A Personal Facial Expression Generation System with Interacting Agents

Sanem Sariel[1], Tevfik Akgun[2]

[1] Istanbul Technical University, Electrical and Electronics Faculty,
Department of Computer Engineering, Istanbul, TURKEY
{sariel@cs.itu.edu.tr}

[2] Yildiz Technical University, Faculty of Art and Design,
Communication Design Department , Istanbul, TURKEY
{akgunbt@yildiz.edu.tr}

Abstract. In this paper, we propose a personal facial expression generation system consisting interacting agents in a multi user domain. The agents consider user profiles and similar users' information while serving the users. By clustering the users' information, the users having similar profiles can be determined. After determining clusters, the agents interact with each other according to the similarity information for an effective work. They try to increase the satisfaction levels of their users by continuous learning processes. Therefore facial expression generation is done according to the users' demands and this process is implemented by a corporation among agents. The system can easily be converted to a personal story telling system.

1 Introduction

Facial expression analysis as one of the most attractive fields of social sciences is very important for creating artificial characters. The initial works were implemented by Darwin on facial expressions [1]. There are still many arguments on facial expression generation processes of the human brain [2]. Facial expressions should be carefully generated for natural and believable visual interaction for artificial characters. This work necessitates social analysis on facial expressions. Some researchers in the fields of computer graphics and pattern recognition have proposed many works ([3],[4],[5],[6],[7],[8],[9]). Some applications statically determine some facial expressions such as, happiness, sadness, fear, surprise, and anger. Rea and Gandalf [10] as conversational characters interacts with the user presenting some facial expressions, but their facial expressions are not specialized for users. Sociable humanoid robots like Kismet [11] and Feelix [12] communicate with people by using sensory information and represent some facial expressions related to their emotions.

The general characteristics were determined statically in design time to produce facial expressions for the systems consisting of artificial characters in previous works. User preferences were 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). The users acting similar behaviors are considered. This is ensured by clustering users with similar preferences. Thus, people belonging to the same group may be served similarly.

We proposed a personal facial expression generation system consisting intelligent agents interacting with each other and considering user profiles. Our proposed system's main objective is producing facial expressions according to the user's demands effectively. In the system, facial expressions are generated by users for some previously determined questions. These questions may be determined by pedagogical experts or they may be sentences from a story or previously determined special situations Each user draws his/her facial expression according to the situations. The facial drawings are processed as parameter arrays ([3],[13]). A collection of facial drawings for many situations is collected. According to this information, society profile is generated, and the users' drawings are clustered. The interaction partners of agents are determined according to these clusters. Each agent serves to its user by the help of this interaction partner agents' information. After the provided service to the user, the user's satisfaction is examined. The profiling task of each agent is continuous and adaptive during the system life-cycle. The agents can request for a change in the cluster information of the society according to the interaction information and the user satisfaction levels. The requests from different individual agents are considered by the central unit. Therefore the system continuously improves itself.

2 The Personal Facial Expression Generator System

Our proposed personal facial expression generator system is composed of interacting user interface agents. Each user interface agent is assigned to a user and responsible to serve according to its user's demands. The service is provided to the user by using information from either the own user or the other agents. Society and group similarities are used by the agents in some cases. Therefore the total workload for individual agents is reduced, and an effective and interactive personal facial expression generation process is implemented. The society groups are formed by clustering techniques. Some special parameter values are learned by using reinforcement learning method.

In the system, each User Interface Agent (UIA) is connected to the related User Interface Unit (UIU). The overall system diagram can be seen in Fig.1. The UIU implements user interaction processes and also contains the Service Unit (SU). The SU is the facial expression generator part of the system for each user. After making some decisions, each UIA provides necessary parameters for the SU to produce facial expressions. The Clustering Agent (CA) implements the clustering and learning processes, and acts as a society advisor having information about users' similarities in the form of clusters. The JATLite[14] Router is responsible for implementing agent communications and having information about agents' addresses and names as a naming server.

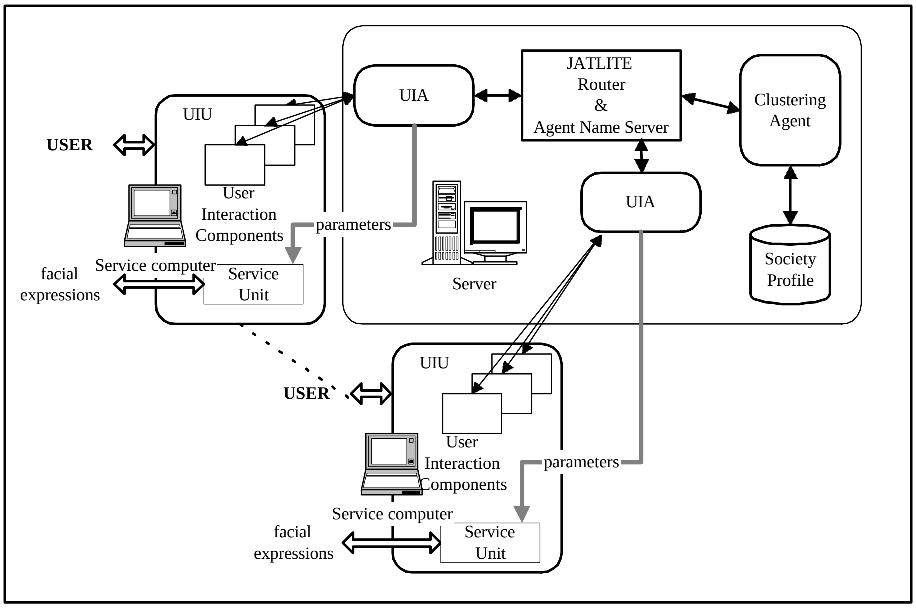

Fig. 1. The Personal Facial Expression Generator System

2.1 The User Interface Unit

The UIU consists of the UIA, The User Interface Program (UIP), The UIA/UIP Interface, The Graphical Facial Animation Program (GFAP) and the Service Unit. The UIU requires cooperation among its components to operate. The UIU components and their relationship can be seen in Fig. 2.

Each UIA requests from its user for drawing facial expressions related to some situations. The user generates these facial drawings by using scrollbars on the display of the UIP, a screen view of whom will be shown in Section 2.2. Facial muscle objects related to the scrollbars request service from the UIA. The communications between the UIA and the facial muscle objects are synchronous and are implemented on different ports of the UIA. UIA serves for different behaviors of facial muscle objects. Protocol definitions are known for both sides. The UIA gets the requested behavior and updates parameters related to the changed value. 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. The other user's drawings can also be shown to the user. The coming parameter values are sent by the UIA for generating facial drawings on the GFAP screen.

The User Interface Program, performs user-agent interaction. It is, written in Java, a subclass of the User Interface Agent. It generates a window to communicate with the user at both directions. Then the user is able to change the expression of the GFAP. The user's feedback is also taken via this program. 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. The User Interface Agent acts as a facial updating server for these objects. The communication between each facial muscle object and the UIA is implemented through the UIA/UIP Interface.

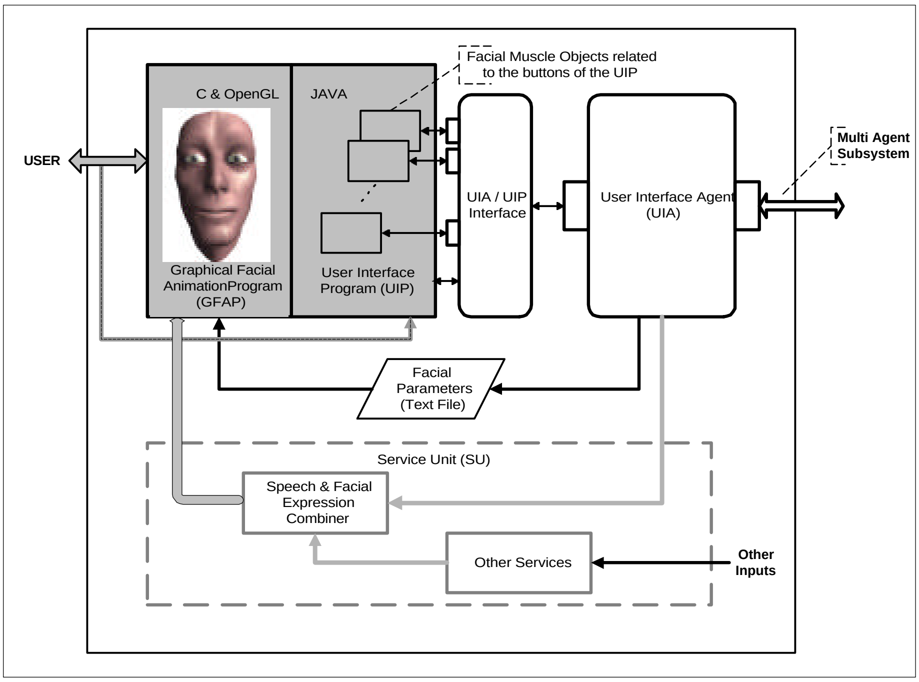

Fig. 2. The User Interface Unit(UIU)

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 was first proposed by Parke. Image generation is based on surfaces constructed from connected networks of polygons in the model he presented [13]. The advantage of using parameterized models is changing all of a facial expression by using only parameters. Such a work was implemented by K. Waters [15] This is a C program using OpenGL. 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 [3]. We used the implementation of Waters in our system. The parameters are the information considered and exchanged during the system work in our 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.

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. The UIA outputs the parameters of which the GFAP uses as an input to generate the facial drawings of the user.

The Service Unit consists of mainly Speech and Facial Expression Combiner Unit and other necessary units. The SU implemented in our work generates facial expressions according to user's demands. We mainly focused on determining personal parameters for facial expressions. The speech and the facial expressions should be successfully combined to implement artificial characters reading stories. There are some valuable works for this process for generating real-faced [4] or artificial [16] story tellers. The Uz's [16] tag and facial expression combining approach can be used in our system.

2.2 Service Generation and User Interaction Phases

In the user interaction process, firstly the users similarities should be determined. To implement this process, the user is first asked to answer for some questions. Static user information (age, sexuality, etc.) is firstly taken. After that, the information about some facial expressions to form the user profile is taken. Because the processed information is facial muscle parameters in our system, the user answers are in the form of facial expression drawings by using UIU components. The initial query screen can be seen in Fig. 3. 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. These taken information is stored as standard user information and sent to the CA for the clustering process. After this querying process, the user is being served continuously as he/she wants the desired service. While serving the user, the feedback is also taken from him/her to increase the satisfaction level. In the continuous learning phase, the CA knows which users have similar profiles. This information is sent to the individual agents. Therefore every agent is being informed about its interaction partners. The agents can get some desired information from these interaction partners.

During the system work agents need to present the user facial drawings for some expressions not determined previously. To implement this process as an intelligent unit, the agent tries to find a drawing without any request from the own user. The drawing found as a parameter set should satisfy the user's demands. After taking information from interaction partners about facial parameters, the facial drawings are presented to the user, and the user's satisfaction level is examined. 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 service based on the information comes from interactions, and the information 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 agents beliefs are updated and a request for changing cluster schema of the society 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. The requests from different individual agents are considered by the CA. Therefore the system continuously improves itself.

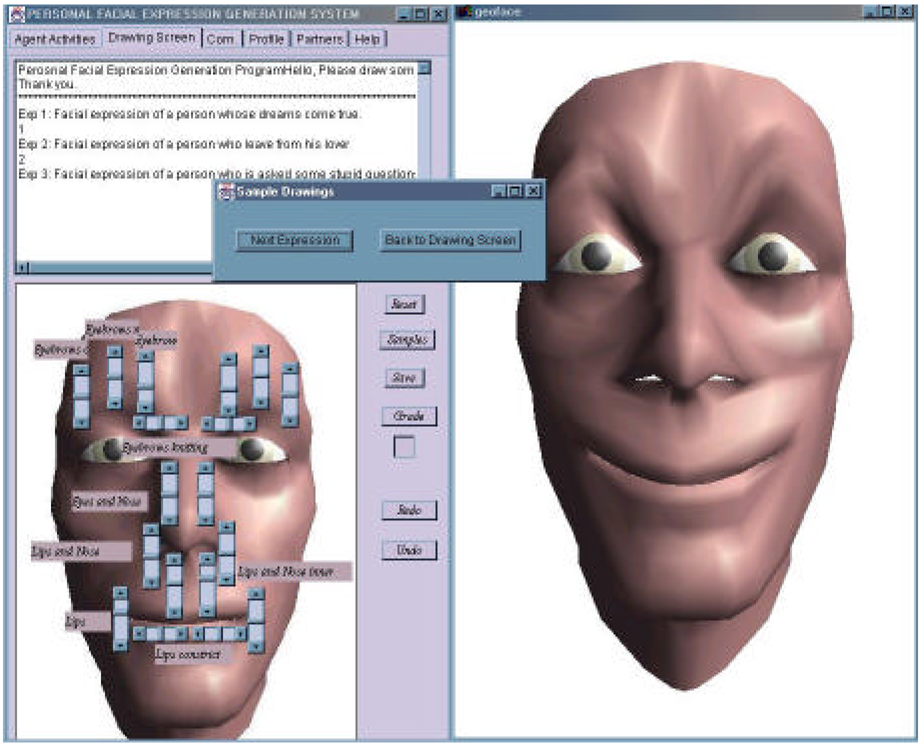

Fig. 3. Initial Query Screen "From The Story Telling Application Domain

If the user is not satisfied with the all of the service provided by the agent based on the formed knowledge or there is no sufficient information from the other agents in the system for the special facial expression, 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:

- A 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 between agents having the required service information. In this process the cluster contents can be changed, and the users representations in the clusters can be changed.

3 Implementation Details

All of the constructed system components were written in Java. However The GFAP was written in C. The communication with some components of UIP with GFAP is implemented through a text file. The communication between a UIP scrollbar object and the UIA is implemented via Java sockets. Thus these objects can be distributed for different purposes like controlling of different face muscles from different machines in a network.

Agent construction and communication is supported by JATLite. Because the social interaction dynamics model is included in our system, the penalty with the JATLite not supporting the social modeling of agents is overcome. The agents in the system uses the KQML[17] layer of JATLITE for inter agent communication. KQML messages are based on speech-act theory. Standard KQML primitives are used in communication.

We considered two versions of the Water's work as the implementation of GFAP. The program written by Waters uses OpenGL[18] library. In the 2^{nd} version, texture mapping is supported, and the used library is Glaux. In the first version, texture mapping is not supported and Glut library is included. These libraries are OpenGL's support libraries, and one of them should be used. We used the 2^{nd} version and needed some manipulations on this program for our system. The original program uses parameters changed by the user with keyboard instructions. It uses the Glaux[18] Library which does not support timer function. In our system, the parameter values are changed by the user. The text file consisting of facial parameters is updated according these user interactions. Therefore new facial parameters should be read at certain time intervals to show the updates on the facial model of GFAP. To integrate our UIP with the GFAP, timer function should have been added. So the program is converted to use Glut[18] Library consisting timer function and supporting the texture mapping with using the 1^{st} version of the GFAP.

The clustering process of the CA is implemented by using three clustering algorithms. These algorithms are, Fuzzy c-means [20], Complete-link [20], and Single-link [20]. These algorithms produce different results for the same facial expressions. We have used partition coefficient criteria [21] with the FCM algorithm to indicate the optimum number of clusters.

Some parameter effectiveness arrays as additional information can be determined by the CA for some special facial expressions. This is implemented by our converted Q-learning process as a reinforcement learning method.

4 The Converted Q-Learning Process For Non-Deterministic Environments

Our converted Q-learning method is based on the Q-learning algorithm[22]. This process is implemented by the CA to determine parameter effectiveness arrays for some facial expressions. The primary aim of the learning process is to provide additional information for the clustering process of facial expressions. Additionally, the results can also be used for some social analyses. After Q-learning criteria is met, a parameter effectiveness array is being learned. In our application domain, the environment consists of all the users and their corresponding UIAs. The users act as critics in the learning process. The environment is non-deterministic, because feedbacks taken from the 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 no importance on which agent criticizing because of the non-deterministic property of the environment.

In our application domain, the Q learning algorithm is accommodated to the facial parameter learning process. The states are the corresponding parameter effectiveness arrays. 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, 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 that muscle parameters (left and right) are effective or not on the current facial expression. The “1” bit represents effectiveness on the facial expression for the corresponding muscle. A sample state is like [111011100] which means that 4, 8 and 9th muscles are non-effective on the facial expression. The learned result will be the desired parameter effectiveness array. The CA holds the state graph for traversing states and applying Q-learning updates on them. The hamming distance between two neighbor states is 1. Therefore 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. While traversing the states, the Q values of states are updated. Convergence criteria is met when a state’s Q value increases more than the others. Because the environment is non-deterministic, to learn the convenient state requires more steps than the deterministic case.

In each learning state a facial expression is mapped to the current state and this facial expression parameters are sent to a random UIA. The CA takes feedbacks from a user by means of the related UIA which the mapped facial expression is sent. The reward in our approach is calculated as a function of the feedback and the belief of the agent on the grade.

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 feedbacks 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.

5 Experimental Results

The facial expression generator system 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 was: "Draw me a picture of a person whose dreams come true." A sample set of 10 user drawings are shown in Fig. 4.

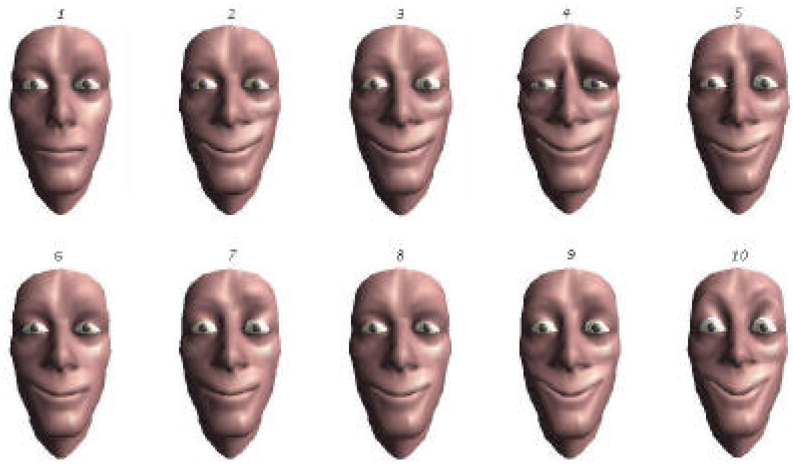

Fig. 4. A sample result set for the expression of a person whose dreams come true

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 were activated. The FCM algorithm generated five optimum clusters. The Complete-link algorithm generated two clusters and the Single-link algorithm generated three clusters. The results of the algorithms are shown in Table 1. Each column in the table points to a user parameter array whose facial drawing can be seen in Fig.4. The numbers in the table indicates the cluster numbers. The FCM, Complete-link and Single-link algorithms have their corresponding clustering distribution in each row respectively. The exceptional situation for the sample question can be seen in the 1st picture of Fig. 4. The data sample for this picture is isolated from other clusters in each of the three algorithm results. The interaction partner choices depend on the selected algorithm.

Table 1. 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 Fig. 5. In this figure, convergence steps for each state can be found. Because different states have different bit representations the convergence steps are different.

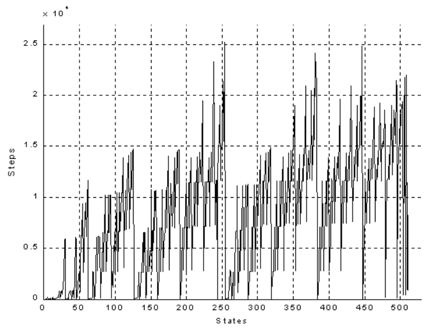

Fig. 5. Learning process simulation results for each state

6 Conclusion

In this work, a personal facial expression generator system with interacting agents is presented. The service for one user is determined in real-time, based on the interaction information by the corresponding agent. Learning specific details of each user profile is a resource consuming process, so we have implemented a system analyzing society information in a multi user domain.

The specific user groups or societies may be formed to consider their data separately. There may be needed to multiply the number of Routers in the system to represent different societies. The 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 learning process is implemented on a distributed user group which 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 weights. The simulation of all the grades generated by a small group causes this facility be non-effective because many trials are needed. If we choose grade values randomly, there will not be meaningful results. However, a pedagogical expertise was not used to get efficient results in the implementation. The non-determinist property of the environment is coped with the Q learning based algorithm in the learning process.

The difficulty on measuring performance of the implemented system for the application is that 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 more friendly. 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 to diagnosis of some disabilities with the help of the other social tests prepared by pedagogues.

According to the Darwin's theories, the facial expressions are universal, and the social culture does not affect them to have major differences. This subject is still being discussed by social experts. Our system results can be useful for some arguments.

Our 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 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 according to the application domain can be added to the system as a future work.

7 References

1. Darwin C.: The Expression of the Emotions in Man and Animals,New York: D. Appleton (1872)
2. Ekman, P.: In Dalgleish, T., & Power, M. Handbook of Cognition and Emotion. New York: John Wiley & Sons Ltd. Chapter 16. Facial Expressions (1999)
3. Terzopoulos D., Waters K.: Analysis and Synthesis of Facial Image Sequences Using Physical and Anatomical Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence,15*(6) (1993), 569-579.
4. Ezzat T., Poggio T.: Mike Talk: A Talking Facial Display Based on Morphing Visemes, *Proc. Computer Animation Conference* Philadelphia, Pennsylvania (1998), 96-102.
5. Tian Y., Cohn J.: Recognizing Action Units for Facial Expression Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(2) (2001), 197-114.
6. SIGGRAPH 97 Panel on Facial Animation: http://www.cs.toronto.edu/~siggraph97-panel/
7. Koda T., Maes P.: Agents with Faces: The Effects of Personification of Agents, *Proc. HCI'96,* London, UK, August 20-23 (1996), 98-103.

8. Luck M., Aylett R.: Applying Artificial Intelligence to Virtual Reality: Intelligent Virtual Environments, *Applied Artificial Intelligence* 14 (2000), 3-32.
9. Liu Y-C., Sato H., Terashima N., Ohya J.: Construction Of Facial Expressions Using A Muscle-Based Feature Model, IEEE International Conference On Multimedia And Expo, August (2002).
10. Rea and Galdalf: http://gn.www.media.mit.edu/groups/gn/
11. Breazeal C., Edsinger A., Fitzpatrick P., Scassellati B.: Active Vision for Sociable Robots, IEEE Transactions on Systems, Man and Cybernetics- Part A : Systems, and humans, Vol .31. No.5 (2001)
12. Cañamero L., Fredslund J.: 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)
13. Parke F. I.: Parametrized Models for Facial Animation. IEEE Computer Graphics and Applications Vol. 2, No.9 (1982) 61-68
14. JATLITE: http://java.stanford.edu
15. GFAP:http://www.crl.research.digital.com/publications/books/waters/Appendix1/appendix1.html
16. Uz B., Gudukbay U., Ozguc B.: Realistic Speech Animation of Synthetic Faces, Computer Animation'98, IEEE Computer Society Publications, Philadelphia (1998)
17. Finin T., Fritzon R.: KQML- A Language and Protocol for Knowledge and Information Exchange. Proceedings of 19th Intl. DAI WorkShop (1997) 127-136
18. OpenGL, Glut and Glaux Libraries: http://www.opengl.org
19. Ross, T.: Fuzzy Logic With Engineering Applications. McGraw-Hill Companies, Inc (1995).
20. Jain A. K., Murty M. N., Flynn P. J.: Data Clustering: A Review, *ACM Computing Surveys*, Vol. 31, No.3 (1999).
21. Bezdek J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
22. Watkins C.J.C.H.: Learning from delayed rewards PhD Thesis, University of Cambridge, England (1989)