to the users.

In our application domain [26], the service generated for each user contains the facial expressions that users like. Initially to form user clusters, some facial expressions are generated by users for previously determined questions. These questions may be determined by pedagogical experts or may be sentences from a story or previously determined special situations (ie., some emotional expressions determined by pedagogues). A sample question is like: "Could you please draw me a picture of a child punished for cheating." This question is asked to all of the users. Each user draws his/her facial expression according to this situation. The facial drawings are processed as parameter arrays [17], [27]. By the help of the parametric model, facial expressions can be represented by a small number of parameters.

A collection of facial parameters for many situations is collected by CA. According to this information society profile is generated, and the users' drawings are clustered. Therefore, the facial expression analysis results show the similar expressions and the exceptions according to the clusters. By using the cluster structure, the cluster-mate users are determined. Each agent serves to its user by the help of this cluster-mate agents' information.

A similar application system is proposed for document filtering by David and Kraus [28]. The aim of that system is to produce multimedia programs according to the integrated user profiles. However this work does not propose a general architecture and it is highly domain specific.

To implement such a system, an agent construction and communication tool is needed. Because we implemented our own interactive social agent model, there is no need that the chosen tool supports the social behavior of agents, so we used JATLite [29]. The agent interactions are implemented by using communication layer of JATLite. 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 [30] layer ensures that the agent communication is implemented as suitable for the speech-act theory.

5.1. The user interface unit

The User Interface Unit communicates with the user. The UIU consists of the User Interface Agent (UIA), The User Interface Program (UIP), The UIA/UIP Interface, The Graphical Facial Animation Program (GFAP) and the Service Unit (SU). The UIU requires cooperation among its components to operate. The UIU components and their relationship can be seen in Figure 4.

Each user is connected to the own User Interface Agent (UIA). UIAs are implemented in the application domain by using our proposed interactive social agent model. The user can generate facial drawings by using scrollbars on the display of the UIP, a screen shot of which is shown in Section VI. Facial muscle objects related to the scrollbars request services from the UIA. Communications between the UIA and the facial muscle objects are synchronous and are implemented on different ports of the UIA. Protocol definitions are known for both sides. A UIA can get the requested behavior of a muscle object and update the related parameter value appropriately. The GFAP uses these parameters as an input according to the information taken from muscle objects. The GFAP reads these parameters within certain intervals and displays the facial expressions related to the updated parameters. Therefore the user can observe the facial drawing with the desired effects.

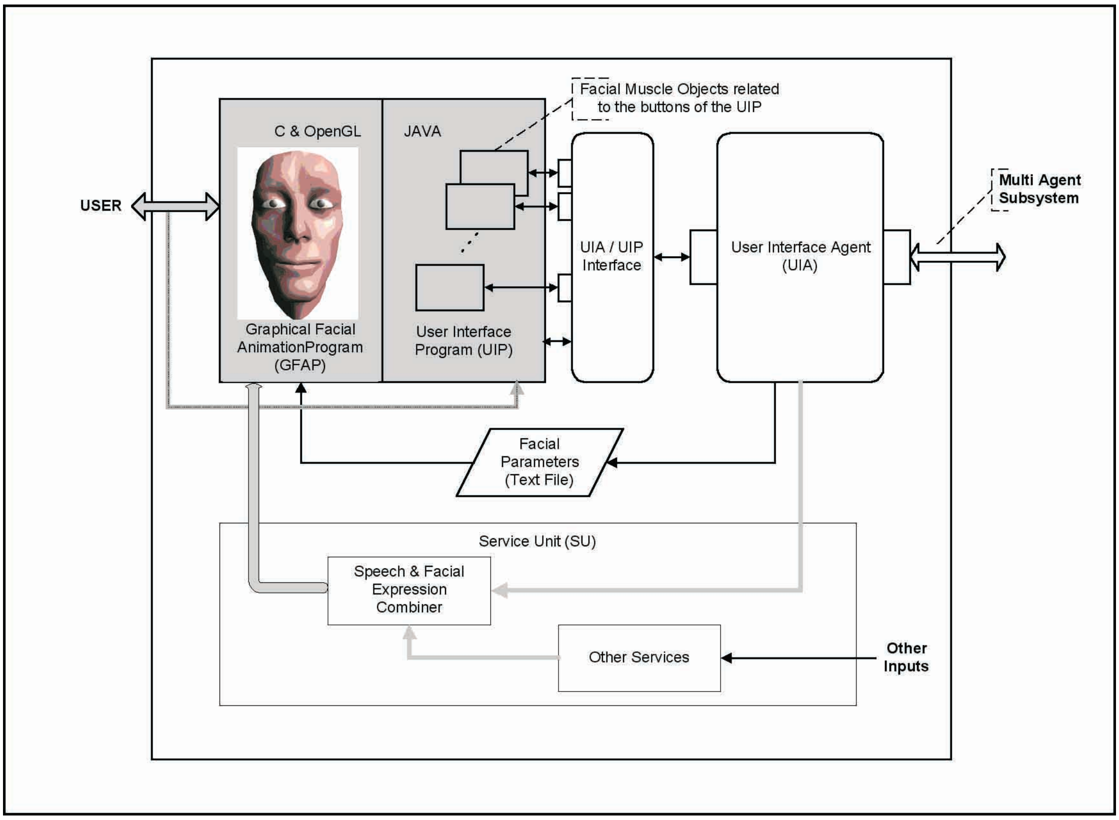

Figure 4. The User Interface Unit (UIU).

After determining some facial expressions for some situations, each UIA in the system stores its user's information. This information is sent to the CA to generate society profile. After sending user information to the CA, each UIA waits for the information related to the cluster-mate agents. After taken this information it can interact with them according to the proposed model. The information exchanged among agents is the parameter values for facial expressions.

5.2. The user interface program

The User Interface Program (UIP) performs user-agent interaction. It is, written in Java, a subclass of the User Interface Agent. It presents a window to communicate with the user at both directions to change the expression of the GFAP and get feedback. The facial muscle objects are related to the scrollbars of the UIP. These objects are responsible for the movement of certain muscles of the face. Each UIA acts as a facial updating server for these objects. Communication between each facial muscle object and the UIA is implemented through the UIA/UIP Interface.

5.3. The graphical facial animation program

The Graphical Facial Animation Program is a facial animation program representing a parameter set for a facial expression. Parametric analysis of facial animation and expressions is first proposed by Parke. Image generation is based on surfaces constructed from connected networks of polygons in the model he represented [27]. The advantage of using parameterized models is ease of changing all facial expression by only a small number of parameters. Such a work is implemented by K. Waters [17]. This is a C program using OpenGL library. In this program, 3D dynamic model of the face is represented. This face model combines a physics-based model of facial tissue with an anatomically based facial muscle control process to synthesize realistic facial motions. The parameters related to the muscles on the non-uniform triangular mesh model can be manipulated, and the facial expression of the program is changed [31]. We used a modified version of Waters program in our system as GFAP. The parameters are the information considered and exchanged in the system. There are 18 parameters related to muscle control points. They are: Left and right Zygomatic_Major, Left and right Angular_Depressor Left and right Frontalis_Inner, Left and right Frontalis_Major, Left and right Frontalis_Outer, Left and right Labi_Nasi, Left and right Inner_Labi_Nasi, Left and right Lateral_Corigator, Left and right Secondary_Frontalis. All of them are controlled by the user. A sample parameter set for the "smiley" facial expression can be seen in Figure 5.

Left_Zygomatic_Major 3.00	Right_Zygomatic_Major 3.00
Left_Angular_Depressor 0.00	Right_Angular_Depressor 0.00
Left_Frontalis_Inner 0.00	Right_Frontalis_Inner 0.00
Left_Frontalis_Major 4.00	Right_Frontalis_Major 4.00
Left_Frontalis_Outer 0.00	Right_Frontalis_Outer 0.00
Left_Labi_Nasi 0.00	Right_Labi_Nasi 0.00
Left_Inner_Labi_Nasi 0.00	Right_Inner_Labi_Nasi 0.00
Left_Lateral_Corigator 0.00	Right_Lateral_Corigator 0.00
Left_Secondary_Frontalis 0.50	Right_Secondary_Frontalis 0.50

Figure 5. A sample parameter set for the "smiley" facial expression.

5.4. The user interface agent

The User Interface Agent is modeled according to our interactive social agent model. The User Interface Agent as an autonomous entity collects its user's information and uses it for further processes. It has ability to communicate and to exchange information with other user agents in the system. This information is used for generating agent beliefs by taking feedback from users. The UIA communicates with the facial muscle objects and updates related parameters, according to the user instructions.

Each UIA plays a central role in generating the service based on the user preferences. To effectively work, each agent needs to work with the other agents in the system. According to the proposed interaction model, each agent tries to provide the best service for its user by interacting with both users and the other agents. Each UIA has an influence constant. This value determines how much this agent relies on the other agents. This constant is assigned initially to the agent.

5.5. The service unit

The Service Unit in the implemented application domain consists of mainly Speech and Facial Expression Integrator Unit and other necessary units. The speech and the facial expressions should be successfully combined to implement artificial characters reading stories. However the SU design is out of scope of this work. There are some valuable works on this process for generating both real-faced or artificial [32] story tellers. Our service unit can be implemented by using related works. Our system parameters can be arranged according to different implementations of the SU.

6. Implementation Details

6.1. Implementation of the system

All of the constructed system components were written in Java except GFAP (in C). Communication with some components of UIP and the GFAP is implemented through a text file. Communication between a UIP scrollbar object and the corresponding UIA is implemented via Java sockets.

Agent construction and communication are supported by JATLite. All of the agents are inherited from JATLite Agent class. Their social constructions are designed according to our social model. JATLite tool provides an implementation of a Router unit which implements agent registration processes and serves as a Naming Server. Each agent has a unique name in the system by agent name serving facility of the Router. Therefore agent names and addresses are accessible by any agent in the system. Although the receiver agent is off-line, messages are stored by queuing and polling service of the Router. Two agents communicate with each other either through Router or directly by learning addresses from the Router. Agents in the system should know the address of the Router. Each user (ie. agent) registers to the system with a user name and a password. Agents get informed about connected agents. The system is scalable because each agent is loosely connected to each other. Adding or leaving an agent from the system does not influence the system performance.

Agents use the KQML layer of JATLITE for inter agent communication. KQML messages are based on speech-act theory. Standard KQML primitives are used in communication. This feature provides communications among different agent organizations.

We considered two versions of the Water's work as the implementation of GFAP. The program written by Waters uses OpenGL [33] library. In the first version, texture mapping is not supported and Glut library is included. In the 2^{nd} version, texture mapping is supported, and the Glaux library is used. These are OpenGL's libraries for user interactions, and one of them should be used. We used the 2^{nd} version of the program and implemented some manipulations on this program to combine with our design. To integrate our UIP with the GFAP, timer function should have been added. The original program uses parameters changed by the user with keyboard instructions. It uses the Glaux [33] Library which does not support timer function. In our system, the parameter values are changed by using the UIP components. Finally the program was converted to use Glut [33] Library consisting timer function and supporting texture mapping. The converted program can dynamically generate facial expression with the given parameters.

6.2. Implementation of user interaction

Initially users are asked to answer to some questions. The first part of the questions contains information related to personal information such as age, sexuality, etc. Then the information to generate user clusters is taken. Because the processed information is the facial muscle parameters in our system, user replies are facial expression drawings by using UIU components. The initial drawing screen for the story telling agent domain can be seen in Figure 6. In this figure some previously determined questions are asked to the user, and the user draws related expressions by using the scrollbars. The GFAP and the UIU programs can be seen in different windows but the user can see related changes on the GFAP as soon as he/she uses the scrollbars of the UIU. The information related to facial expressions is stored as standard user information and sent to the CA for the clustering process.

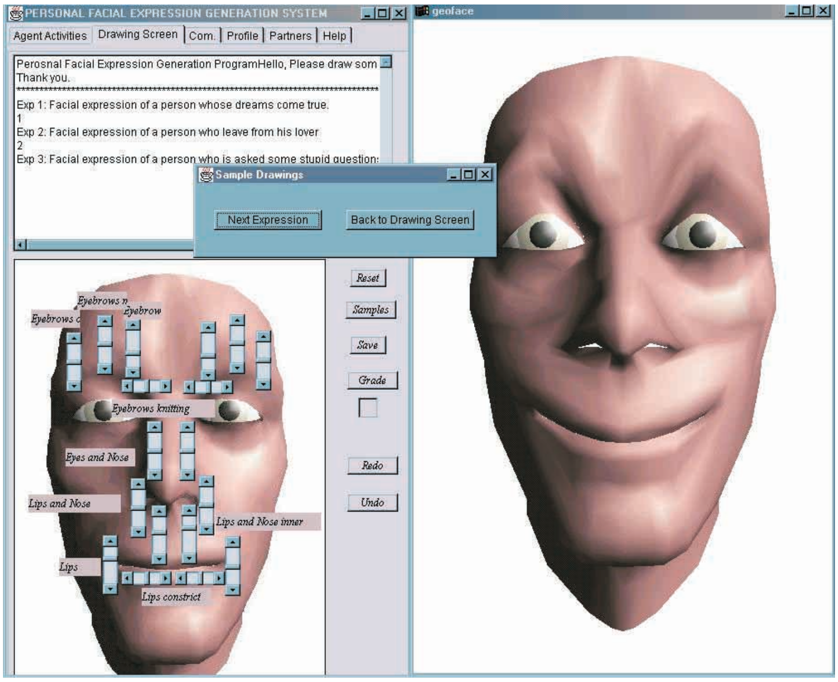

Figure 6. Initial Query Screen from the Story Telling Application Domain.

After the querying process, each user is being served continuously by the UIA in a desired manner by the user. Feedback is taken from users to increase their satisfaction levels after the services. Feedback can be in two forms:

1. Feedback for the overall service: The complete story telling process by using facial expressions of the agent is graded by the user.

2. Satisfaction query at discrete times: There are two types of initialization of requesting a feedback from the user.

   a. Initialization by the user: The system does not generate an alarm situation. The grades do not result in that calculated satisfaction level is very small. However, user may not like the provided service. In this case, the user initiates the grading session. Single facial drawings of others are shown to the user, and the user grades them. These facial drawings are taken from the cluster-mate agents initially or from the random agents if necessary according to the proposed social interactive agent model.

   b. Initialization by the system: The system generates an alarm situation. In this case, user satisfaction is detected as very small. The system does not provide a service and tries to improve the service until satisfaction level increases to a desired point.

7. The Modified Q Learning Process for Non-Deterministic Environments

The modified reinforcement learning process is implemented by the CA to determine parameter effectiveness arrays for some facial expressions. The primary objective of the learning process is to provide additional information for the effective clustering of facial expressions. The parameter (attribute) weights can be learned after the learning process. Additionally, the results can also be used for some social analyses.

In our application domain, the environment consists of all the users and their corresponding UIAs. Users act as critic in the learning process. The environment is non-deterministic, because feedback taken from users will be different for the same situations or states. The system is scalable itself. Because the CA as a learner chooses a different agent to get feedback at each step, there is not dependency on which agent criticizing because of the non-deterministic capability of the learning process.

In our application domain, the Q learning algorithm [14] is used for the learning process. This algorithm is known as an off-line policy TD control algorithm. In one-step of the Q learning algorithm, Q values of the states are updated as in Eq. 3.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\Big] \tag{3}$$

where t indicates the current step number, Q is the learned action-value function, s is the state, a is the action taken by the agent, α is the learning rate, and r is the reward. The states are the corresponding parameter effectiveness arrays in our system. Each state is also mapped to a facial drawing. There are 18 parameters on the facial model. The parameters are used for left and right muscles. To reduce the number of states, the left and right parameters are merged in the state representation. Therefore in the state representation (a binary array) there are only 9 state bits indicating the corresponding muscle parameter (left and right) is effective or not on the current facial expression. Value 1 represents effectiveness of the corresponding muscle for the facial expression. A sample state is given such as: [111011100] meaning that 4, 8 and 9^{th} muscles are non-effective on the facial expression. The learned result is the desired parameter effectiveness array. The CA holds the state graph for traversing and applying Q-learning updates on states. Hamming distance between two neighbor states is 1 in the model. The Q values of states are stored in a look-up table. While traversing the states, the Q values of states are updated. Convergence criterion is met when a state's Q value increases much more than the others. Because the environment is non-deterministic, the convergence occurs in a longer time than the deterministic case.

In each learning state, a facial expression related to a previously determined social expression is mapped to the current state and the parameters of this facial expression are sent to a random UIA. The UIA presents the facial expression to its user and gets a grade for that expression. The CA takes this grade as feedback for the learning process. 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 Figure 7.

Clear for all table entries,
While a convergence criteria is not met
    Take the action
    Generate the facial expression 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 parameters related to the facial expression to a random agent
    Wait for the grade
    ..
    Calculate the reward as a function of the belief of the agent and the taken grade
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\Big]$
    Find the new state s' and action

Figure 7. The modified Q learning algorithm.

In the learning process, the CA visits a state for each parameter mapping step. The neighbors are chosen from nine 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. The belief of the agent indicates a change on the feedback between neighbor states as a boolean number. While going from one state to another, the related parameter (the different one) changes in the mapped facial drawing. If the subsequent feedbacks 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. Figure 8 demonstrates a sample reward calculation while going from one state to its neighbor.

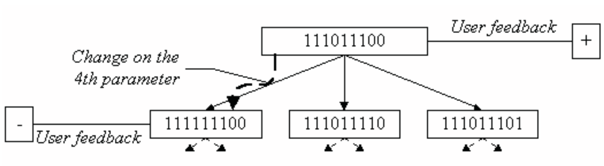

Figure 8. State transition in the learning process (Positive Reward).

In the sample scenario, the change is on the 4^{th} parameter for the first transition. The corresponding parameter value in the first facial expression will be different from the next one. Feedback taken at the first step from one user is +. Prediction indicates change on the feedback. Feedback at the next step is -. Therefore a positive reward is calculated, because the prediction is consistent with the taken feedback. Table 1 shows the rewards for all the possible situations.

Table 1. Calculation of the reward as a function of the grade and the belief of the agent.

Change on the grade	Prediction: "change"	Prediction: "no change"
+→-	100	0
-→+	100	0
+→+	0	100
-→-	0	100

8. The Clustering Process

The clustering process uses all users' information. Incoming information from each user arrives in the CA at different time steps. The clustering process is implemented for certain number of data samples. In the facial expression analysis domain, each facial parameter in the array is considered as one dimension in a multi-dimensional space. Each parameter scale, determined in the facial model, is different from each other. Therefore normalization is required to cluster data samples. The feature reduction may be implemented by using learning process outputs. The distance calculation is done by using Euclidian distance.

Different clustering algorithms produce different cluster outputs for the same input data [9]. Cluster analysis on the given data may be implemented by using different algorithms for different domains. We analyzed Fuzzy c-means (FCM) [34], Complete-link (Clink) and Single-link (Slink) [9] algorithms to be implemented on the sample data set. The CA produces three different types of outputs for these algorithms. According to the application, one of these outputs can be chosen. All of the three algorithms are implemented in the system to produce clusters of parameter arrays for specific facial expressions. After clustering the facial expressions for different situations, users are clustered, the cluster-mate agents are indicated and corresponding UIAs are informed.

FCM Algorithm

The FCM algorithm is a more efficient version of the k-means [9] algorithm with a fuzzy membership approach. Outputs indicate fuzzy membership values of the data samples for the clusters. The de-fuzzification process is applied to convert the output to a crisp type. The FCM algorithm generates clusters on a given data set. However cluster validity should be checked and performance for different number of clusters should be examined and the best cluster number should be chosen. We have used partition coefficient criteria [35] with the FCM algorithm to test validity of the clustering. The partition coefficient criterion considers the membership values of the samples.

Clink and Slink Hierarchical Algorithms

Complete-link (Clink) and Single-link (Slink) algorithms are hierarchical clustering algorithms producing dendograms as outputs. The Clink and the Slink algorithms start clustering with all data samples are in their own clusters and continue to iterate by merging the most convenient two clusters at each step. When all of the data samples are collected in one cluster, the algorithm stops. The generated dendograms show relations of the data samples in the clusters at each step. To find the optimum number of clusters, the final dendogram is cut at a desired level. The difference between the Clink and the Slink algorithms is on the distance calculation between clusters.

9. Experimental Results

The designed system for the application domain was tested in a small group of people at ages of 20. The users were requested to draw facial expressions related to 10 questions. A sample question asked to users is: "Draw me a picture of a person whose dreams come true." A sample set of 10 user drawings are shown in Figure 9.

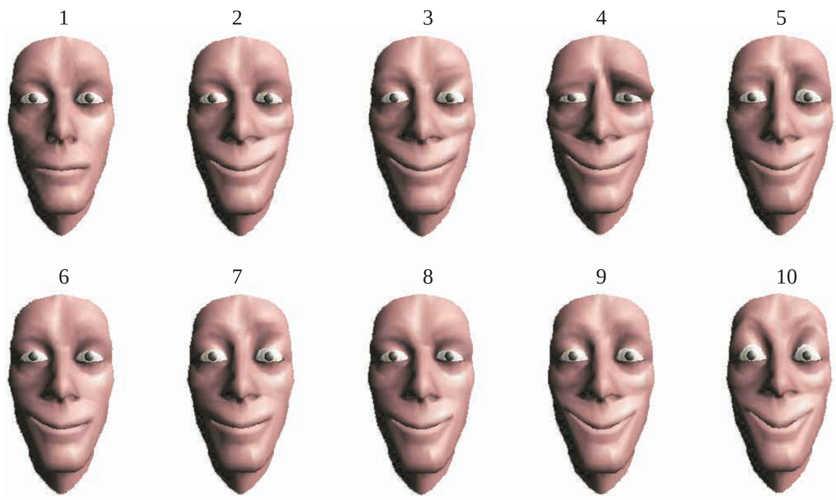

Figure 9. A sample result set for 'the expression of a person whose dreams come true'.

The three clustering algorithms were implemented on this small data set. Four dimensions were chosen (the parameter effectiveness array values are manually entered) for the clustering process. Left and right lips and the eye knitting muscles parameters are activated. The FCM algorithm generates five optimum clusters. The Clink algorithm generates two clusters and the Slink algorithm generates three clusters. The results of the algorithms are shown in Table 2. Each column in the table represents the user numbers from Figure 9. The table entries show the assigned cluster numbers of the users. The exceptional situation for the sample question can be seen in the 1^{st} picture of Figure 9. The data sample for this picture is isolated from other clusters in each of the three algorithm results.

Table 2. The clustering results for the three algorithms.

User No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
FCM	1	5	4	4	5	2	2	3	5	5
Clink	1	2	2	2	2	2	2	2	2	2
Slink	1	3	3	3	3	3	3	2	3	3

The learning process is implemented in a simulation. The simulation results can be seen in Figure 10. In this figure, convergence steps for each state can be found. Because different states have different bit representations the convergence steps are different.

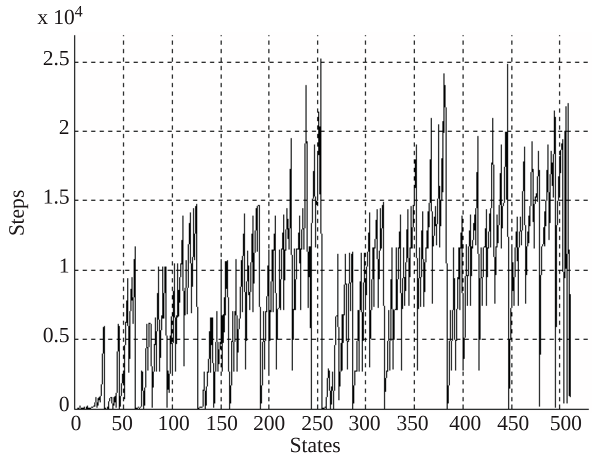

Figure 10. Learning process simulation results for each state.

10. Conclusion

In this study, a general multi-agent system model for systems serving to users according to their demands is presented. To model the specified society, the samples directly come from the society itself. The proposed interactive social agent model presents a model for providing the best possible service by interacting with either the own user or the other agents in the system. User preference is mainly considered to provide the appropriate services for the user. In the model, agents are allowed to communicate with each other to provide social information exchange. The interaction partners of the agents are changed dynamically to serve user's demands. Therefore the service for one user is determined in real-time based on the interaction information by the corresponding agent. A central unit accessed by all agents processes user group information to provide effective interactions among them for high quality services. Learning specific details of each user profile is a time and resource consuming process, so the system is modeled as analyzing society information in a multi user domain. The learning process is implemented on a distributed user group in a scalable manner. The reason for using distributed data in the learning process of the central unit is to model all of the society and to ensure just one critic per user for determining parameter weights. The non-determinist property of the environment is coped with the Q learning based algorithm in the learning process.

The influence constant in the social interactive agent model 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. It may be needed to multiply the number of Routers in the system to represent different societies. These societies may be formed by using information about sexuality, age, etc. Therefore differences among groups may be determined. Although grouping process reduces computational and storage requirements, a huge data set should be considered. The overall system model is suitable for systems providing services according to the users' specific demands. The SU may be any kind of unit working for the user. The system generates necessary parameters for this unit.

The application domain is chosen to produce facial expressions for personal story teller agents. The difficulty on measuring performance of the implemented system for the application is there are no test sets for graphical facial expressions. The system should be tested on the real world (e.g. on the students in a primary school). This work requires an inter-disciplinary team. The questions asked to users can be modified adaptively according to the groups of users having similar profiles. Users may be allowed to grade questions to form an effective question set. The graphical interface should be friendlier. Different face textures may be used for groups of different ages or sexuality. Then the results can be used for pedagogical experiments. The system may be used for diagnosis of some disabilities with the help of the other social tests prepared by pedagogues. Facial muscle objects may be modeled as intelligent agents to form facial expressions. All muscle objects are reactive, and they communicate with UIA through different ports in the original implementation. Different parts of the same face may live on different hosts and may be controllable by different users. The social results of the application system can be used for some pedagogical applications or for determining some cultural differences statistically among nations or groups of people (different age, sexuality, etc.) by analyzing the results in an interdisciplinary work. Our application system can be used for generating not only the facial expressions but also the body language having gestures different for many cultures. In this case, the service output will be 3D sequence arrays as animation. The parameters can easily be arranged for this service in our application system.

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

References

[1] M. Wooldridge and N.R. Jennings, "Intelligent Agents: Theory and Practice.", The Knowledge Engineering Review, Vol. 10, No.2, pp. 115-152, 1995.

[2] J. P. Müller, "The Right Agent (Architecture) to do the Right Thing", In: Intelligent Agents V, Jörg P. Müller, Munindar P. Singh, and Anand S. Rao (eds.), pp. 211 - 226, 1999.

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

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

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

[6] Active Recommendation System for the Library without Walls: http://www.c3.lanl.gov/∼rocha/lww/

[7] G. Weiss, Multi Agent Systems: A Modern Approach to Distributed Artificial Intelligence, Editör: G. Weiss, England: The MIT Press, 1999.

[8] P. Maes, “Agents that Reduce Work and Information Overload”, Communications of the ACM, ACM Press, Vol. 37, No.7, pp. 31-40, 1994.

[9] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data Clustering: A Review,” ACM Computing Surveys, Vol. 31, No.3, 1999.

[10] C. Darwin, “The Expression of the Emotions in Man and Animals”, D. Appleton, New York, 1872.

[11] P. Ekman, “Chapter 16. Facial Expressions”, In T. Dalgleish & M. Power, Handbook of Cognition and Emotion, New York: John Wiley & Sons Ltd, 1999.

[12] S. Sariel and B.T. Akgun, “A Multi Agent System Analyzing Parameters Related to Facial Expressions”, IASTED International Conference, AI 2002, International Symposium on Artificial Intelligence and Applications, Innsburck, Austria, pp. 302-307, 2002.

[13] S. Sariel, B.T. Akgun, “Intelligent Agents Serving Based on the Society Information”, The Eighth IEEE Symposium on Computers and Communications (ISCC), pp.763-768, 2003.

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

[15] C. Fazli, “Incremental Clustering for Dynamic Information Processing”, ACM Transactions on Information Systems, Vol.11, No. 2, 1993.

[16] M. Luck, and R. Aylett, “Applying Artificial Intelligence to Virtual Reality: Intelligent Virtual Environments”, Applied Artificial Intelligence Vol. 14, pp. 3-32, 2000.

[17] D. Terzopoulos and K. Waters, “Analysis and Synthesis of Facial Image Sequences Using Physical and Anatomical Models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.15 No.6, pp. 569-579, 1993.

[18] T. Ezzat, and T. Poggio, “Mike Talk: A Talking Facial Display Based on Morphing Visemes”, Proc. Computer Animation Conference Philadelphia, Pennsylvania, pp. 96-102, 1998.

[19] Y. Tian, and J. Cohn, “Recognizing Action Units for Facial Expression Analysis”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.23 No.2, pp. 197-114, 2001.

[20] SIGGRAPH 97 Panel on Facial Animation: http://www.cs.toronto.edu/∼siggraph97-panel/

[21] T. Koda and P. Maes, “Agents with Faces: The Effects of Personification of Agents”, Proc. HCI’96, London, UK, pp. 98-103, 1996.

[22] Rea and Gandalf: http://gn.www.media.mit.edu/groups/gn/

[23] C. Breazeal, A. Edsinger, P. Fitzpatrick, and B. Scassellati, “Active Vision for Sociable Robots”, IEEE Transactions on Systems, Man and Cybernetics- Part A : Systems, and humans, Vol.31, No.5 2001.

[24] L. D. Cañamero, and J. Fredslund, "How Does It Feel: Emotional Interaction with a Humanoid LEGO Robot", Socially Intelligent Agents: The Human in the Loop, Papers from the 2000 AAAI Symposium Technical Report FS-00-04, pp. 23-28, 2000.

[25] L. Cañamero, J. Fredslund, "I show you how I like you –Can you read it in my face?", IEEE Transactions on Systems, Man and Cybernetics- Part A : Systems, and humans, Vol .31. No.5, 2001.

[26] S. Sariel, B.T. Akgun, "A Personal Facial Expression Generation System with Interacting Agents", Technologies for the Interactive Digital Story Telling and Entertainment (TIDSE) Conference Darmstadt, Germany, Proceedings, Computer Graphik Ed, pp. 82-93, 2003.

[27] F.I. Parke, "Parametrized Models for Facial Animation", IEEE Computer Graphics and Applications Vol.2, No.9, pp. 61-68, 1982.

[28] E. David and S. Kraus, "Agents for Information Broadcasting", Intelligent Agents, 6^{th} International Workshop, ATAL'99, USA, 1999.

[29] JATLITE: http://java.stanford.edu

[30] T. Finin and R. Fritzon, "KQML- A Language and Protocol for Knowledge and Information Exchange", Proceedings of 19^{th} Intl. DAI WorkShop, pp. 127-136, 1997.

[31] GFAP: http://www.crl.research.digital.com/ publications/books/waters/Appendix1/appendix1.html

[32] B. Uz, U. Güdükbay, and B. Ozguc, "Realistic Speech Animation of Synthetic Faces", Computer Animation'98, IEEE Computer Society Publications, Philadelphia, 1998.

[33] OpenGL, Glut and Glaux Libraries: http://www.opengl.org

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

[35] T. Ross, "Fuzzy Logic with Engineering Applications", McGraw-Hill Companies Inc, 1995.