Published in Proceedings of the 6th International Conference on Intelligent Virtual Agents IVA 2006

[HUGE]: Universal Architecture for Statistically Based HUman GEsturing

Karlo Smid[1], Goranka Zoric[2], Igor S. Pandzic[2]

[1]Ericsson Nikola Tesla, Krapinska 45, p.p. 93, HR-10 002 Zagreb
karlo.smid@ericsson.com
[2]Faculty of electrical engineering and computing, Zagreb University, Unska 3, HR-10 000 Zagreb
{Igor.Pandzic, Goranka.Zoric}@fer.hr

We introduce a universal architecture for statistically based HUman GEsturing (HUGE) system, for producing and using statistical models for facial gestures based on any kind of inducement. As inducement we consider any kind of signal that occurs in parallel to the production of gestures in human behaviour and that may have a statistical correlation with the occurrence of gestures, e.g. text that is spoken, audio signal of speech, bio signals etc. The correlation between the inducement signal and the gestures is used to first build the statistical model of gestures based on a training corpus consisting of sequences of gestures and corresponding inducement data sequences. In the runtime phase, the raw, previously unknown inducement data is used to trigger (induce) the real time gestures of the agent based on the previously constructed statistical model. We present the general architecture and implementation issues of our system, and further clarify it through two case studies. We believe that this universal architecture is useful for experimenting with various kinds of potential inducement signals and their features and exploring the correlation of such signals or features with the gesturing behaviour.

1. Introduction

Gesturing is integral part of human face-to-face communication. In parallel to the production of gestures in human behaviour other signals may occur, e.g. text that is spoken, audio signal of speech or bio signals. In most cases these signals and gestures happen simultaneously based on common motivation and they supplement, complement or duplicate each other. Those signals may be in a statistical correlation with the occurrence of gestures. We use this fact to build the statistical model of facial gestures based on any kind of such signal. In our system we call such signals *inducement*. Therefore, we think of inducement as any data sequence that happens in parallel with generation of gestures, and that is expected to have some correlation with the gestures. The statistical model is built from a training corpus consisting of gesture sequences and corresponding inducement data sequences. For every state of inducement, a statistical model generator produces statistical data for every gesture type with probabilities for gesture type, duration and amplitude values. In the runtime

phase, the statistical model for gesture generation is used to automatically generate facial gestures from raw inducement data. These gestures are then used to produce real time animation corresponding to the underlying inducement. The idea of our system is to provide a universal architecture for the use with various kinds of potential inducement signals correlated with HUman GEstures (thus the name HUGE). Thus by using our system it is possible to generate facial gestures in real time using statistical model driven by different signals occurring in parallel to the production of gestures with no need of changing the complete system architecture, but only adding modules specific for the used signal. We believe that this universal architecture will ease experimenting with various kinds of potential inducement signals and their features and exploring the correlation of such signals or features with the gesturing behaviour.

The HUGE architecture has its roots in our previous work in the field of Embodied Conversational Agents. In [1] we proposed a system architecture for an Autonomous Speaker Agent that is capable of reading plain English text and rendering it in a form of speech accompanied by the appropriate facial gestures. The statistical model is obtained by analyzing a training data set consisting of several speakers recorded on video and transcriptions of their speech. A lexical analysis of the transcription texts allowed to correlate the lexical characteristics of a text with the corresponding facial gestures and to incorporate this correlation into a statistical model. Lexical structure of input text is the inducement that triggers the obtained statistical model of facial gestures. In [2] we presented a method for automatic Lip Sync of graphically embodied animated agents where parameters of speaker's speech signal are used as inducement.

There are many existing systems that generate facial gestures from the single input (e.g. the text, speech) using statistical models or some other methods driven by rules, semantic data and similar. In [3], Albrect et al. introduce a method for automatic generation of facial gestures from speech: head and eyebrow raising and lowering dependent on pitch, gaze direction, movement of eyelids and eyebrows, and frowning during thinking and word search pauses, eye blinks and lip moistening, or random eye movement during normal speech. Poggi and Pelachaud in [4] focused on the gaze behaviour in simulating automatic generation of face expressions driven by semantic data. The Eyes Alive system [5] reproduces eye movements that are dynamically correct at the level of each movement, and that are also globally statistically correct in terms of the frequency of movements, intervals between them and their amplitudes based on the statistical analysis of eye-tracking video. Cassell et al. [6] automatically generate and animate conversations between multiple human-like agents including intonation, facial expressions, lip motions, eye gaze, head motion and hand gestures from the speaker/listener relationship, the text and the intonation. The BEAT system [7] controls movements of hands, arms and the face and the intonation of the voice from the input text, relying on the rules derived from the extensive research in the human conversational behaviour. Graf et al. in [8] analyze head and facial movements that accompany speech and they relation to the text's prosodic structure, where prosody describes the way speech is intonated with elements such as pauses, pitch, timing effects and loudness. Cao et al. in [9] present a technique for automatically synthesizing speech-driven expressive facial animation from given input utterance.

The system is capable of automatic detection of emotional content of arbitrary input utterances by using support vector machine classifier with probabilities with which a given utterance belongs to each of the emotional classes. In addition, lip-synchronization with correct co-articulation is performed. Gutierrez-Osuna et al. in [10] generate animations with realistic dynamics of three-dimensional human faces driven by speech signal represented with perceptually based parameters combined with two prosodic cues (fundamental frequency and frame energy). Similarly in [11], Granström and House use audiovisual representation of prosody to create an animated talking agent capable of displaying realistic communicative behaviour. Brand in [12] generates full facial animation from expressive information in an audio track. An animated agent, in this case called voice puppet, learns a facial control model from computer vision of real behaviour, automatically incorporating vocal and facial dynamics.

As opposed to previously described systems which use a single signal (either text or speech) to produce the animation, our system provides universal architecture that accepts various kinds of potential inducement signals related to facial gestures.

In this section the idea of our proposed architecture is given, as well as the related work. Next section describes in details our universal architecture. Section 3 explains the implementation issues of our system, while it is further clarified through two case studies in Section 4. The paper ends with the conclusion and the discussion of future work.

2. The Architecture

In this section we describe the logical modules and the data flow in the proposed architecture of our HUGE system. The system works in two distinct phases: the statistical model generation phase, and the runtime phase. The statistical model generation phase is typically done offline and may involve some manual steps in analysing the input data. In general, this phase takes the training corpus in form of gesture data and corresponding inducement data and produces the statistical model by correlating the gesture sequence with the inducement sequence. The runtime phase must run in real time and therefore must be fully automatic, without any manual processing of data. This phase takes a new sequence of inducement data and uses it to trigger the statistical model and produce real time animation corresponding to the inducement. We will now describe the two phases in more detail.

Figure 1 describes the statistical model generation and runtime phase. In figure 1 rounded rectangles represent data, plain rectangles represent processes, and arrows represent data flow between processes. The inputs to the statistical model generation are the timed gesture data and timed inducement data. The raw gesture data typically comes in form of recorded video clips of speakers performing natural speech, but it could otherwise come from a feature tracking algorithm or hardware. The raw gesture [1]data is automatically or manually annotated in order to produce timed sequences of

gestures. These sequences are represented in the universal gesture data format that is a part of the architecture, and described in detail in the Implementation section.

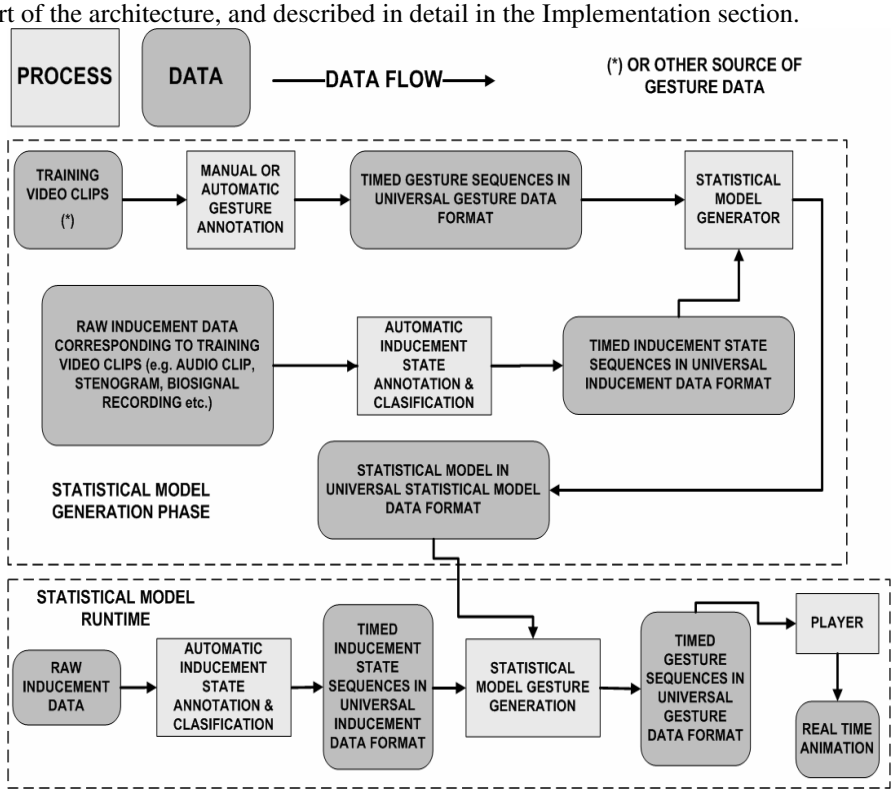

Fig. 1. Universal Architecture for Statistically based HUman GEsturing (HUGE) system.

The raw inducement data is the other input to this phase. For example, textual transcript of a video clip can be used as inducement, as it is expected that gestures will be correlated to the text. Another example is the speech recording, since there is also a correlation between certain speech features and the gestures that happen at the same time. Yet another example may be bio signal measurements.

It has to be noted that we use the term inducement because in the runtime phase of this architecture these inducement signals are used to trigger, i.e. induce gestures. The use of this term is strictly system-related, and does not refer to the way these inducement signals relate to the gesture production in real human behaviour - indeed, in most cases the process is quite different, e.g. speech intonation and corresponding gestures happen simultaneously from common motivation, and not by one inducing the other in any way. Still, within this system we find the term appropriate because it describes quite accurately the way the system works.

From the raw inducement data (e.g. text, speech signal, bio signals), an automated feature extraction and classification algorithm produces a timed sequence of inducement states, stored in the universal inducement data format. An inducement

state can be any state determined from the inducement that is expected to correlate well with production of gestures. For example, raw text may be lexically analysed to determine when new terms are introduced, and thus create a simple classification of states into theme or rheme. In case of audio signal inducement, a set of audio features may be analysed and classified into a number of states. The choice of states, their number, and the algorithm to extract them is the issue of each implementation.

From the timed sequences of gestures and inducement states, the statistical model generator produces the statistical model in the universal statistical model data format. Facial gesture parameters that are described in universal gesture data format are gesture type, duration and amplitude value. Those parameters represent the components of universal statistical model data. For every state of the inducement, statistical model generator produces the probability of occurrence for every gesture type and for particular type it produces the frequency distribution estimation functions for gesture type duration and amplitude values. Statistical model generator correlates the time-aligned inducement and gesture universal data sequences in order to calculate those parameters. The statistical model describes the probability of occurrence and expected amplitudes and durations of various gestures with respect to the inducement states. This is the basis for the runtime phase. The detail explanation of how statistical model generator works is given in the Implementation section.

The runtime phase generates real time animation based on raw inducement data, in such a way that the overall behaviour is similar to the training corpus in terms of statistics of gesture occurrence and characteristics, as related to the inducement. The input to this phase is the raw inducement data (i.e. plain text, speech recording, bio signal sequence). This input is processed by the same automatic states extraction and classification algorithm that was used in the statistical model generation phase. Therefore it is important that this algorithm can run in real time. Again, a timed sequence of inducement states is produced. The gesture generator now uses these states as a trigger to the statistical model in a semi-random process. This process aims to produce the same global statistics for the frequency of occurrence, amplitude and duration of various gestures as described in the statistical model. These gestures may be output in the gesture data form to the player that can interpret them. The player must synchronize the playing of gestures with the reproduction of the raw inducement data (e.g. the generated gestures must be synchronized with the audio reproduction of speech). The modules are presented here in a way that highlights the logical flow of data. The actual implementation may have to integrate some of the modules of the runtime phase because of different interaction needs. For example, if text is used as inducement, there is no temporal reference in the raw input text - the timing is obtained only when the player synthesizes the speech from the text using a speech synthesizer. Therefore in this case the player and the gesture generator must be closely integrated in order to produce gestures on the fly. These issues will be clearly described in the case studies section.

In this section we explained the architecture of our proposed HUGE system. Processes, data and data flows between processes of two system phases are explained in detail. In the following section a concrete implementation of HUGE system will be presented and explained.

3. Implementation

As we explained the architecture of HUGE system in previous section, in this section our goal is to introduce HUGE system implementation issues. First, we will present the universal data formats for gestures, inducements and statistical models along with technologies that we used in order to realize them. After data elaboration, we introduce the HUGE Application Programming Interface (API) model. Using HUGE API, developers are able to realize and connect processes of HUGE architecture. Furthermore, entry points of HUGE API in HUGE architecture will be presented and explained in detail.

For description of universal data formats that exists in HUGE architecture, we chose Extensible Markup Language (XML[2]).
Table 1 shows the example of XML document snippet that is structured and typed according to our universal data formats for inducements.

Inducement row from table 1 holds valid XML document instance snippet that contains inducement data. Identified inducement has three parameters: starting point in time, ending point in time and inducement type. In order to be in accordance with our universal data format for inducement data, XML document instance has to satisfy following requirements:

- File has to be in accordance with the inducement XML schema definition. Schema definition file defines the structure and basic data types (long, decimal).
- Every inducement end time has to be grater or equal than the start time.
- All time intervals have to be sorted in accessing order.
- Interval overlapping is not allowed because we postulate that at the given time interval only one inducement state is possible.

Inducement	<Inducement xmlns="http://www.fer.hr/gestures" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <IdentifiedInducement> <start>0</start> <end>1</end> <type>STATE00042</type> </IdentifiedInducement> </Inducement>

Table 1. Example of XML document snippet for inducements

Every identified gesture has four parameters: starting point in time, ending point in time, gesture type and gesture amplitude value. Gesture universal data has to be in accordance with the following requirements:

- File has to be in accordance with the gesture schema definition file for the same reason as the inducement XML document.
- Overlapping of gesture time intervals is allowed only for the following gesture groups: head movement group (all nod and swing movements, including reset head

[2] http://www.w3.org/XML/

movement) can be overlapped with eyebrows raise and eyes blink. Reason for this behavior is simple: at the same time interval humans could blink, move head in various directions and raise their eyebrows.

- Time constraint on gesture intervals is that start time must be less or equal than the end time.

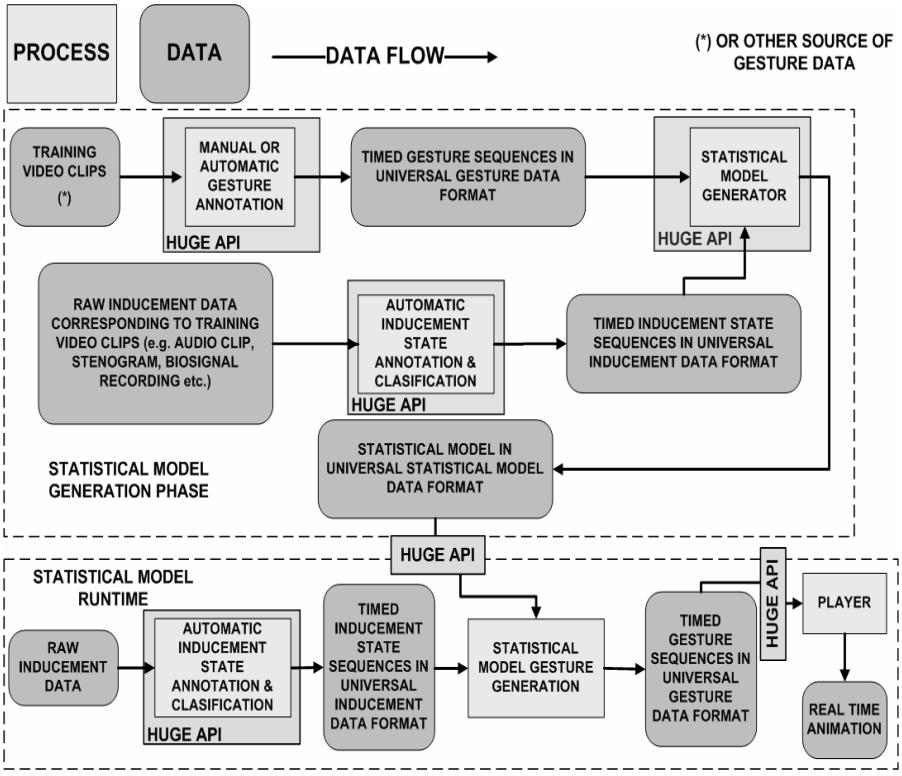

Fig. 2. HUGE API entry points in universal architecture HUGE system

Statistical data are grouped according to identified inducement types. Every inducement statistical data element consists of gesture statistical data elements. Gestures are grouped in four groups: nods, swings, eye blinks and eyebrows raises. It is important to note that we consider head shakes as to consecutive head nods (left nod followed by right nod or vice versa) and that eyebrows frowns are considered as eyebrows raises with negative amplitude. Every sub gesture element has following parameters: type, statistical probability of occurrence value, and cumulative frequency histogram distributions for amplitude and duration parameters. The only exceptions are eyes blink gesture, which does not contain amplitude distribution and overshoot nod, which contains two cumulative histograms of amplitudes. We are using cumulative frequency histogram approximation functions instead of standard frequency histogram because approximation function of cumulative frequency histogram is injection function (one-to-one function). Overshoot nod has two

cumulative histograms of amplitude because it consists of two consecutive nods: nod up immediately followed by nod down. Statistical model universal data has to be in accordance with following requirements:

- It has to be in accordance with schema definition file.
- For every inducement type, total sum of probability occurrences for gesture groups has to be one.
- For every inducement type, total sum of probability occurrences for sub gestures also has to be one.
- All frequency histograms are cumulative.

After the data formats of HUGE system, we are going to explain the HUGE API. These API is realized using c# programming language of Microsoft .NET[3] platform and its purpose is to provide methods for manipulation with HUGE universal data formats and method for calculation of statistical models. Figure 2 shows how HUGE API fits into the universal architecture of HUGE system. It provides methods for parsing, creating and validating XML document instances of gestures, inducements and statistical models. Those methods are based on functionality for manipulating xml data formats provided in .NET platform libraries. This functionality is extended using c# programming in order to provide features needed by proposed HUGE processes and data formats. Using those API methods, application developers that implement their own instances of HUGE system, are in accordance with our proposed system architecture and they are able to use universal data formats for gestures, inducements and statistical models in object-oriented model. Parsing methods transform universal data from XML format into object model. Creating methods operate in opposite direction, they produce XML data format from the data object model. Validating methods validate the correctness of universal data XML format.

It is important to explain the core process of HUGE architecture. Statistical model generator process (figure 2) first uses validation methods that validate inducement and gesture data formats. Inducement and gesture data formats must have the same starting and ending points in time (have to be aligned in time) so the statistical model generator process is able to produce correct statistical model data. It parses inducement data and counts which gesture types were triggered by the particular inducement state. Only gestures that start in time interval corresponded with inducement state are counted. Probability of occurrence for particular gesture type triggered by the particular inducement is simply calculated by dividing number of particular gesture occurrence (e.g. nod up) with total number of gesture group (e.g. nods) occurrences triggered by the particular inducement state. Gesture type duration and amplitude data are grouped in intervals (interval values for duration and amplitude are specific parameters that depend on the amount of the training corpus, more training data means narrower interval) in order to produce their frequency distributions of occurence. Based on cumulative frequency distribution values, we calculate simple linear approximation of probability functions for duration and amplitude values.

$$f(x)=ax+b \quad (1)$$

[3] http://www.microsoft.com/net/default.mspx

Where x is uniformly distributed random number and represents the probability of occurrence for duration or amplitude value.

Creating methods are used by automatic/manual inducement state and gesture annotation and classification processes (figure 2) and also by statistical model generator process because outputs of those processes are inducement, gesture and statistical model XML data formats. Statistical model gesture generation process uses parsing methods in order to transform statistical model data from XML format into object model. It also uses validation methods to check the correctness of statistical model XML data format. Player process uses methods for parsing and validating universal gesture data in order to obtain gesture data object model.

Using HUGE API, application development is much easier and quicker and universal data formats databases can be easily shared among developers of HUGE systems.

In this section we discussed the implementation issues of HUGE system: HUGE universal data formats and HUGE API. In the following section we will further elaborate universal HUGE architecture by presenting two case studies, which represent our ongoing work that is based on the universal HUGE system architecture. First, we will present HUGE system where inducement data is lexical structure of spoken text. In second system, inducement data are parameters of speech signal. In the Introduction we also mentioned bio-signals as gestures inducement. Since we have not performed any analysis of inducements based on bio-signals, we will not further discuss them in this paper.

4. Case Studies

4.a. Text-induced gestures

In this section we will present our HUGE system implementation that uses as inducement lexical structure of uttered text. Figure 3 represents the HUGE system architecture adapted to the lexical structure of uttered text as the system inducement in system real-time part.

In statistical model generation phase, our HUGE system implementation is identical to HUGE system architecture. Using observation of training video clips, and simple Graphical User Interface for HUGE API, we manually created timed gesture sequences in universal gesture data format.

Timed lexical states in universal inducement data format were created semiautomatic. Using automatic lexical analysis of transcriptions corresponded to training video clips, we obtained the lexical states of the particular transcript word groups. Manually correlating transcripts with training video clips and using simple GUI interface for HUGE API, we produced lexical states time information (beginning and ending times).

Input of automatic lexical state annotation and classification HUGE process is plain English text. This process performs linguistic and contextual analysis of a text written in English language. The main goal of this process is to determine if word (or group of words) is new in the utterance context (theme or STATE00001), if extends some previously mentioned word or group of words (rheme or STATE00002) or does not belong to any of these lexical groups (STATE00003). More details about this module can be found in our previous work [1].

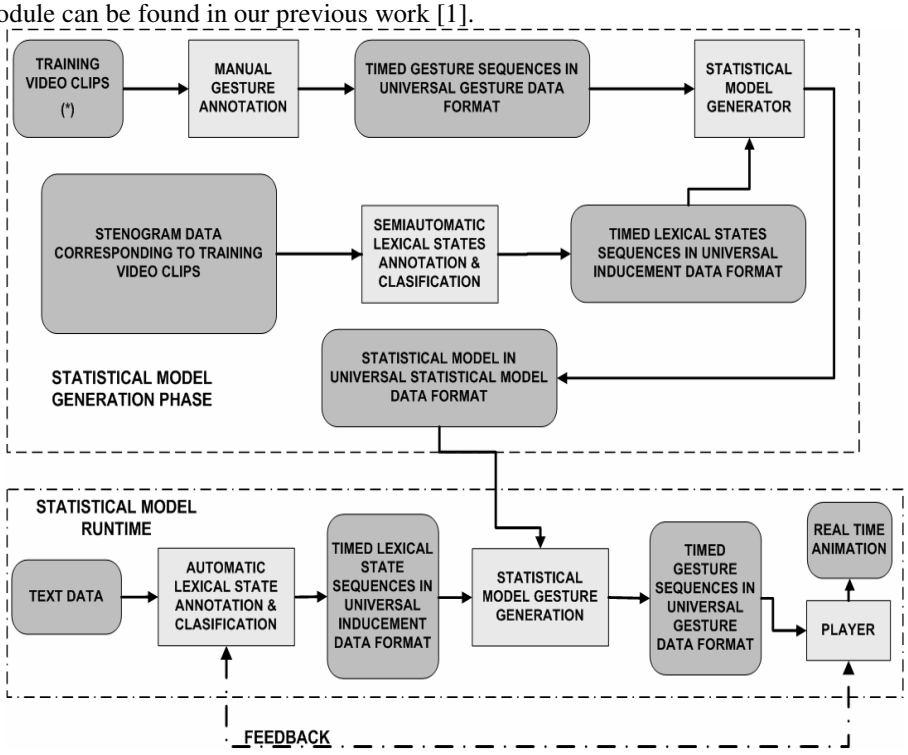

Fig. 3. Universal architecture of HUGE system adapted to text data as inducement

Then, using the feedback link to HUGE player process (implemented using Visage SDK API[4] and SAPI 5.0 Microsoft TTS engine[5]), it determines the actual timings and durations of the inducement states. This is possible, because all timing events in statistical model runtime are driven by the Text To Speech (TTS) HUGE player sub module. Timed lexical state sequences in universal inducement data format are input to statistical model gesture generation which, using the previously calculated statistical model data creates the timed gesture sequence in universal gesture data format. For example, if inducement state is STATE00002, gesture type for this state is determined by first generating a uniformly distributed random number between 0 and 100. Depending on its value and the statistical model for the particular state, gesture

[4] Visage Technologies AB http://www.visagetechnologies.com/
[5] Microsoft speech technologies http://www.microsoft.com/speech/

type is determined. If that gesture type has duration and amplitude parameters, again two uniformly distributed random numbers are generated and those numbers represent inputs into linear approximation of probability functions for those parameters (equation 1).

HUGE player process, using its TTS sub module, defines the timings of the real time animation. Combining timed gesture data sequence (transformation of the gesture universal data format into Microsoft's SAPI 5.0 TTS event format) and the TTS timing events, player process generates the real time gesture animation supporting all gesture types defined in the universal gesture data format. For example the SAPI5 event bookmark code for rapid left head movement is MARK= 8000000. Those bookmarks with appropriate values are inserted at the starting point of every gesture. Every facial gesture has a corresponding bookmark value. The head and eyebrows movement bookmark values not only define the type of facial gesture, but also contain the amplitude data and duration of the facial movement. For example, bookmark value 8051212 (Bmk_value) defines the rapid head movement to the left (symbol L) of amplitude (A) 1.2 MNS0 (Mouth Nose separation units) and duration (D) of 512 milliseconds. The functions for duration and amplitudes of facial gestures rapid L are:

$$D=(Bmk\_value - Bmk\_code)/100 \quad (\mathbf{2})$$

$$A=((Bmk\_value - Bmk\_code) - (D \times 100))/10 \quad (\mathbf{3})$$

In the interval [8000000, 9000000>for bookmark values for rapid L is possible to code maximal amplitude value of 9.9 MNS0 and maximal duration of 9.999 seconds. This limitation is not the problem because the statistical data showed that the maximal amplitude value for facial gesture L was 2.2 MNS0, and duration was never longer than 2 seconds.

TTS/MPEG-4 Playing Module plays in real-time, using the bookmark information, appropriate viseme and gestures model animation. It is based on Visage SDK API[6]. and on Microsoft Speech API SAPI 5.0 engine[7]. The synchronization between the animation subsystem (MPEG-4 Playing) and the speech subsystem (Microsoft's TTS engine) can be realized in two ways: with time-based scheduling and event-based scheduling. Which synchronization method will be used depends on the underling TTS engine implementation. In time-based scheduling, a speech is generated before nonverbal behaviors. Event-based scheduling means that speech and nonverbal behaviors are generated at the same time. In our system we are using the event-based scheduling method. We have implemented simple animation models for eyes blink, simple gaze following and head and eyebrows movement. Our system implementation is open, so every user is able to easily implement its own animation models. The animation model for head movement and eyebrows movement facial gestures is based on the trigonometry sine function. That means that our agent nods his head and raises eyebrows following the sine function trajectory. In gaze following animation model

[6] Visage Technologies AB http://www.visagetechnologies.com/
[7] Microsoft speech technologies http://www.microsoft.com/speech/

the eyes of our agent are moving in opposite directions of a head movement if a head movement amplitude is smaller than the defined threshold. This gives the impression of eye contact with agent.

4.b Speech-induced gestures

A particular type of signal that frequently accompanies facial gestures is speech signal. In speech-induced gestures an idea is to find a statistical correlation between speech signal and occurrence of gestures and to produce a speech driven facial animation. Such system is a special case of our universal architecture (figure 4). In the statistical model generation phase, audio data is used as raw inducement data to produce statistical model. Similarly, in the statistical model runtime phase, audio data is used as input data. To produce realistic animation, the player needs timed gesture sequences and also correct lip movements.

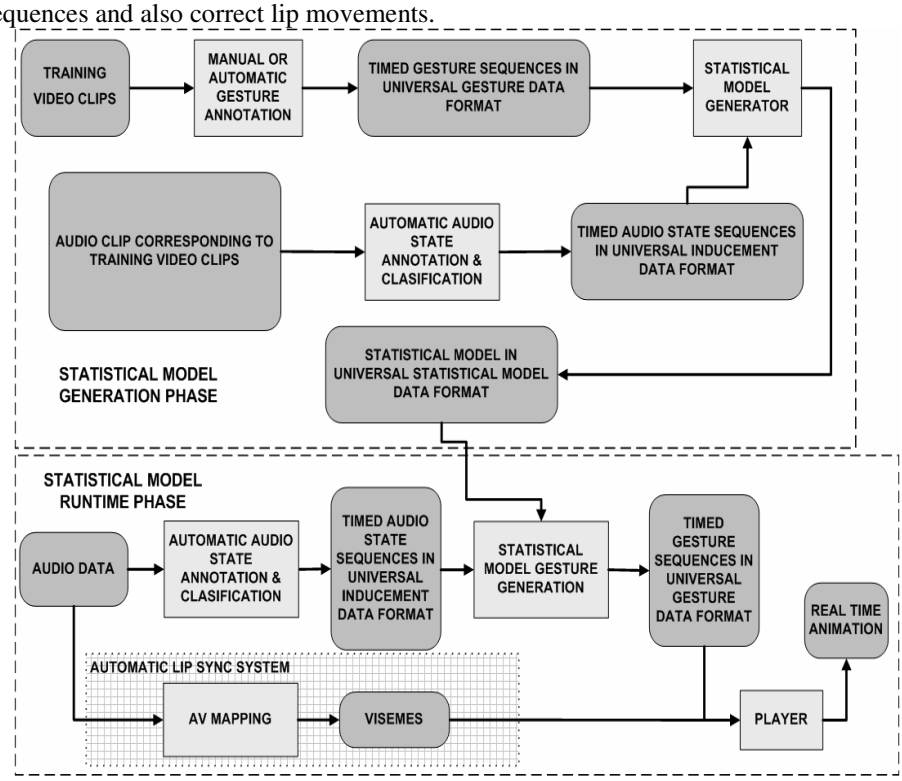

Fig. 4. Universal architecture of HUGE system adapted to audio data as inducement

Implementation of a system that uses speech-induced gestures is still ongoing work. In our previous work we have implemented an automatic Lip Sync system [13]. It takes speech signal as input and performs audio to visual mapping in order to produce the viseme, the visual representative of a phoneme. Next, we need to define

audio states using speech representation that takes into consideration speech prosody (e.g. pitch together with fundamental frequency, pauses). Once we have suitable audio states we will create statistical model and use it for gesture generation.

5. Conclusions and Future Work

In this article we presented our universal architecture for statistically based HUman GEsturing (HUGE) system for producing and using statistical models for facial gestures based on any kind of inducement. As inducement we consider any kind of signal that occurs in parallel to the production of gestures in human behaviour and that may have a statistical correlation with the occurrence of gestures, e.g. text that is spoken, audio signal of speech, bio signals etc. The correlation between the inducement signal and the gestures is used to first build the statistical model of gestures based on a training corpus consisting of sequences of gestures and corresponding inducement data sequences. In the runtime phase, the raw, previously unknown inducement data is used to trigger (induce) the real time gestures of the agent based on the previously constructed statistical model. We presented the general architecture and implementation issues of our system, and further clarify it through two case studies which represent HUGE systems adapted to lexical structure of spoken text and audio speech signal as inducement data.

There are several important benefits of using HUGE system architecture. First, we believe that this universal architecture is useful for experimenting with various kinds of potential inducement signals and their features and exploring the correlation of such signals or features with the gesturing behaviour. All data formats and structures are defined by universal data formats for inducements, gestures and statistical models. We postulated that at the given time interval only one inducement state is possible, so inducement parameters have to be grouped in inducement states in the manner to satisfy this time requirement. Also, overlapping of gesture time intervals is allowed only for the following gesture groups: head movement group (all nod and swing movements, including reset head movement) can be overlapped with eyebrows raise and eyes blink. This requirement is based on our observation of training video clips: at the same time interval humans might blink, move head in various directions and raise their eyebrows. HUGE API is defined in order to help developers to easily manipulate those data and to connect the HUGE system processes. Combining data and HUGE API, large database of statistical models could be easily produced and shared among different research teams.

We implemented HUGE API using .NET c# programming language. In order to make HUGE universal architecture easily implemented on other platforms, HUGE API should also be implemented using Java programming language. Furthermore, gesture statistical model should be produced using as the inducement data speech audio signal characteristics.

References

1. Smid, K., Radman, V., Pandzic, I. 2005. Automatic Content Production for an Autonomous Speaker Agent. Conversational Informatics for Supporting Social Intelligence and Interaction: Situational and Environmental Information Enforcing Involvement in Conversation / Nakano, Yukiko I. ; Nishida, Toyoaki (ur.). - Hatfield : AISB, The Society for the Study of Artificial Intelligence and the Simulation of Behaviour , 2005. 103-113
2. Zoric, G., Smid, K., Pandzic, I. 2005. Automatic facial gesturing for conversational agents and avatars. Proceedings of the 2005 International Conference on Active Media Technology (AMT 2005) / Tarumi, Hiroyuki ; Li, Yuefeng ; Yoshida, Tetsuya (ur.). - Piscataway, NJ, USA : IEEE , 2005. 505 - 510.
3. Irene Albrecht, Jorg Haber, and HansPeter Seidel. Automatic Generation of Non-Verbal Facial Expressions from Speech. In Proc. Computer Graphics International 2002 (CGI 2002), pages 283--293, July 2002.
4. I. Poggi and C. Pelachaud. Signals and meanings of gaze in Animated Faces. In P. McKevitt, S. O' Nuallàin, Conn Mulvihill, eds.: Language,Vision, and Music,John Benjamins, Amsterdam (2002), 133-144.
5. Lee, S. P., Badler, J. B., and Badler, N. I. 2002. Eyes Alive. In Proceedings of the 29th annual conference on Computer graphics and interactive techniques 2002, San Antonio, Texas, USA, ACM Press New York, NY,USA, 637 – 644
6. J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douvillle, S. Prevost and M. Stone. Animated Conversation: Rule-based Generation of Facial Expressions, Jesture & Spoken Intonation for Multiple Conversational Agents. In Proceedings of SIGGAPH '94, 1994.
7. Cassell, J., Vilhjálmsson, H., and Bickmore, T., 2001. BEAT: the Behavior Expression Animation Toolkit. In Proceedings of SIGGRAPH 2001, ACM Press / ACM SIGGRAPH, New York, E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 477-486.
8. Graf, H. P., Cosatto, E., Strom, V., and Huang, F. J., 2002. Visual Prosody: Facial Movements Accompanying Speech. In Proceedings of AFGR 2002, 381-386.
9. Cao, Y., Tien, W. C., Faloutsos, P., and Pighin, F., Expressive speech-driven facial animation, ACM Trans. Graph. 24, 4 (Oct. 2005), 1283-1302.
10. R. Gutierrez-Osuna, P. Kakumanu, A. Esposito, O.N. Garcia, A. Bojorquez, J. Castillo and I. Rudomin, Speech-driven Facial Animation with Realistic Dynamics, IEEE Trans. on Mutlimedia, Feb. 2005, 7, 1, 33- 42, ISSN: 1520-9210.
11. Granström B and House D., Audiovisual representation of prosody in expressive speech communication. 2005. Speech Communication 46: 473-484.
12. M. Brand, "Voice Puppetry", Proceedings of SIGGRAPH'99, 1999.
13. G. Zorić, I. S. Pandžić, A Real-time Lip Sync System Using a Genetic Algorithm for Automatic Neural Network Configuration, in Proceedings of the IEEE International Conference on Multimedia & Expo ICME, Amsterdam, The Netherlands, July 2005.