# Socially Communicative Characters for Interactive Applications

Ali Arya
iMediaTek, Inc.,
Vancouver, BC, Canada
aarya@sfu.ca

Steve DiPaola
Simon Fraser University,
Surrey, BC, Canada
sdipaola@sfu.ca

Lisa Jefferies, James T. Enns
University of British Columbia,
Vancouver, BC, Canada
{ljefferies,jenns}@psych.ubc.ca

## ABSTRACT
Modern multimedia presentations are aggregations of objects with different types such as video and audio. Due to the importance of facial actions and expressions in verbal and non-verbal communication, the authors have proposed "face multimedia object" as a new higher-level media type that encapsulates all the requirements of facial animation for a face-based multimedia presentations within one single object. In this paper, Interactive Face Animation - Comprehensive Environment (iFACE) is described as a general-purpose software framework that implements the "face multimedia object" and provides the related functionality and tools for a variety of interactive applications such as games and online services. iFACE exposes programming interfaces and provides authoring and scripting tools to design a face object, define its behaviours, and animate it through static or interactive situations. The framework is based on four parameterized spaces of Geometry, Mood, Personality, and Knowledge that together form the appearance and behaviour of the face. iFACE can function as a common "face engine" for design and run-time environments to simplify the work of content and software developers.

## Keywords
face animation, parameter, behavior, personality, geometry, game, interactive

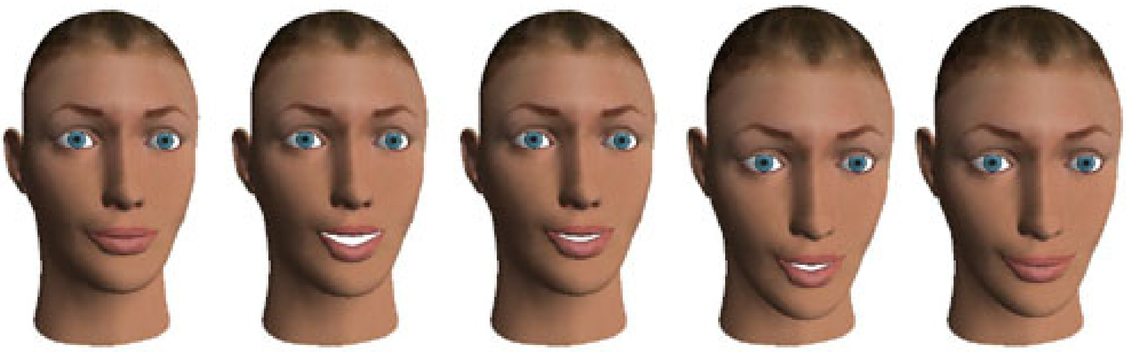

**Figure 1. Sample animated 3D heads showing expressions, talking, and moving head**

## 1. INTRODUCTION
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.



*Conference proceedings ISBN 80-86943-03-8*
*WSCG'2006, January 30-February 3, 2006*
*Plzen, Czech Republic.*


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.

Some of the issues facing content and application developers of realistic interactive characters 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.

*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.

*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.

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.

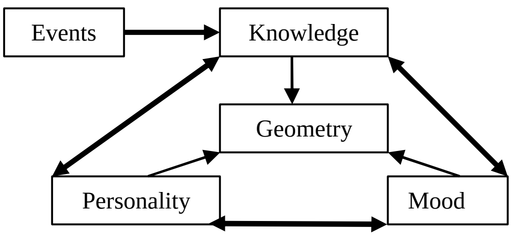


**Figure 2. iFACE Parameter Spaces**

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 [Bat99]. 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.

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 software architecture is also briefly introduced in this section. In section 4, a few applications are presented that use iFACE. Some concluding remarks and discussions are the subject of Section 5.

## 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 [Ekm78]. 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 parameterized head models was done by Parke [Par72]. The parameterized models are effective ways due to use of limited parameters, associated to main facial feature points, compared to complicated simulation of physical entities [Par00]. Facial Action Coding System (FACS) [Ekm78] 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 and Animation Parameters (FDP,FAP) [Bat99, Ost98].

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 [Byu02, Cas94, Dec02]. 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". Fidaleo and Neumann [Fid02] propose using regions to group points with similar movements. Pasquariello, and Pelachaud [Pas01] (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.

Finally, the behavioral modeling of animated characters has been studied by some researchers. Funge et al. [Fun99], 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. [Cas94, Cas01] 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 [Byu02] 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. [Dec02] uses behavioural rules to animate a face when a given text is spoken. Smid et al. [Smi04] 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 [Pel03] propose performative dimensions (dominance and orientation) and emotion dimensions (valence and time) as behavioural parameters to control facial actions. 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

## Face Multimedia Object

The ability to create a multimedia presentation as a combination of parts with different types has resulted in new multimedia standards such as MPEG-4. 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) [Ary04] as a more systematic approach to encapsulate face functionality into one autonomous but controllable object.

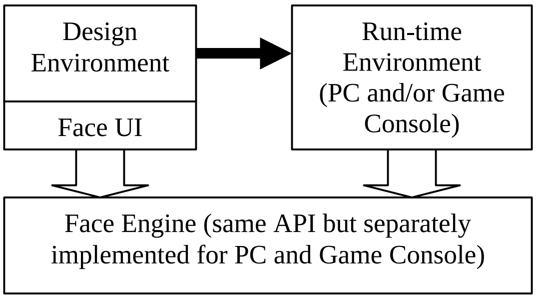


**Figure 3. Using Face Multimedia Object**

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:

- Less information saved by design tool and passed to run-time
- Ease of run-time development due to black-box use of FMO
- Possibility of dynamic applications and user-controlled event-driven scenarios without the need of a pre-design

## Parameter Spaces

Rousseau and Hayes-Roth [Rou97] 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 that can be defined using 2D and/or 3D data (i.e. pixels and vertices). This geometry is based on a hierarchy of facial components, features and points as illustrated in Figure 4.

- *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.

- *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 [Wig88, Bor92]. 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 [Rus80].

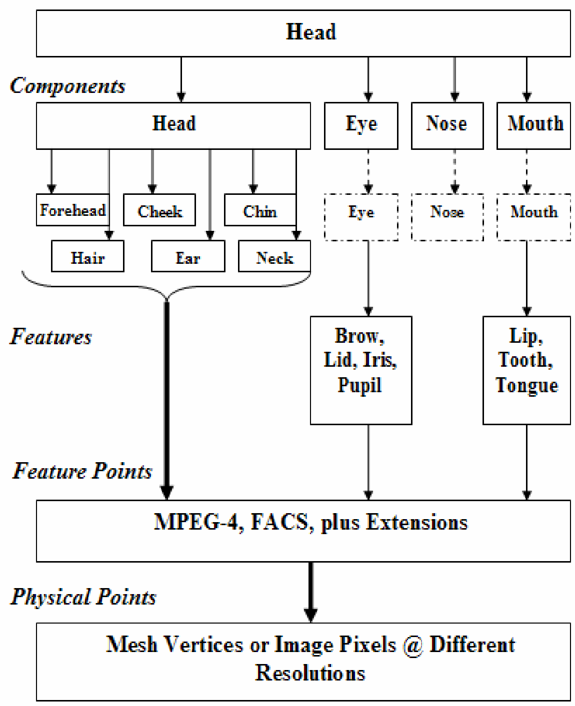


**Figure 4. iFACE Geometry Hierarchical Model**

## Face Geometry

As explained in Section 2, hierarchical geometry in iFACE provides a more efficient way of accessing parameters. Geometry Components 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. iFACE authoring tool (iFaceStudio) allow users to select Feature Points and Regions, i.e. groups of points that usually move wit a feature point (similar to co-articulation regions by Fidaleo and Neumann [Fid02]). Each level of Geometry accesses the lower levels internally, hiding the details from users and programmers. Regions can be defined by selecting 3D point using a 3D editor, or defining an area on 2D texture. iFACE head model includes 60 Regions.

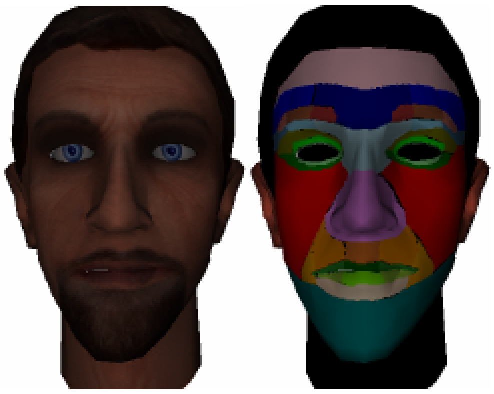

**Figure 5. Face Regions for a 3D head. These are small areas that usually move together and are controlled by Feature Points. iFACE Components are related groups of these Regions, e.g. eye area**

Eventually, all the facial actions are performed by applying MPEG-4 FAPs to the face. FAPs themselves move feature points and regions. Each FAP has a maximum displacement vector (normalized in FAP units) and an activation value. Together these two define the movement of the corresponding feature point due that FAP. Other Region points follow this feature point by distance-dependent weight function as illustrated in Figure 6. The slope and half-weight distance on this function are configurable. The movement of each point is the sum of all movements related to active FAPs (Better methods are being studies).

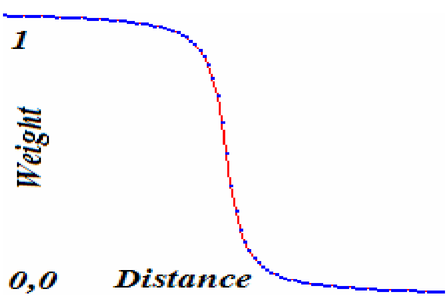


**Figure 6. Weight of movement as a function of distance from feature point**

## 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) [Ary04] 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 -->
      <!—-depending on event value-->
      <!--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.

## 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 modulated (i.e. overlaid on top of actions, Figure 7), as follows:

$ACT = sum(FAP_1, FAP_2, \ldots, FAP_n)$
$EXP = sum(FAP_1, FAP_2, \ldots, FAP_m)$
$DISPLAY = f(BASE,EXP)$

In these equations, *ACT* is the facial action to be performed consisting of *n* FAPs, *EXP* is the current mood made of *m* FAPs, and *DISPLAY* is the final face output. *f* is currently a simple addition but more complicated functions are being investigated.

(a) (b)

**Figure 7. Talking for a Cartoonish 3D Head in a Neutral (a) and Happy (b) Mood**

iFACE supports two types of moods each with a zero to one activation level:

- Standard emotions (joy, sadness, surprise, anger, fear, disgust) [Ekm78]
- 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 [Rus80] as shown in Figure 8 where horizontal and vertical dimensions are Stress and Energy, respectively.

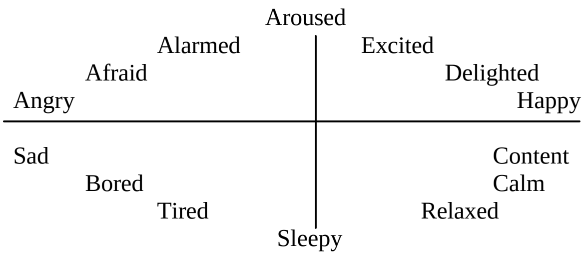


**Figure 8. Russell's Circumplex Mood Model**

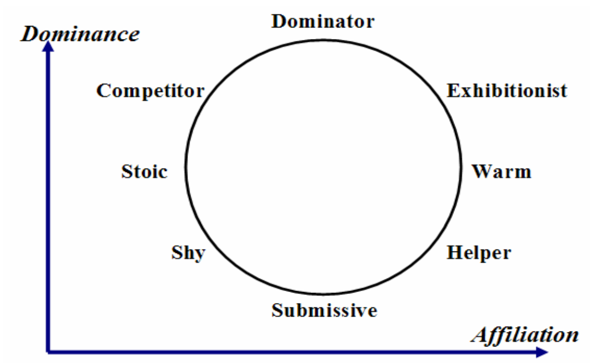


**Figure 9. Personality Circumplex Model**

## Face Personality

Interpersonal Adjective Scale [Wig88] is a widely accepted personality model that links different personality types to two Affiliation and Dominance parameters in a two dimensional Circumplex model (Figure 9). Facial actions and expressions are shown to cause perception of certain personality traits [Bor92, Knu96].

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

For each one of personality-related facial actions, 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. Sample associations between visual cues and perceived personality types are shown in Table 1. These are the results of an on-going study which is the subject of another paper.

| **Visual Cue** | **Perceived Personality Type** |
|---|---|
| Happiness and surprise | high in dominance and affiliation |
| Anger | high in dominance and low in affiliation |
| Sadness and fear | low in dominance |
| Averted gaze | low affiliation |
| Looking away (turning) | low affiliation |
| Raising head | high dominance |
| One-sided eyebrow raise | high dominance |
| Tilted head | high affiliation |
| Lowering head | Low dominance |
| Frequent blinking | low dominance |

**Table 1. Visual Cues and Personality Types**

## 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. 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. iFACE 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:

- Data: main objects such as Geometry and Behaviour
- Stream: extra objects for streaming
- Wrapper: form controls to simplify the use of face functions

Current version of iFACE supports only 3D head models and animation. Full support for the hierarchical head model on top of 2D data is under development. The head models are initially created using traditional 3D software and then can be customized using iFaceStudio software which also helps define the regions and components, and configure parameters (e.g. FAP maximum displacements), and also define FAP combinations corresponding to higher-level parameters. iFaceStudio can also be used to create animation files with FML commands or key-frames defined by parameter values.

# 4. EXPERIMENTAL APPLICATIONS WITH IFACE

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

## COMPS

Simulation and Advanced Gaming Environment for Learning (SAGE, http://www.sageforlearning.ca) initiative is a joint project among Simon Fraser University and some other Canadian universities. Collaborative Online Multimedia Problem-based Simulation Software (COMPS) is a system being developed by SAGE to support online Problem-based Learning (PBL) for medical students. PBL works by introducing students to a case (problem), giving them some facts, and taking them through cycles of discussion and hypothesizing until the learning objectives have been met. iFACE is used in COMPS to represent a remote instructor and simulate patients.

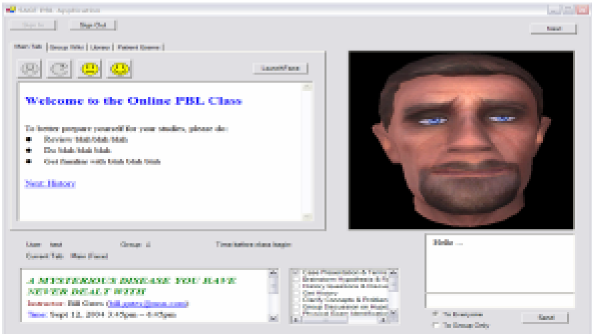


**Figure 10. COMPS with a Simulated Patient**

## 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, as illustrated in Figure 11.

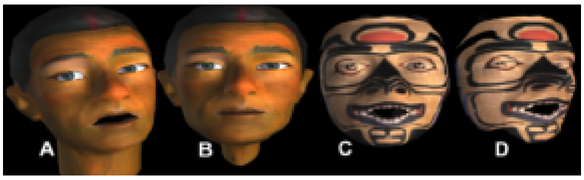


**Figure 11. Frames from Storytelling Masks**

## 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.


**Figure 12. Screenshot of Evolving Faces**

 ISBN 80-86943-03-8

## 5. CONCLUSION

In this paper, we introduce the 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.

iFACE is fully compatible with MPEG-4 standard. Although higher level parameters exist, they are all translated to MPEG-4 FAPs before being performed. The system uses a dedicated XML-based language which provides all necessary scripting functionality in addition to normal XML Document Object Model (DOM).

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. More information on iFACE system can be found at:
http://ivizlab.sfu.ca/research/iface
http://www.imediatek.com

## 7. REFERENCES


[Ary04] Arya, A., and DiPaola, S. 2004. Face As A Multimedia Object, In *Proceedings of WIAMIS*, Lisbon, Portugal, April 21-23.

[Bat99] Battista, S., et al. 1999. MPEG-4: A Multimedia Standard for the Third Millennium, *Multimedia*, vol. 6, no. 4, IEEE Press.

[Bor92] 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.

[Byu02] 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.

[Cas94] 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.

[Cas01] Cassell, J., et al. 2001. BEAT: The Behavior Expression Animation Toolkit. In *Proceedings of ACM SIGGRAPH*, ACM Press.

[Dec02] Decarlo, D., et al. 2002. Making Discourse Visible: Coding and Animating Conversational Facial Displays. In *Proceedings of Computer Animation Conference*, IEEE Press.

[Ekm78] Ekman, P. , and Friesen, W. V. 1978. *Facial Action Coding System*, Consulting Psychologists Press Inc.

[Fid02] Douglas Fidaleo, D., and Neumann, U., 2002. CoArt: Co-articulation Region Analysis for Control of 2D Characters, In *Proceedings of Computer Animation*.

[Fun99] Funge, J. et al. 1999. Cognitive Modeling. In *Proceedings of ACM SIGGRAPH*, ACM Press.

[Knu96] Knutson, B. 1996. Facial Expressions of Emotion Influence Interpersonal Trait Inferences, *Journal of Nonverbal Behavior*, 20, 165-182.

[Ost98] Ostermann, J. 1998. Animation of Synthetic Faces in MPEG-4. In *Proceedings of Computer Animation Conference,* IEEE Press, 49–55.

[Par72] Parke, F. I. 1972. Computer Generated Animation of Faces. In *Proceedings of ACM Annual Conference*, ACM Press, 451–457.

[Par00] Parke, F. I., and Waters, K. 2000. *Computer Facial Animation.* A. K. Peters.

[Pas01] Pasquariello, D., and Pelachaud, A., 2001. Greta: A Simple Facial Animation Engine. In *6th Online World Conference on Soft Computing in Industrial Applications*, Session on Soft Computing for Intelligent 3D Agents.

[Pel03] Pelachaud, W. and Bilvi, M. 2003. Computational Model of Believable Conversational Agents. In *Communication in MAS: Background, Current Trends and Future*, Marc-Philippe Huget (Ed), Springer-Verlag.

[Rou97] Rousseau, D., and Hayes-roth, B. 1997. Interacting with Personality-Rich Characters. *Knowledge Systems Laboratory Report No. KSL 97-06,* Stanford University, September 1997.

[Rus80] Russell, J.A., 1980. A Circumplex Model of Affect. In *Journal of Personality and Social Psychology*, 39, 1161-1178.

[Smi04] Smid, K. et al. 2004. Autonomous Speaker Agent. In *Proceedings of Computer Animation and Social Agents*, Geneva, Switzerland.

[Wig88] Wiggins, J.S. et al. 1988. Psychometric and Geometric Characteristics of R-IAS, *Multivariate Behavioural Research*, 23, 517-530