A Framework for Socially Communicative Faces for Game and Interactive Learning Applications

Steve DiPaola
Simon Fraser University
Vancouver, BC, Canada
778-782-6579
sdipaola@sfu.ca

Ali Arya
Carleton University
Ottawa, Ontario, Canada
613-875-2792
arya@carleton.ca

Figure 1. Sample animated heads showing expressions, talking, and moving, all created from one synergistic system.

ABSTRACT

In this paper, we describe a modular multi-dimensional parameter space for real-time face game-based animation.. Faces are our most expressive communication tools. Therefore a synthetic facial creation and animation system should have its own tailored authoring environment rather than using general purpose tools from image, 2D and 3D animation. This environment would take advantage of a knowledge space of faces types, expressions, and behavior, encoding known facial knowledge and meaning into a comprehensive, intuitive facial language and set of user tools. Since faces and face expression work on so many cognitive levels, we propose a multi-dimension parameter space called FaceSpace as the basic face model, and a comprehensive authoring environment based on this model. We describe the underlying mechanisms of our environment, and also demonstrate its early game applications and content process**.**

Categories and Subject Descriptors

I.6.7 [**Simulation and Modeling**]: Simulation Support Systems. I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism.

General Terms

Algorithms, Design, Experimentation, Human Factors.

Keywords

Facial Animation, Gaming, Communication Systems.

1. INTRODUCTION

The last decade of twentieth century experienced the merging of some traditionally separate forms of audio-visual art and entertainment. The boundaries between "live action" feature films, animation, and games started to disappear, and the key to the newly forming comprehensive medium was "interactivity". Advances in computer hardware and software have introduced the Interactive Multimedia Presentation as a common base for a variety of audio-visual applications, and computer-generated facial animation is a rapidly growing part of such presentations. For instance, although current computer games make limited use of facial expressions, next generation game platforms provide hardware capabilities for computations involved in having more degrees of freedom in characters. One of the main objectives of game designers is to utilize these new platforms to introduce more realistic characters who can change expressions more frequently, demonstrate personality traits more clearly, and behave more interactively. A virtual customer service representative can be considered another application of such characters.

FuturePlay 2007, November 15-17, 2007, Toronto, Canada.

Some of the issues facing content and application developers in this regard are:

Behaviour. Designing different facial actions, expressions, and personality traits usually involve a painstaking and time-consuming process where artists create the related animation using conventional 3D software and by defining key frames for the movement of each facial feature. This is one of the major difficulties of increasing the number of "moveable" features (and so the realism).

Re-usability. Designs for one head model are not generally usable on another model. As a result, even a similar action on a new head requires the design process to be repeated.

Interaction. The need for a detailed design process limits the amount of interactivity and dynamic behaviour a character can have at run-time. In other terms, the characters can not be completely autonomous.

Programmability. There is a serious lack of programmable components that can be re-used in new applications to provide facial animation capabilities. Each application has to be developed by implementing such functionality from scratch.

Level of Details. The animators, especially when using conventional graphics software, have to deal with all the details of a head model to perform actions. An intelligent software that is aware of head regions and their function can hide the details unless necessary, by performing group actions on all the points that are functionally related. For example, averting the gaze direction is a simple action that should involve only a single input as new direction. The rest, i.e. rotating eyeball points, should be taken care of by the software. Such a feature is missing in most design and runtime environments due to the fact that they are not customized for face animation.

In this paper, we introduce Interactive Face Animation – Comprehensive Environment (iFACE) that provides solutions to all of the above problems in a unified face animation framework. iFACE parameter spaces allow animator and/or programmer to effectively control facial geometry, perform MPEG-4 compatible facial actions, show expressions, and display behaviours based on definable personality types. All of these are encapsulated within a Face Multimedia Object (FMO) that can be used in any applications through programming interfaces. The framework includes stand-alone graphical design environments and plug-in components for 3D software programs such as Maya.

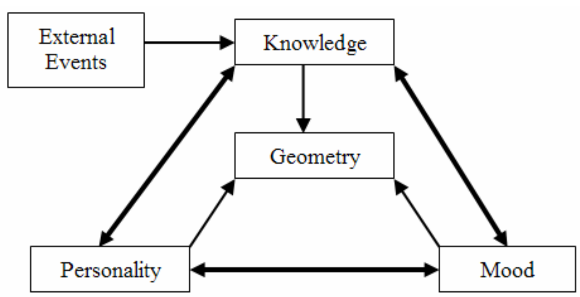

Figure 2. iFACE Parameter Spaces

iFACE hierarchical geometry arranges parameters in different layers of abstraction to allow exposure to proper level of details. The physical points can be pixel or vertex to support 2D and 3D models, both with the same control interface. On top of them are the feature points which correspond to MPEG-4 Face Animation and Definition Parameters. Using only these parameters almost any facial action is possible. Features and component layers are at higher levels and allow grouping of functionally related parameters. The parameters and their respective API are independent of the head model, so a set of parameters can be applied to any head model, resulting in the same facial actions.

Dynamic behaviours in iFACE are possible through Knowledge, Mood, and Personality parameter spaces. They allow defining interaction rule and scripts written in Face Modeling Language (FML), expressions, and personality types. iFACE personality types are based on the state-of-the-art in behavioural psychology. They are defined as a combination of Affiliation and Dominance factors and control the way facial actions are performed (e.g. frequency of blinking, typical head movements, etc). Using these parameters, an autonomous character can be created which interacts properly in a dynamic environment.

The parameterized approach with its behavioural extensions allow the animators and runtime programmers to use "face" as a self-supporting object without the need for dealing with details, and apply the same parametric design to any head model. iFACE API on the other hand, provides a powerful flexible component-based structure to be used in any application that requires face animation. The API itself uses a layered architecture to access different functionality.

Some related works in face animation are briefly reviewed in Section 2. The main concepts of FMO and iFACE framework are discussed in Section 3. iFACE architecture and object model are presented in Section 4, and some experimental results related to SAGE project are presented in Section 5. Some concluding remarks and discussions are the subject of Section 6.

2. RELATED WORK

The common practice for face animation is to use general-purpose 3D modeling and animation tools such as Alias Maya, Discreet 3DS Max or SoftImage XSI. While providing very powerful design environments, these tools lack dedicated face-centric features that allow efficient and realistic modeling and animation of facial states and actions [12]. The run-time environments, consequently, animate the limited degrees of freedom provided by the authoring tools, and do not support any face-specific run-time support (such as dynamic creation of typical behaviours) that can simplify application development. Although this situation might be marginally sufficient for current level of face animation in games, and match the computational power of existing game consoles, it is far less than ideal, especially for next generation games running on much more powerful platforms.

One of the earliest works on computerized head models for graphics and animation was done by Parke [19]. It can be considered as the first parameterized head model which was extended by other researchers [20, 9] to include more facial features and add more flexibility. Different methods for initializing such "generic" model based on individual (3D or 2D) data have been proposed and successfully implemented 20]. Parameters are usually grouped into conformation and expression categories, the former for building a particular head and the latter for animating it. The parameterized models are effective ways due to use of limited parameters, associated to main facial feature points. Such parameters can be used to calibrate a standard model and animate it. Facial Action Coding System (FACS) [12] was an early and still valid study of possible facial actions related to such feature points. Although not originally a computer graphics technique, FACS has been widely used by researchers in parameterized models and others. This approach has been formalized in MPEG-4 standard by introduction of Face Definition Parameters (FAPs) and Face Animation Parameters (FAPs) [3]. The former group of parameters defines the important features and the latter actions applied to a subset of them (not every FDP can be animated).

The primary issue with such parameter space is its "flatness". All parameters are worked with in a similar way while not every application actually needs all of them. A hierarchical grouping of parameters is necessary for efficient management of parameter space. Also, the relatively huge amount of parameters (result of extensions to the original models) makes application development and authoring hard. A common solution to this issue has been defining higher-level parameters and behaviors [5, 6, 8]. For example, “smile” is an action that involves a few features and can be defined at a higher level of abstraction, knowing the combined effect of movements in feature points. MPEG-4 FAPs define two groups of such high level parameters for standard facial expressions and visemes (visual representation of uttered phonemes). Although such “macro” parameters make it easier to use the underlying model, the simple two-tier model is still not very effective for managing facial activities and providing local control over level-of-details. The MPEG-4 standard allows definition of parameter groups but it is only a standard to be used by particular models, which are still mainly “flat”. Pasquariello, and Pelachaud [21] (among others) have proposed hierarchical head models that allow a more efficient parameter control through grouping and regions. Our approach, as explained later, uses this idea and extends it to multiple layers of abstraction on top of actual data points (2D pixels or 3D vertices) to ensure maximum flexibility and minimum effort when group actions are required. Our head model pyramid has a head object on top, and components, features, feature points, and physical points are at lower levels.

Physically-based head models form another approach in modeling head and face [29, 18, 16]. Here the physical and anatomical characteristics of bones, tissues, and skin are simulated to provide a realistic appearance (e.g. spring-like elasticity). Such methods can be very powerful for creating realism but the complexity of facial structures make them (1) computationally expensive, and (2) almost always not enough adequate. Considering the effectiveness of parameterized models for communicative purposes (as explained in the next section), it maybe argued that physically-based models are not a very efficient choice in many applications. This does not deny the advantages of physically-based models and the fact that they can even be used within the context of parameterized models to provide local details when needed.

Image-based methods have also been used for head and face modeling. In animation, image-based methods are mainly based on morphing between given images and applying pre-learned transformations in order to create new ones [2, 13, 26]. No need for complicated 3D computation and data, and also image-based photo-realism, are major advantages of such methods, but their capability in creating a wide range of facial actions is limited due to unavailability of 3D information. A general head model that can utilize both 2D and 3D data with the same user and programming interface will be very promising, in this regard.

The concept of “facespace” as a universal space of all faces that can be created by given parameters (features or image templates) has also been studied by researchers such as Valentine [28] and DiPaola [10]. Focusing more on face recognition, Valentine considers this space to have dimensions formed by distinctive geometric features or Principal Component Analysis (PCA) vectors (i.e. Eigenfaces) [27]. In either case, the space is not time-based. DiPaola introduces the behaviors, and effectively extends the facespace to include temporal changes. But the nature of these time-based dimensions and their relation to the geometry has not been clearly defined.

Finally, the behavioral modeling of animated characters has been studied by some researchers. Funge et al. [14], for instance, define a hierarchy of parameters. At the base of their parameter pyramid is the geometric group. On top of that come kinematic, physical, behavioral, and cognitive parameters and models. Although very important for introduction of behavioral and cognitive modeling concepts, the model may not be very suitable for face animation purposes due to the interaction of parameter groups and the need for emotional parameters as opposed to physically-based ones.

Cassell et al. [6,7] defined behavioral rules to be used in creating character actions but do not propose a general head model integrating geometrical and behavioral aspects. Byun and Badler [5] propose the FacEMOTE system that allows four high-level “behavioural” parameters (Flow, Time, Weight, and Space) to control the expressiveness of an input FAP stream. Although it demonstrates how high-level behavioural parameters can control facial animation, it does not intend to be a comprehensive face object. On the other hand, three spaces of Knowledge, Mood, and Personality (each with their own parameters as explained later) can control the facial behaviour in a more explicit way. RUTH system by DeCarlo et al. [8] uses behavioural rules to animate a face when a given text is spoken. Smid et al. [25] use a similar approach but associate a considerably larger set of facial actions (head, eye, brow movements) to features of a given speech through behavioural rules in order to create an autonomous speaker agent. Although these rules can be base for defining personality types, the possibility has not been explored by these researchers. Pelachaud and Bilvi [22] propose performative dimensions (dominance and orientation) and emotion dimensions (valence and time) as behavioural parameters to control facial actions. Published at the same time as original iFACE [10], the systems share common concepts but iFACE provides a more comprehensive framework for defining personality types and custom expressions, and it is based on studies in behavioural psychology to associate facial actions to these personality types and expressions. Also, iFACE allows interactive non-verbal scenarios through an XML-based scripting language, MPEG-4 compatibility at lower levels, multimedia streaming, authoring tools, programming interfaces, and wrapper applets for form-based applications. Personality space also allows a more general mechanism for defining facial personalities.

3. IFACE SYSTEM

3.1 Face Multimedia Object

The ability to create a multimedia presentation as a combination of parts from different sources (e.g. separate foreground object and background scene) has resulted in new multimedia standards such as MPEG-4 [3] that treat the presentation not as one piece of data but a collection of objects of different types. The simplest of these types can be audio and video. The need for more efficient multimedia authoring and management suggest that such object-based approach be extended to more “aggregation” in multimedia content, i.e. grouping of related content elements into higher-level “types”. For a variety of cases where human figures play a key role (“face-centric” applications) “face” is a primary candidate for such a data type. The introduction of Face Definition and Animation Parameters (FDPs and FAPs) in MPEG-4 standard was a step toward such higher-level of abstraction on top of face-related multimedia content. The authors have proposed Face Multimedia Object (FMO) [1] as a more systematic approach to encapsulate face functionality into one autonomous but controllable object.

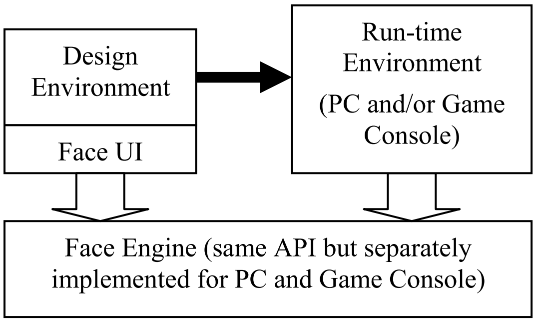

Figure 3. Using Face Multimedia Object

To be successful, FMO needs to provide means of creating and displaying the desired appearance and behaviours, and also expose proper control mechanisms through user interaction or program access. The face appearance depends primarily on its geometry. In general this geometry can be 2D or 3D, realistic or stylistic. It includes high-level functionality such as resizing a region and low-level ones such as manipulating a point (2D pixel or 3D vertex). High-level functionality should be independent of the type of geometry we are using (i.e. the same programming/control interface for all the faces in Figure 1). This suggests a hierarchical model that exposes as much detail as necessary through different layers of abstraction. Facial behaviour on the other hand, depends on individual-independent rules of interaction and scenarios and also individual characteristics such as short-term moods and long-term personality traits. All of these have to be developed in a way to be suitable for interactive and real-time performance.

As shown in Figure 3, FMO operates as a "face engine" for design-time and run-time applications. Using FMO, animators and authors can design proper geometry and facial actions and pass them to run-time modules only as commands, instead of keyframes with information on all moving parameters. At run-time, the application/game only detects the required action and asks the engine to replicate the same result. This has the advantages such as:

1- Less information saved by design tool and passed to run-time

2- Ease of run-time development due to black-box use of FMO

3- Possibility of dynamic applications and user-controlled event-driven scenarios without the need of a pre-design

3.2 Parameter Spaces

For a large group of applications, facial presentations can be considered a means of communication. A "communicative face" relies and focuses on those aspects of facial actions and features that help to effectively communicate a message. This may not require very detailed data, but efficient use of parameters and their structural patterns. This means that the head/face model has to provide local control of level-of-details and high-level to low-level functionality.

Rousseau and Hayes-Roth [23] consider Personality Traits, Moods, and Attitudes as major parameters in their social-psychological avatar model. In a similar but revised way, we believe that the communicative behavior of a face can be considered to be determined by the following parameter spaces

Geometry: This forms the underlying physical appearance of the face. Creating and animating different faces and face-types are done by manipulating the geometry that can be defined using 2D and/or 3D data (i.e. pixels and vertices). This geometry is based on a hierarchy of facial regions and sub-regions.

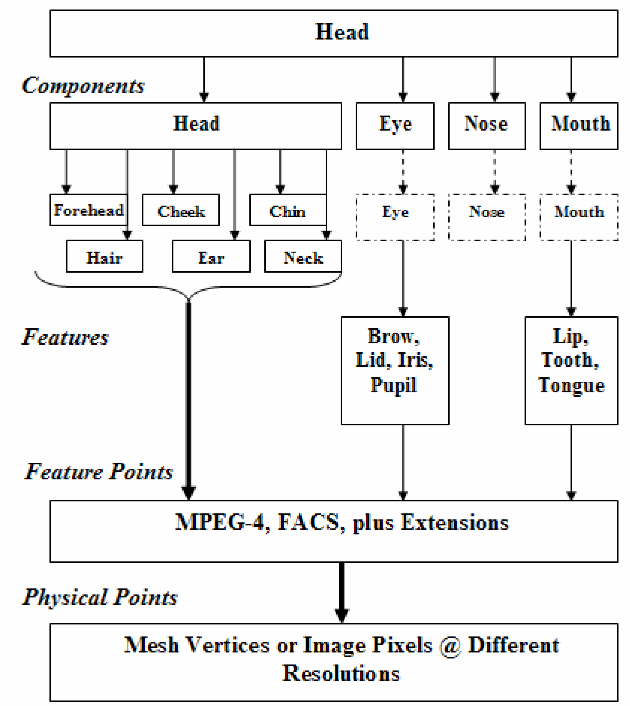

Figure 4. iFACE Geometry Hierarchical Head Model

- *Knowledge*: Behavioral rules, stimulus-response association, and required actions are encapsulated into Knowledge. In the simplest case, this can be the sequence of actions that a face animation character has to follow. In more complicated cases, knowledge can be all the behavioral rules that an interactive character learns and uses (see Funge et al. work on Cognitive Modeling [14]). Knowledge acts through an XML-based language that defines scenarios, events, and decision-making logic.
- *Personality*: Different characters can learn and have the same knowledge, but their actions, and the way they are performed, can still be different depending on individual interests, priorities, and characteristics. Personality encapsulates all the long-term modes of behavior and characteristics of an individual [30], 4]. Facial personality is parameterized based on typical head movements, blinking, raising eye-brows and similar facial actions.
- *Mood*: Certain individual characteristics are transient results of external events and physical situation and needs. These emotions (e.g. happiness and sadness) and sensations (e.g. fatigue) may not last for a long time, but will have considerable effect on the behavior. Mood of a person can even overcome his/her personality for a short period of time. Emotional state can be modeled as point in a 2D space where two axes correspond to energy and value [24].

3.3 Face Geometry

Geometry Components and Regions allow grouping of head data into parts that perform specific actions together (e.g. resizing the ears or closing the eye). Features are special lines/areas that lead facial actions, and Feature Points (corresponding to MPEG-4 parameters) are control points located on Features. Only the lowest level (Physical Point) depends on the actual (2D or 3D) data. iFACE Geometry object model corresponds to this hierarchy and exposes proper interfaces and parameters for client programs to access only the required details for each action. Facial Regions are shown in Figure 5. iFACE authoring tool (iFaceStudio) allow users to select Feature Points and Regions. Each level of Geometry accesses the lower levels internally, hiding the details from users and programmers. Eventually, all the facial actions are performed by applying MPEG-4 FAPs to the face.

Although geometric parameters can change over time, they have a pure spatial nature by themselves, and can build a multi-dimensional spatial facespace in Valentine's terms, i.e. all geometrically possible faces. Knowledge, Personality, and Mood, on the other hand, have a time-based nature since they define the way different geometric states are used and related. Together, these four groups of parameters serve as "meta-dimensions" of a spatio-temporal facespace.

Time does not act as an explicit dimension in this facespace but it is implied in it. On the other hand, geometry plays a particularly different role, as it is possible to create any face by changing only geometric parameters. The other meta-dimensions only make these changes meaningful and purposeful. This is illustrated in Figure 2, where Geometry acts as the foundation while others interact with it and with each other.

Figure 5. Face Regions. These are small areas that usually move together and are controlled by Feature Points. iFACE Components are related groups of these Regions, e.g. eye area

3.4 Face Modeling Language

The behaviour of an iFACE character is determined primarily by Knowledge. It provides the scenario that the character has go through as an XML-based script. iFACE uses Face Modeling Language (FML) [2] that is specifically designed for face animation. FML document can be a simple set of sequential actions such as speaking and moving the head, or a complicated scenario involving parallel actions and event-based decision-making similar to the following script:

```
<fml>
  <model>
    <event name="kbd" />
  </model>
  <story>
    <action>
      <!--parallel actions-->
      <par>
        <hdmv type="yaw" value="80"
            begin="0" end="2000" />
        <play file="Audio1.wav" />
      </par>
      <!--exclusive actions -->
      <!--only one of options will run-->
      <excl ev_name="kbd">
        <talk ev_value="F1_down">Hello</talk>
        <talk ev_value="F2_down">Bye</talk>
      </excl>
    </action>
  </story>
</fml>
```

iFACE Knowledge module exposes interfaces to allow opening new scripts or running single FML commands. It also allows defining and raising program-controlled events that are base for dynamic and interactive scenarios.

3.5 Mood

Although scripts can select a new personality or modify the mood, but Knowledge space is generally independent of the "character". Mood and Personality spaces deal with character-dependent parameters. Mood controls short-term emotional state that can affect the way a certain action is animated. For instance actions in a part of script can be performed in a "happy" mood and in another part in a "sad" one and be visually different. In general, moods are represented with a certain facial expression with which any facial action is as in Figure 6. iFACE supports two types of moods each with a zero to one activation level: 1) Standard emotions (joy, sadness, surprise, anger, fear, disgust) predefined based on previous studies and 2) Custom expressions defined by user. It is also possible to select the current mood of character by adjusting Energy and Stress values which will result in activation of standard emotions at some level according to Russell's Circumplex mood model [24] where horizontal and vertical dimensions are Stress and Energy, respectively.

3.6 Face Personality

Interpersonal Adjective Scale [30] is a widely accepted personality model that links different personality types to two Affiliation and Dominance parameters in a two dimensional Circumplex model (Figure 8). Facial actions and expressions are shown to cause perception of certain personality traits [4, 17]. The foundation of iFACE personality is associating major facial actions and expressions with personality parameters and types, i.e. visual cues for personality. This is done based on published works and our own on-going research (described in Section 4 as an iFACE application). When the personality parameters are changed or a certain personality type is activated, the associated facial actions are selected to be performed (i.e. visual cues are presented) in order to make perception of that personality type more probable in the viewer. Following personality-related actions are defined: Expressions, 3D head movements, Nodding, Raising/lowering/squeezing eyebrows, Gaze shift, Blinking.

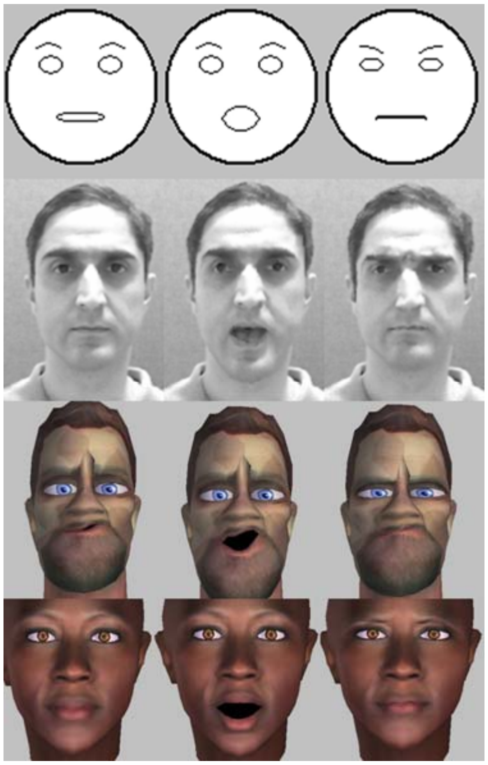

Figure 6: Neutral, Talking, and Frowning facial states (left to right) of four different characters.

For each one of these, strength, duration, and frequency of occurrence are controllable. The visual cues can happen randomly, with a pre-defined order, or based on the voice energy when talking. Two threshold values are set for speech energy: Impulse and Emphasis. When the energy reaches any on of these thresholds, certain visual cues of current personality type are activated, for instance nodding when emphasizing on a part of speech. We use ETCodec lip-sync module by OnLive that calculates a speech energy for each audio frame of 60mSec. ETCodec also gives values for mouth shape which are translated to MPEG-4 FAPs by iFACE.

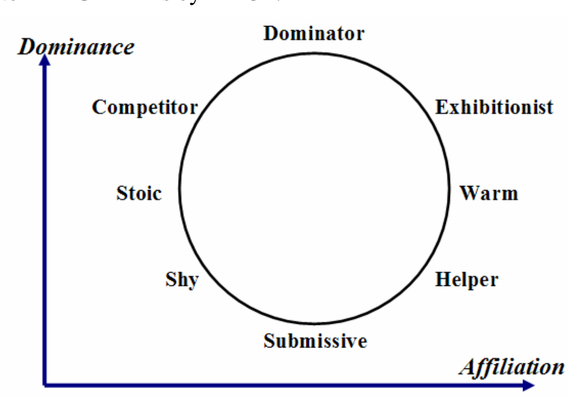

Figure 8. Personality Circumplex Model

Sample associations between visual cues and perceived personality types are shown in Table 1. These are the result of an on-going study which is the subject of another paper.

Table 1. Visual Cues and Personality Types

Visual Cue	**Perceived Personality Type**
Happiness and surprise	high in dominance and affiliation
Anger	High dominance / low affiliation
Sadness and fear	low in dominance
Averted gaze	avoidance (low affiliation)
Moving away	low affiliation
Frequent moving	high dominance
eyebrow raise - 1 sided	high dominance
Tilted head	low dominance and/or high affiliation
Wide-open eyes	high affiliation
Frequent blinking	low affiliation / low dominance

3.7 Software Architecture

iFACE is developed as a set of .NET components written with C#. It is possible to access them directly from .NET Managed Code or through .NET/COM Interop from Unmanaged Code[1]. It uses Microsoft Direct3D and DirectSound for graphics and audio purposes. In cases where Unmanaged Code was required (for instance using existing ETCodec lip-sync library) a COM object is developed to wrap the code and use it in iFACE. Implementation of iFACE FMO on game consoles as a possible run-time environment, and iFACE plug-in components for Maya and 3DS-MAX are on-going projects. iFACE is designed to work with a variety of client types. Depending on the level of details exposed by the components, iFACE objects are grouped into three layers shown in Figure 9.

4. APPLICATIONS WITH IFACE

4.1 Face Personality Study

Behavioural psychology researchers usually use photographs and less commonly video to perform experiments. They can benefit from an interactive environment that can create realistic animations with different features (e.g. mood and personality). This can replace actors which are hard or expensive to find with software that does not need external setup and can be easily configured. iFACE system is being used in such an application which in turn provides information regarding how viewers perceive the personality of a subject based on his/her facial actions.

Using iFACE the researcher can change the personality traits (or any other aspect) of the subject and observe the reaction and

[1] Code written for .NET framework (e.g. a C# program) is called Managed Code. Normal Windows and Component Object Model (COM) code are considered Unmanaged. .NET allows interoperability with COM objects using a mechanism called COM Interop.

perception of the viewers. For more information see iFACE web site: http://ivizlab.sfu.ca/arya/Research/FacePersonality

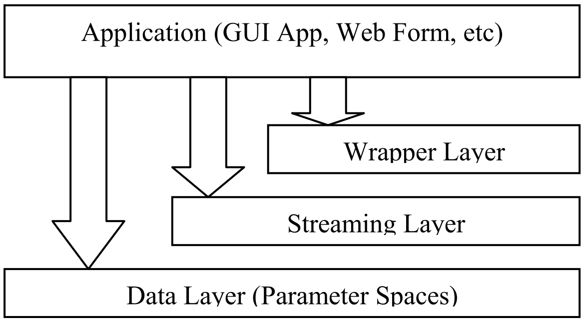

Figure 9. iFACE Layered Architecture

4.2 COMPS

Collaborative Online Multimedia Problem-based Simulation (COMPS) [11] is a system being developed to provide Problem-Based Learning (PBL) tools for medical students. PBL works by introducing students to a case (problem), giving them some facts, and taking the students through cycles of discussion and hypothesizing until the disease is correctly identified.

A major part of a PBL-based approach for medical students is to interact with patients, especially listening to them describing their symptoms. Bringing patients to a classroom or examination room is hard and in some cases impossible. Using "actors" for this purpose is a common but rather expensive alternative. A Social Conversational Agent (SCA) is an ideal replacement. SCA can also be a proper for "automated instructor" or to represent a remote instructor (or patient) when transmitting real-time video is not possible but SCA can be animated based on real audio data. Here, we briefly review two examples of "simulated patient" and "remote instructor" as typical applications of iFACE in COMPS:

- *Simulated Patient*: An FML script file is primary animation control file for iFACE. Using iFaceStudio authoring tool, a set of keyframe animations are created to represent typical head movements of the patient. These are then associated with a new personality type. The script selects the type and then gives the face object a text or audio file to "speak". During the speech, the typical behaviours (head movements) are selected randomly and performed by the animated head, as explained in Section 3. The presentation can be more complicated using event processing and decision-making capabilities of FML. Events can be associated with user selections (e.g. from pre-defined set of questions) and the animation can go through different branches (see script sample of Section 3 and the usage of keyword **`excl`**).

- *Remote Instructor*: A simpler mechanism for controlling iFACE animation is to provide only the audio data as input. Data can come from a local file or a network stream. A remote instructor can use iFACE recording capability to send his/her voice data to another (or group of) remote iFACE objects which in turn use the data to drive the animation. Again proper personality and mood can be selected.

Figure 10. COMPS Screenshot with a Simulated Patient

4.3 Storytelling Masks

iFACE is used in a museum environment to create animations of native North American artists explaining their work and the myths related to them using a game metaphor, as illustrated in Figure 11 where a real artist's voice, passion, stories and expression first introduces himself and his work (A), begins to transform into his artwork (B), has his work tells it's back story with full voice and expression (C,D) and can return to his persona to interactively answer questions or give other educational content (A)

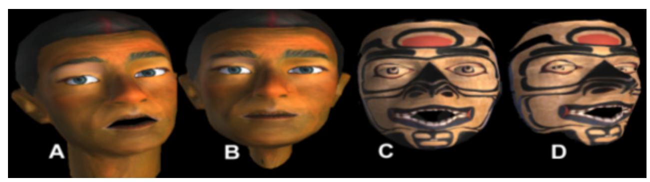

Figure 11. Frames from "Storytelling Masks".

4.4 Evolving Faces

Human migration, as explained in "out of Africa" theory, is illustrated in this application using talking faces of each region/age, as shown in Figure 12 where emotive talking faces describe the DNA science of human migration out of Africa, actively "morphing" facial types accordingly.

Figure 12. Screenshot of 'Evolving Faces'.

5. CONCLUSION

In this paper, we describe iFACE as a framework for face multimedia object. iFACE encapsulates all the functionality required for face animation into a single object with proper application programming interface, scripting language, and authoring tools. iFACE use a hierarchical head model that hides the modeling details and allows group functions to be performed more efficiently. Multiple layers of abstraction on top of actual head data make the client objects and users independent of data type (3D or 2D) and provide the similar behaviour regardless of that type.Behavioural extensions in form of Knowledge, Personality, and Mood control scenario-based and individual-based temporal appearance of the animated character. On the other hand, streaming and wrapper objects make the use of iFACE components easier in a variety of applications. iFACE framework is a powerful "face engine" for character-based online services, games, and any other "face-centric" system.

Future research on iFACE will involve comprehensive association of all facial actions and expressions to most likely personality type to be perceived, exploring the possibility of higher level parameters in face personality (on top of affiliation and dominance) in order to define practical character types such as nervous and heroic), and realistic combination of current mood and facial actions by using non-linear functions

6. REFERENCES

[1] Arya A, DiPaola S, 2004 Face as a Multimedia Object, 5th International Workshop on Image Analysis for Multimedia Interactive Services, Portugal.

[2] Arya, A., and Hamidzadeh, B. 2003. ShowFace: A Framework for Personalized Face Animation. In Proceedings of RichMedia-2003, Lausanne, Switzerland, IEEE Press.

[3] Battista, S., et al. 1999. MPEG-4: A Multimedia Standard for the Third Millennium, Multimedia, vol. 6, no. 4, IEEE Press.

[4] Borkenau, P., and Liebler, A., 1992. Trait Inferences: Sources of Validity at Zero Acquaitance. In Journal of Personality and Social Psychology. Vol 62. no 4, pp 645-657.

[5] Byun, M., and Badler, N.I., 2002. FacEMOTE: Qualitative Parametric Modifiers for Facial Animations. In Proceedings of ACM SIGGRAPH/ Eurographics symposium on Computer Animation, ACM Press.

[6] Cassell, J., et al. 1994. Animated Conversation: Rule-based Generation of Facial Expression, Gesture and Spoken Intonation for Multiple Conversational Agents.. In Proceedings of ACM SIGGRAPH, ACM Press.

[7] Cassell, J., et al. 2001. BEAT: The Behavior Expression Animation Toolkit. In Proceedings of ACM SIGGRAPH, ACM Press.

[8] Decarlo, D., et al. 2002. Making Discourse Visible: Coding and Animating Conversational Facial Displays. In Proceedings of Computer Animation Conference, IEEE Press.

[9] DiPaola, S. 1991. Extending the Range of Facial Types. Visualization and Computer Animation, 2, 4, 129-131.

[10] DiPaola, S. 2002. FaceSpace: A Facial Spatial-Domain Toolkit. In Proceedings of InfoViz 2002, 49–55.

[11] DiPaola S, Arya A, Chan J. 2005. Simulating Face to Face Collaboration for Interactive Learning Systems, Proceeding from E-Learn.

[12] Ekman, P. , and FrieSEN, W. V. 1978. Facial Action Coding System, Consulting Psychologists Press Inc.

[13] Ezzat, T. and Poggio, T. 1999. MikeTalk: A Talking Facial Display Based on Morphing Visemes. In Proceedings of Computer Animation Conference, IEEE Press.

[14] Funge, J. et al. 1999. Cognitive Modeling. In Proceedings of ACM SIGGRAPH, ACM Press.

[15] Gross, J., et al. 2000. The Dissociation of Emotion Expression from Emotion Experience: A Personality Perspective. In Personality and Social Psychology Bulletin, vol 26. no 6. pp 712-726.

[16] Kahler, K. et al. 2002. Head Shop: Generating Animated Head Models with Anatomical Structure. In Proceedings of ACM SIGGRAPH/ Eurographics symposium on Computer animation, ACM Press.

[17] Knutson, B. 1996. Facial Expressions of Emotion Influence Interpersonal Trait Inferences, Journal of Nonverbal Behavior, 20, 165-182.

[18] Lee, Y., Terzopoulos, D., WATERS, K., 1995. Realistic Modeling for Facial Animation. In Proceedings of ACM SIGGRAPH, ACM Press.

[19] Parke, F. I. 1972. Computer Generated Animation of Faces. In Proceedings of ACM Annual Conference, ACM Press, 451–457.

[20] Parke, F. I., and Waters, K. 2000. Computer Facial Animation. A. K. Peters.

[21] Pasquariello, D., and Pelachaud, A., 2001. Greta: A Simple Facial Animation Engine. In 6th Online World Conference on Soft Computing in Industrial Applications.

[22] Pelachaud, W. and Bilvi, M. 2003. Computational Model of Believable Conversational Agents. In Communication in MAS: Background, Current Trends and Future.

[23] Rousseau, D., and HAYES-ROTH, B. 1997. Interacting with Personality-Rich Characters. Knowledge Systems Laboratory Report No. KSL 97-06, Stanford University.

[24] Russell, J.A., 1980. A Circumplex Model of Affect. In Journal of Personality and Social Psychology, 39, 1161-1178.

[25] Smid, K. et al. 2004. Autonomous Speaker Agent. In Proceedings of Computer Animation and Social Agents, Geneva, Switzerland.

[26] Tiddeman, B. et al. 2001. Prototyping and Transforming Facial Textures for Perception Research. Computer Graphics and Applications, IEEE Press.

[27] Turk, M. J., and Pentland, A. P. 1991. Face Recognition Using Eigenfaces. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR-91), IEEE Press, 586–591.

[28] Valentine, T. 1999. Face-Space Models of Face Recognition. In Computational, geometric, and process perspectives on facial cognition: Contexts and challenges. Wenger, M. J. & Townsend, J. T. (Eds.), Lawrence Erlbaum Associates Inc.

[29] Waters, K. 1987. A Muscle Model for Animating 3D Facial Expression. ACM Computer Graphics, ACM Press, 21, 4.

[30] Wiggins, J.S. et al. 1988. Psychometric and Geometric Characteristics of the Revised Interpersonal Adjective Scale, Multivariate Behavioural Research, 23, 517-530.