

Simulating Face to Face Collaboration for Interactive Learning Systems

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

John Chan
Simon Fraser University, Canada
cchany@sfu.ca

Ali Arya
Simon Fraser University, Canada
aarya@sfu.ca

Abstract: The use of Problem-Based Learning (PBL) in medical education and other educational settings has escalated. PBL's strength in learning is mostly due to its collaborative and open-ended problem solving approach. Traditional PBL was designed to be used in live team environments rather than in an online setting. We describe research that allows for web-based PBL via geographically distributed physical locations that emphasize PBL's collaboration and open brainstorming approach using interactive web, gaming and simulation techniques. We describe Interactive Face Animation - Comprehensive Environment (iFACE) which allows for expressive voice based character agents along with Collaborative Online Multimedia Problem-based Simulation Software (COMPS) which integrates iFace within a customizable web-based collaboration system. COMPS creates an XML-based multimedia communication medium that is effective for group based case presentations, discussions and other PBL activities. We also discuss our medical school prototype that allows for brainstorming sessions, remote instructors and simulated patients.



Figure 1: Instructor remotely delivering web based collaborative content in COMPS while controlling the expressive iFACE face agent via speaking through his computer microphone in real-time.

Introduction

In a traditional learning scenario, facts are presented to the students while in Problem-Based Learning (PBL) (Barrows, 2000), problems are presented instead. Students acquire new knowledge and problem solving skills from hypothesizing and validating their hypothesis by going through reference materials, group discussions and presentations facilitated by the instructor. The responsibilities of the instructor has been changed from presenting facts to a mix of presenting related materials, mediating discussions, and customizing learning activities (group discussions, presentations etc.) based on students' progresses in order to achieve the learning objectives. The roles of the students have also changed from passive receivers to participants who actively engage in the problem solving process.

Over the last decade, a number of PBL tools have been developed by educators and computer scientists. However, most of them are designed to be used in a face-to-face environment rather than in an online setting in which the students and the instructor are in geographically distributed physical locations. In addition to presenting static contents efficiently and clearly as in most online learning tools, an online PBL tool needs to provide a communication medium that is effective to be used for case presentations,

discussions and other group activities in real time. From the instructor's standpoint, the system has to allow her to pre-author the flow and the activities of a PBL case study while allowing her to dynamically rearrange the flow or inject new materials or activities during the class. On top of all these functionalities, a system has to be user friendly to both the students and the facilitators to be worthwhile.

Although free of the above mentioned problems, face-to-face PBL has its own challenges to be overcome. Since most of the learning occurs during the process of a PBL session (e.g. during a group discussion), it is not as easy to record the learning outcomes as in static materials like text. Voice recording for the class can be used to archive a discussion session but it is hard to extract or search for specific information from the voice data.

iFACE System

Social user interfaces (Cassell 1994, 2001) with computer-generated characters or "agents" are major topics of graphics, multimedia, and interactive systems research. Advances in computer graphics and multimedia technologies have had a significant impact on the design of a variety of systems. Considering the key role of "communication" as a main aspect of many computer-based applications, such advances make it possible to have audio/visual interaction with users. Faces are sources of considerable verbal and nonverbal information and require special attention when designing multimedia presentations involving human communication (Parke and Waters, 2000, DiPaola 2002). This includes appearance (realistic or stylistic) and behaviour (rules of interaction, personality, mood, and general "expressiveness") for such characters.

The authors have proposed Face Multimedia Object (FMO) (Arya and DiPaola 2004) as a more systematic approach to encapsulate face functionality into one autonomous but controllable object. FMO can be considered as a "face engine" for developers who need face-based multimedia content for applications such as online services, games and elearning. FMO uses 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. Facial Geometry Regions and SubRegions allow grouping of head data into parts that perform specific actions together (e.g. resizing the ears or closing the eye). Facial SubRegions are shown in **Figure 2**.

In this paper, we report on FMO implementation within our proposed framework called iFACE (Interactive Face Animation – Comprehensive Environment). iFACE is based on a hierarchical parameterized head model that can effectively control the face geometry (2D or 3D), and behavioural extensions in form of Knowledge, Mood, and Personality spaces (see **Figure 3**). iFACE provides an XML-based scripting language, object model and API (Application Programming Interface), and an authoring/design tool (iFaceStudio). The framework is used within the context of SAGE (Simulation and Advanced Gaming Environment) project of Simon Fraser University to create a "game-like" online learning tool for medical students (see **Figure 1**). SAGE utilizes iFACE to replace live-action patient footage and real instructors with Social Conversational Agents (SCAs).



Figure 2. Face SubRegions. These are small areas that usually move together and are controlled by FeaturePoints. iFACE Regions are related groups of these SubRegions, e.g. eye area.

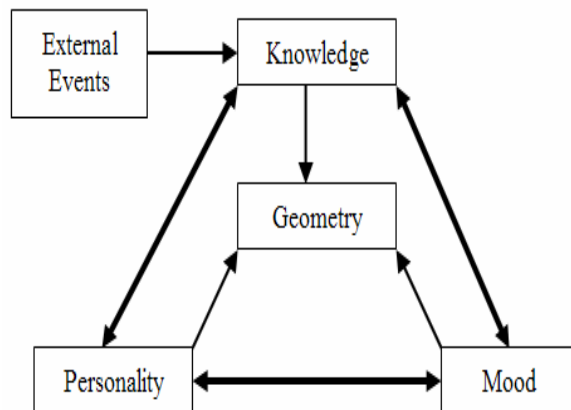


Figure 3: iFACE Parameter Spaces

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) (Arya and Hamidzadeh, 2003) 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. The 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.

Although scripts can select a new personality or modify the mood, 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, facial actions are modulated with the current mood of the character. At the moments, supported moods correspond to MPEG-4 (Battista, 1999) emotions.

There are also a set of head movements defined for each character type, among them one is selected randomly when energy reaches the defined threshold. These movements correspond to character type we are designing (e.g. slow vs. sharp, calm vs. neurotic). We use ETCCodec lip-sync module by OnLive (DiPaola and Collins, 2003) that calculates a speech energy for each audio frame of 60mSec. ETCCodec also gives values for mouth shape which are translated to MPEG-4 Facial Animation Parameters (FAPs) by iFACE. This setup up give iFace the ability to use Voice over IP (VoIP) both to send compressed voice over the internet to a client application as well as to synch with an expressive and emotive lip-sync 3D head. Note how in Figure 1, the remote instructor uses his computer headset microphone to talk with remote students in real-time, while the iFACE software analyses his voice stream and renders 3D face model (the African-American female) that mimics his expressions. iFACE is developed as a set of .NET components written with C# and is designed to work with a variety of client types.

Collaborative Online Multimedia Problem-based Simulation Software

Here we outline an experimental application in the context of Simulation and Advanced Gaming Environments (SAGE), describe how the application employs iFACE, and demonstrate the capabilities of iFACE within the context of an interactive online learning environment.

SAGE-COMPS Project

Simulation and Advanced Gaming Environment (SAGE) (Kaufman and Sauve 2004) initiative is a joint project among Simon Fraser University and some other Canadian universities. Among the areas of research in SAGE are e-learning tools in general, and Problem-Based Learning (PBL) [Barrows, 2000] in particular. Collaborative Online Multimedia Problem-based Simulation Software (COMPS) is a system being developed by SAGE to support online 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. A typical flow of a PBL scenario is shown in **Figure 4**.

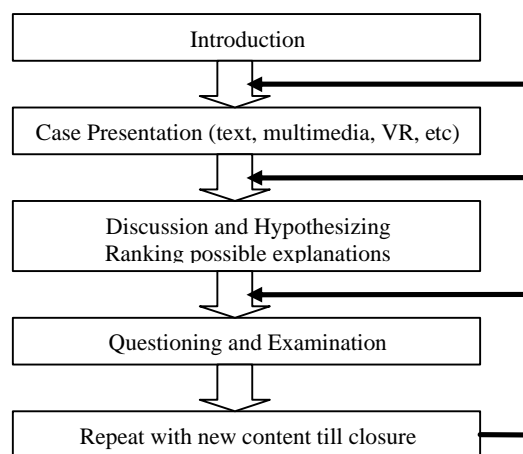


Figure 4: Problem-Based Learning



Figure 5: Screenshot of COMPS User Interface with a Simulated Patient

Although free of the above mentioned problems, face-to-face PBL has its own challenges to be overcome. Since most of the learning occurs during the process of a PBL session (e.g. during a group discussion), it is not as easy to record the learning outcomes as in static materials like text. Voice recording for the class can be used to archive a discussion session but it is hard to extract or search for specific information from the voice data. Additional procedure of transcribing the recording to text is thus needed which poses additional time and monetary cost for running PBL cases. In medical schools, where PBL is widely adopted, it is difficult for students and practitioners, who have busy schedules, to sustain meeting in person across sparsely located clinical sites. Also, it is often costly to invite content experts from other countries to facilitate a specific PBL case. Real and actor patients' interviews are also expensive and time consuming. COMPS is designed to try to overcome these challenges faced by online PBL tools and at the same time ease the burden of implementing face-to-face PBLs from medical schools.

COMPS User Interface and Components

Figure 5 shows a screenshot of COMPS client's user interface. It is designed as a .NET form that includes an iFACE wrapper control working like a picture box but displaying the animation.

The COMPS client's interface is a single window that consists of several panels. The single window design avoids clustering of UI elements, shows all functionalities at a glance and frees users from the task of multiple windows management. User can resize the panels to fit their needs. The panels as illustrated in the figures are:

Information panel

- Agenda panel
- Content panel
- Resource switch panel
- Face panel
- Text chat panel
- Menu

The panels' functionalities and responsibilities will be discussed in the next section.

COMPS is a complete software platform that includes several backend servers, a client application and some helper tools. The servers are responsible for supporting the functionalities of the client such as authentication, managing multiple PBL sessions, and synchronizing activities among students. The flow of a PBL lesson is stored in the case server as an XML script. The XML case script is designed in a way that is easily understood and can be authored or modified either by using Windows notepad or by the COMPS case script authoring tool that makes the task straight forward. In COMPS, a PBL lesson is broken down into PBL stages. A stage contains a sequence of activities such as case presentation or discussion that are grouped together to represent a milestone in the learning progress. The flow of a PBL study is constructed by going through a sequence of PBL stages.

COMPS Experience

When a PBL lesson starts, the instructor and the students launch the COMPS clients and login to the server. The server looks up the login user's record and presents a list of courses for the user to select from. Based on the user's login credentials, the system verifies the identity and the role of the user (whether she is a student or a facilitator), enabling and disabling functionalities for the user. Now, the students and the instructor in the same class are connected to the same case server instance and the lesson is ready to start or continue from where the last session ended. Students are divided into groups for different class activities and they can see who are in the class or in their own groups at anytime from the information panel. They can always check vital information related to the class, the PBL case, and the current facilitator.

Let us consider the following case. Students and the facilitator review the PBL stage that they deliberated in an earlier session where they were presented with a patient with a liver problem. They had participated in activities including a question-and-answer session to discover that the patient also has depression problem. The facilitator does not expect the students to find out the patient's mental problem so early but since they do, she decides to skip some prescribed PBL stages. She clicks a menu item and selects the PBL stage titled 'mental problem' to jump to. All students see the agenda has been updated and the content panel has new information about depression in late adolescent. The current activity has now been identified as 'case presentation'. The face panel now shows a depressed face. The instructor speaks through a microphone talking about how long the patient has had the depression and the patient rants about her irregular menstrual cycle. Every student watches the face in the face panel talk with facial expressions mimicking that of a real depressed person. The instructor presses a button and everyone notices it is a group discussion time from the agenda panel. The content panel shows that they have to separate into groups and try to come up with a list of questions to interview the patient with. The students then perform the group discussion through the text chat panel or through the built in Voice-Over-IP/Internet Telephony (VOIP) functionality. They turn on the privacy mode of their own group channel so that other groups cannot hear their discussions. The instructor can now choose different channels so that she can observe students' discussions and give guidance if needed. Students are provided with additional resource such as access to sample video clips of patient interview. They can access the resource by clicking on a button in the resource switch panel and the clip is shown in the content panel. They can always switch back to the discussion topic by clicking on another button in the resource switch panel. Now that 15 minutes have passed and the instructor clicks a button again and now everyone is at the interview activity. The groups can individually interview the patient using the text chat panel because things in the text chat are saved by the chat server and will be available for reference later. The instructor, who prefers to pretend to be the patient for a short time, responds to some of the questions in the interview sessions, revealing new pieces of information that the patient's father and one of her brother also have liver and mental problems. While students contemplate the new piece of information, the

instructor decides that it's time to add an additional activity to the current stage. She clicks on a menu item, types in the name and a short description of the topic, asking the students to come up with some laboratory tests to help diagnose the patient. Then, she types in the URL of a web site that contains some laboratory test reference material and clicks OK.

COMPS Advantages

As evident in the above case snippet, COMPS presents a user-friendly, learner-centric, instructor-driven, and context-rich problem-based learning experience. Most of the functionalities are a click or two away. The use of VOIP provides a natural means for communication. The use of iFace in COMPS enhances the expressive power of the instructor to students to an unmatched degree, in comparison with a voice only or a text only communication. iFace's capabilities of customizing facial features and emotion allow the facilitator to assume multiple roles representing multiple people, which is quite difficult to achieve with webcam technology. For example, instead of hiring an actor or bringing in a real patient to the class who has serious facial injuries, iFace can comfortably simulate such requirements at a low cost. The selection of open or group voice or text chat enhances group dynamics in PBL. Multiple level text chat archiving (for the whole class and for individual groups) provides traceability and enables searching of important discussion events that are not readily available in face-to-face situations. The content panel is an embedded browser that not only makes multimedia presentations readily available but can also be used to provide means for asynchronous communication such as hosting a forum, and group publishing functionality like hosting a Wiki. The ability for a case facilitator to jump to any PBL stage and inject new activities during the class allows the facilitator to dynamically control the flow of a PBL session. The use of online learning tool reduces travel time and allows content experts to facilitate case with students who are geographically distributed all over the world. This also promotes cultural interactions.

Simulated Patients

A major requirement in 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 to an 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 (Decarlo 2002). SCA can also be a substitute for "automated instructor" and can represent a remote instructor (or patient). Such a substitution may not be possible when transmitting real-time video but SCA can be animated based on real audio data. Here, we briefly review two examples, first a "simulated patient", and second a "remote instructor". They exemplify the technological know-how of how iFACE co-exists within COMPS.

Simulated Patient: An FML script file is the 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 the iFACE Section. 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.

Remote Instructor: A simpler mechanism for controlling iFACE animation is to provide only the audio data as input as illustrated in the above discussions. 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 other (or group of) remote iFACE objects which in turn use the data to drive the animation. Again proper personality and mood can be selected.

The face's outlook and emotional status can be changed from activity to activity. How the face changes is determined by the case script. When the instructor authors the case script, all the resources including the face meshes and behavioural specifications, are specified and assigned an ID at the scripts global section. At each activity specification, parameters for iFace are specified by referencing the ID. As the user logs in, all the necessary resources will be downloaded to the user's machine. COMPS will then instruct iFace to use specific parameters when it starts a new activity.

We have just described how iFace works from within COMPS. We have seen how iFace plays a significant role in COMPS, how it enhances the learning and teaching experiences, and how it helps overcome some of the challenges faced by online PBL tools today.

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.

We also outline COMPS, an experimental application in the context of Simulation and Advanced Gaming Environments, describing how COMPS employs iFACE, and demonstrate the capabilities of iFACE within the context of an interactive PBL environment. Future directions for COMPS includes VOIP for all participants (it is currently limited to the facilitator, while other participants use text chat), integration of other multimedia applications for simulated anatomy, and medical procedures, as well as a simpler authoring environment for non-technical medical educators.

References

- Arya, A., & DiPaola, S. (2004). Face As A Multimedia Object. 5th International Workshop on Image Analysis for Multimedia Interactive Services
- Arya, A., & Hamidzadeh, B. (2003). ShowFace: A Framework for Personalized Face Animation. In Proceedings of IEEE/EURASIP RichMedia-2003, Lausanne, Switzerland, IEEE Press.
- Barrows, H. (2000). Problem-Based Learning Applied to Medical Education. Springfield, IL: SIU School of Medicine.
- Battisa, S., et al. (1999). MPEG-4: A Multimedia Standard for the Third Millennium, Multimedia, vol. 6, no. 4, IEEE Press.
- 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.
- Cassell, J., et al. (2001). BEAT: The Behavior Expression Animation Toolkit. In Proceedings of ACM SIGGRAPH, ACM Press.
- Decarlo, D., et al. (2002). Making Discourse Visible: Coding and Animating Conversational Facial Displays. In Proceedings of Computer Animation Conference, IEEE Press.
- Dipaola, S. (2002). FaceSpace: A Facial Spatial-Domain Toolkit. In Proceedings of InfoViz 2002, 49–55.
- DiPaola, S. & Collins, C. (2003). "A Social Metaphor-based 3D Virtual Environment", ACM Siggraph 03, (Education).
- Kaufman D, Sauve L (2004). Simulation and Advanced Gaming Environments for Learning, Proceeding of the Ed-Media Conference.
- Parke, F., and Waters, K. (2000). Computer Facial Animation. A. K. Peters.