IBM Talking Heads

TECHNICAL FINAL REPORT

This task was submitted in cooperation with IBM Research Center, CZ

Submitter: Prof.Ing. Pavel Slavík, CSc.
Solvers: Charvát Michael
          Chrastinová Adéla
          Kunc Ladislav

1 Problem

The goal of our semestral project was to study, what for freely available „talking head SDK and libraries exist. To choose one of them and do some experiments, that prove extensity and quality of their functions.

Further to make network interface, so the head control (text, emotions, movements) is available by TCP/IP protocol.

We should design the interface to allow connection of speech syntethiser.

2 Retrieval

The main task was choosing a suitable tool for Talking Heads realisation. Our priorities was an open licence and source codes availability, possibility of connecting with IBM voice synthetiser and the head shape visage first of all.


Unknown license conditions, possibly very expensive. Unavailable source programs, unknown HW&SW requirements. Among tools belong AV Tools and 3D Talking Head. For animation uses parametrically deformed 3D polygonal models. Flexible architecture allows creating of new faces by reading of some static 3D wireframe. Supports call-reply systém and whole head moving, possibility to show inner mouth during talking.

This Toolkit is suitable for audio-handicapped people, but we found it unsuitable for our task.

Free license for university studies, but RUTH face can be used just for RUTH environment. Interface implemented in C++ language with OpenGL, uses the GLUT library. It is a multiplatform application (Solaris, Linux, MS Win). Tools: TTS (Festival) for synthetic voice, allows face animation export into a file, allows displaying of animated faces. Face animation can be done with a simple 3D polygonal mesh deformation consists of triangles. Allows lips movement and speech synchronization, allows eye-movement.

Unfortunately its a closed project now, so its also unsuitable for our task.

2.3 [ BALDIE ] http://cslu.cse.ogi.edu/toolkit/

Free license for university studies, but cannot be fully controlled, just in a users level, we cannot use the source codes. Interface uses Tcl script, which can be used for the scripts turn on. The scripts can change and create emotions or change heads. There are four heads, but mostly plain. Implemented in C language, Visual C++, OpenGL libraries, under MS Windows. Among tools belongs AV Tools, speech recognising, TTS (Festival), RAD developer. Animated faces (Baldie, CU Animate), but we cannot create another faces, we miss source codes for CU Animate release. We are able to conect face + audio + TTS synthetiser (but its non-transparent, we cannot modify it), the speech synthetis is very well done, if there is a synthetiser interface, we are able to connect them and then we are able to use it, but in spite of it we are not able to satisfy all the IBM claims.

This Toolkit can be used for the whole-deaf people to exercise the speech, but not for our task.

2.4 [ XFACE ] http://xface.itc.it/

There is just one head and most probably cannot move (there exist just a frontal look). There is a possibility of creating new haed out of VRML models (but it is very time-consuming). It has an Xface Player, which could be used as a server, and a Xface Client, which communicates with server and sends him visemes through the scripts. Based on a technology, which uses mpeg4.

This Toolkit is also unpracticable for our task.

2.5 [ EXPRESSION ] http://expression.sourceforge.net/

This Toolkit is completely OpenSource. Has two available and a rather good-looking heads sculptured. Its one male and one female head. Has very sophisticated graphics and a group of functions for muscles, mimicry, emotions, head and eyes movement control. Allows creating of some new heads with a Poser programm, with a .msh (mesh) conversion in 3DMax’s (but creating a new face and defining of 14 new muscles, its very time-consuming, suitable for some new, further task for somebody elses project). EBBA – an independent project, which uses Expression core, completely OpenSource. There is a possibility of some other speech synthetiser connection. We could be able to connect the IBM synthetiser.

This Toolkit is the most suitable for our task processing.

3 Implementation

3.1 Implementation Concept
As a development platform we used Windows, development tool Microsoft Visual C++ 2003 .NET, because of chosen Toolkit Expression, which is written for this platform.

Toolkit Expression use for head visualization pseudo-muscle model, there is rendered mesh, which represents head, which is also deformed by parameters of six types of muscles. Eyelids and jaw movements are controlled apart. Expressions are represented in the separated files as sets of parameters. Toolkit contains big quantity of these expressions, implemented expressions will be named in the description of possible commands for method ACT. Expressions are executed by events. Each Expression is a facial pose, which can be transitioned to with another facial pose. The animation system in effect is a state machine, with every new animation event is transitioned from the previous one.

Models of head, which are rendered by the system are loaded in format .MSH, for which exists export plugin to the systém 3D Studio MAX, source code of this plugin was attached to our project. Description of head muscles can be also exported by scripts in 3D Studio MAX.

We used following simple diagram in the implementation of our toolkit extension, its „heart“ is toolkit Expression. You can clearly see that the systém which renders head will be so called server that will get work from clients (theoretically more than one) and this server will execute these commands. Clients connect to the server by TCP/IP interface, so from anywhere in the Internet.

![Diagram of connection](image)

We got offered library ChiliX from IBM that can solve our connection problem by remote procedure calls. It can be used in the Java and C++ development environments.

There will be also connected speech synthetiser to the server, we will get each sets of phonemes from it, according to them we will render each face expressions, whose are represented by them.

Clients will send to the server commands in the form of two methods:
- **ACT** – This method is more difficult than the second and it will take care of commands, which turn head, eye positioning or rendering expressions. Method input is text string.
- **SPEAK** – This is simple method, which will be used for string input, that head can speak. The input will be a sentence.
3.2 Syntax Of Input Of ACT Method

Input of this method will be formatted this way:
KeyWord:Value;

Group of keywords is created of these simple words: PrimitiveExpression, CompoundExpression, Mood, TurnHead angle, TurnEyes v_angle TurnEyes h_angle.

- **PrimitiveExpression** – it means command for rendering of simple (primitive) expression, value is one of these words:
  
  neutral, turn right, turn left, bend forward, bend backward, tilt right, tilt left, smile, open mouth, anger, surprise, sadness, fear, disgust, left blink, right blink, close eyes

- **CompoundExpression** – tells system to render more complicated expressions, those values are following:
  
  yes, quick yes, no, sceptical no, laugh, lookaround, fear lookaround, whistle, roll head, sleep.

- **Mood** – this keyword commands head mood, values are:
  
  happy, sad, tired, angry, skeptical, scared.

- **TurnHead angle** – Here we can set-up turning of head by angle, which is handed over as value. Angle can be in interval of values -90...90 degrees. Negative values represent turning left and positive represent turning right.

- **TurnEyes v_angle TurnEyes h_angle** – This doubled command is there for setting eye position. Its values are also angles, now in interval -45...45 degrees. Eye position is controlled by two angles horizontal and vertical. These angles are in context of head turning, so they represents own system of coordinates.

3.3 Head Speaking

As the speech synthesizer, which we decide to connect to our system, we choose library Microsoft SAPI. We compiled this library into our system and we awaited from it events.

Visualization of each phoneme is done by so called visemes. There exist 22 visemes (expressions of face), which represent each group of phonemes or phonemes. If our head speaks, visemes are rendered in the same sequence as phonemes. This system is also used in animated film.

Synthesizer hand over group of phonemes in the form of events, which are plotted in the loop and they are also interpreted as command to the head for showing of expression, that represents viseme.

3.4 Head Choice

Head choice is made by a parameter on the command line, to our program are attached three head types – Sarah, Gedalia and Calle. By a parameter we also choose port on that the server will be listening. You can see heads and examples of expressions on the bundled screenshots.
3.5 Client Implementation

Client is implemented in the form of simple program which gets as parameter script file, whose example can be seen in the part Testing, which interprets and sends each command to the server. Syntax of the script is nearly the same as the syntax of ACT method. Only one keyword is added, Sleep, whose value is number of seconds, this means how long must client sleep before sending following command. Client of course uses ChiliX library.

3.6 Problems

3.6.1 ChiliX Library Problem

From the first moment of getting and finding out function of connection library ChiliX by IBM company we were challenging many problems. First we cannot run sample programs, by course of that we should put this library to our project. This was partly solved by IBM company, but other unexpected problems arised, whose solving took our big part of time (debugging and setting up
compiler). Fortunately we could use this library in the end.

3.6.2 Java Client Problem

Original intention was implementation of client in the language Java, but that could not be realized because of bug on the client side in the time of sending any action to the server, probably badly generated interface of ChiliX library. So we implemented client part in the language C++, that works without problems.

3.6.3 SPEAK Method Problem

Fundamental error of the interface of method SPEAK (generated from ChiliX library) is that it does not work. This problem we bypassed by adding keyword `Speak` to the ACT method, in whose value we hand over sentence, which head should speak.

4 Testing

4.1 Connecting With Client

Part of implemented functions was tested by several scripts, that were loaded by client and were pre-parsed and sended to server. We verified functionality of connection of client and server and also parser function, that processed incoming commands on the server side. All of tested and implemented functions for positioning eyes and head, also i embedded functions for setting-up primitive and compound expressions, they behaved so, that we were satisfied (as defined in scripts).

4.2 Example Of One Of The Submitted Scripts

```
PrimitiveExpression: neutral;

TurnEyes h_angle:0; TurnEyes v_angle:0;
TurnHead angle:0;
Sleep:2;
TurnEyes h_angle:5; TurnEyes v_angle:5;
Sleep:2;
TurnEyes h_angle:10; TurnEyes v_angle:15;
Sleep:2;
TurnEyes h_angle:20; TurnEyes v_angle:20;
Sleep:2;
TurnEyes h_angle:-5; TurnEyes v_angle:-5;
Sleep:2;
```

4.3 Priority Of Server

There was a small problem in the communication client-server, just in the time of link-up, which shows like neverending stop of the client and resulting server stop. From this state we cannot activate the client and restart is necessary.

Problem is probably caused by that server eat much CPU-time and the client cannot end communication in normal way.

Considering the problems with ChiliX library, which led to a large time delay we could not
implement the client on the platform Java and server, that is implemented in C++. Instead of that we implemented 100% functional client on the platform C++.

5 License
We release the whole project under conditions of GNU GPL. Server that renders talking head cannot be probably released under conditions of GNU GPL, because it would be against Qt Public License, under it was released the core of our server Toolkit Expression. We leave Qt Public License that would be compatible with GNU GPL.

6 Conclusion And Future Usage Possibilities
We managed to extend the Expression Toolkit with functions for eye-positioning and head-turning. We also developed text interface API for possession of the head by ACT method. We did not succeed in using method SPEAK because of the ChiliX library malfunctions.

We connected Toolkit with MS SAPI synthesizer and we achieved synchronized movements of lips with spoken words.

We implemented client on the C++ platform and we connected client-server through the ChiliX library.

For future work it is possible to model new heads in a common way. Project is now prepared for a possible connection with IBM synthesizer, also with possibility of implementation of some more accurate lips synchronization (generating of new expressions during the program run).

7 References
- [KTH] http://www.speech.kth.se/multimodal/
- [xFace] http://xface.itc.it/
- Wojdel Anna: Knowledge Driven Facial Modelling (Thesis, Technical University of Łódź,)
- Martin Šimůnek: Face Animation (Thesis, CTU Prague)