Personalized expressive embodied conversational agent EVA

IZIDOR MLAKAR, MATEJ ROJC
1Roboti c.s. d.o.o, 2Faculty of Electrical Engineering and Computer Science, University of Maribor
1Tržaška cesta 23, 2Smetanova ulica 17
SLOVENIA
1izidor.mlakar@revolutionary-robotics.com, 2matej.rojc@uni-mb.si

Abstract: - In this paper a new modular framework (EVA framework) and expressive embodied conversational agent EVA are presented. From talking heads to fully animatable bodies, and by techniques such as behavioral modeling and emotion modeling, researches are trying to present interaction interfaces providing as natural behavior as possible. The ECA EVA presented in this article is a mesh based multi-part model supporting both bone and morph target based animation. EVA can speak and perform body gestures such as facial gestures, gaze, hand gestures, etc. Each gesture is described by composition of movements of one or more base elements (bones and/or morphs), and can further be fine tuned by using time (speed) and space attributes (stress). Each gesture can therefore be unique, and either predefined (“offline” modeling) or described by xml based description (“online modeling”). The suggested multi-part model concept enables EVA to perform several gestures simultaneously and independently from each other (e.g. emotion blending, expressive speech). Additionally, the multi-part concept also enables for each of the model parts to be easily updated even whilst the EVA is being animated. The embodied conversational agent EVA, presented in this article, provides both personalization of its behavior (gesture level) and personalization of its outlook. EVA’s animation engine is Panda 3D based and fully supports Python programming language.

Key-Words: - Expressive ECA; mesh model; distributive; bone and morph based animation; Python

1 Introduction

The research goal of multimodal interaction research has moved from user-friendly interfaces towards platforms simulating human-like interaction, ranging from simple speech embedded applications to complex applications, such as intelligent environment interfaces [1], e-commerce applications [2] etc. Virtual characters, more often referred as to embodied conversational agents (ECAs) [3] play central roles in such interfaces. The ECAs are embedded into multimodal human-machine interfaces as animated talking heads [3][4][5], or fully functional conversational agents [1][6]. Most of the current research incorporates input processing stage and combined behavior planning and realization stage. For instance, MAXINE [1] incorporates various inputs, input processing and behavior modeling within a programmable interfaced structure. Similarly, most speech and facial emotion synthesis oriented frameworks (most popular concepts within ECA context), suggest incorporating one or more input processing techniques. In [8] the system for generating facial animation (including emotional speech synthesis) is based on statistical modeling and HUGE architecture [9]. More elaborated segmentation of interaction processes is presented in [11] and [12]. In this case the framework follows the principle of three stage based interaction architecture as described in [13].

The approaches stated above can, in general, be described as mesh based [14], and use morphed shapes and baked animations (predefined animation sequences) as templates for animating the desired behaviour. Animation by using the morphed target blending concept is very popular and is used in most speech centred agents. The mesh-based models and morphed target animations are relatively easy to handle and in contrast to appearance based models [15], mesh models require reasonable amounts of system resources. The skeleton-based (bone based) animation is seldom used when generating speech centric interfaces. Since most of the speech centric interfaces usually incorporate only the head, morphed targets are sufficient for natural representation of finite sets of facial gestures. The presented embodied conversational avatar EVA presents a modular environment for the realization of natural human-machine interaction (e.g. facial, body gesture, audio/visual speech synthesis, etc.). This EVA framework is based on the concept of distributed environments, as presented in DATA [16] and Panda 3D [17] as an animation engine. In order to achieve believable gesture and facial animation the mesh based animation procedures of the EVA framework support both skeletal and morph based animation. Each behavior can, therefore be described as a set of morphed target translations and bone movements, moving in parallel or sequential intervals.
2 EVA – 3D model

The 3D model of EVA can be described as a set of polygonal meshes with corresponding textures, each mesh containing underlying bone chain and morphed models. EVA’s 3D model was modeled by Maya 3D modeling software. A multi-part actor based model is described as a model built from different 3D sub-models (e.g. body parts) that share the same segments of the skeleton. Since each of the 3D sub-models has its own bone chain, a base bone chain is defined with one bone common to each of the sub-model bone chains. A multi-part actor is within animation engine environment generated by connecting the corresponding bones to the base skeleton. Figure 1 further shows basic concept of the multi-part actor and displays the overlay of the EVA’s 3D model.

EVA is (Figure 1) a multi-part actor built from 3D models of body, hair, tongue, teeth, eyes, dress, and accessories connected by shared skeleton. Each of the 3D models also has its own set of materials and textures applied. Similarly, as most of the mesh based models, the textures and materials are applied to each of the 3D sub-models by processes known as texturing and UV mapping. Bones, bone chains and morphed shapes present a set of control units. Behavior, such as visual speech synthesis, facial gestures, facial emotions, gaze etc., can then be formed and animated. Figure 1 also shows mesh example of the multi-part based 3D model (actor) and its underlying bone chain. The end result of 3D modeling process, however, is an, in relation to Panda 3D, Actor stored within .egg or .bam files. Since each 3D sub-model is stored within its own .bam file, each of the sub-models and corresponding animations can be modeled independently and by using any 3D modeling software that can store 3D models in appropriate formats. The main advantages of using multi-part concept, as presented in Figure 1, are:

- Efficient way of personalization of ECA’s outlook. Each 3D sub-model can be replaced by a different 3D sub-model (e.g. changing hair style by changing 3D sub-model of hair) without any setting/code correction, and can even be achieved ‘‘online’’.
- Implementation of expressiveness. Each body part is correlated to a unique segment of the skeleton and influences unique segment of the body. Therefore, different 3D models can display different animations simultaneously (e.g. moving head while speaking, smiling while speaking, gesturing with hands and performing different facial gestures during a dialog, etc.), and even different gestures can be displayed on the same segment of the body (e.g. emotion masking – sad face is masked by a smile).

3 Animation Engine

The animation engine animating EVA’s behavior is Panda 3D based and supports Python programming language and all Panda 3D native animation techniques, such as forward and inverse kinematics. The architecture of the EVA framework’s animation engine is presented in Figure 2. The engine separates behavior generation (e.g. behavior and emotion modeling), and animation realization (from behavior description to animation). Following the concept of distributive processing, presented in [16], the animation engine of EVA framework is built as either DATA module server, or DATA client. The build depends only on the usage context of the engine within the system. When the animation engine is used as an endpoint and the ECA is presented on the local display, DATA client implementation is adopted, whereas when the animation engine is a service that produces audio/video stream the DATA module server implementation is adopted. Regardless the implementation, the base architecture of the animation engine remains unchanged, as displayed in Figure 2.
The concept of animation engine (Figure 2) consists of four layers. The first layer, so called static layer, consists of those ECA’s definitions that are relatively static. Therefore, it encapsulates base description of the ECA’s 3D model (3D sub models describing mesh, texture, skeletal and morph information), and behavior description specific to ECA (personality). Basically, the first layer provides animation control points of the ECAs and therefore describes concepts necessary for synthesis of human-like body movements such as viseme articulation, FAP based description of facial animation, hand gesture movement chain and base hand gestures etc.

The second layer of the animation engine (generation layer) provides means and interfaces to animate control points provided by first layer, and to transform external animation parameters into smooth, continuous animated sequences. The network interface of the second layer contains DATA System’s related procedures and describes both DATA client and DATA module server usage modes of the animation engine. The event processing interface processes the external XML based animation parameters descriptions and transforms them into sequences of animation strings realized in the animation realization layer. The generation of animation and its realization from XML provided animation parameters will be further described in subsections 3.1 and 3.2 of this paper.

The third layer of the animation engine (behavior generation) can be presented as DATA system’s related DATA service cloud. The service cloud contains sources that can provide animation parameters, such as text-to-speech synthesizer (TTS), automatic speech recognizer (ASR), or video analyzer and sources that can modulate the behavior such as ALMA [2], EMOTE [18], etc. The output of this behavior generation block contains animation parameters encapsulated within XML structure and stored in the form of TCP/IP packets (events).

The final, fourth layer of the animation engine is animation realization layer. The layer transforms animation streams into sequences of animated movement. By using VLC and/or G-Streamer open source audio/video processing libraries the animated sequences can be further processed and transmuted into different video streams (e.g. HTTP or RTP), or stored into video files. The audio/video post-processing also plays essential role when implementing animation engine as a network-based distributive service. In the following subsection of the paper the core of animation engine, the animation generation and realization layers will be presented.

3.1 XML description of animated behavior

Input/behavior processing is separated from the core animation engine (Figure 2). Therefore, animation engine only takes care for processing of input data in the XML form (text), and audio streams. The animation parameters of each animation segment (behavior event) are encapsulated within XML based structures. EVA XML descriptions follow the general XML schemes used for describing ECA behavior, such as AMPL, BML, EML, etc. Additionally, EVA XML descriptions further extend these schemes in order to provide the ability of animating expressive behavior such as expressive speech (speech + facial gestures + hand gestures). Based on the functionality and location of body movement we define three general types of body movement, speech sequence, facial gesture and body gesture. The body movement tags are also defined and based on the assumption that the human body can be segmented into several independent regions, and that the unique groups (based on the gestures performed within each region) of control joints and joint chains can be designated to each of these regions. The example of XML definitions are shown in Figure 3-5. The control units follow MPEG-4 FAP standard. Their movement is described by rotation (H, P, R) and/or transition (X, Y, Z) 3D vector. The morphed shapes, however, have only one dimension of movement. The movement of such shapes is always defined as movement on the X-axis of the 3D space and on the floating interval [0, 1].

```
<Speech emotion="joy" stress="3.0" start=""/>
</fgesture>
</speech>
</Speech>
```

Figure 3: Example of speech sequence description

Speech (Figure 3) is described as a correlated sequence of visemes/phonemes (speech units). Each speech unit is defined by its name, duration and the level of articulation. Each speech unit in speech sequence, encapsulated within EVA XML <speech> tag, is therefore described by name, duration of displaying (durationUp + persistence) and duration of reverting to neutral state (durationDown). If the overall speech sequence is colored with an emotion, the emotion attribute describes the emotion.

Facial gestures (Figure 4) (e.g. emotions, gaze) present additional information channel in human interaction. The facial gesture related animated sequences are encapsulated within <fgesture> tag. Each facial gesture is defined by its type (e.g. speech related), name (e.g. smile), duration (durationUp, durationDown and persistence) and stress level (to what extent will the
gesture be defined). These general attributes are used when a gesture is predefined (already described within the set of predefined gestures). The EVA framework also provides the ability to define and describe new gestures (not defined in the predefined set). Descriptions of such gestures follow the general EVA framework idea: let be each body movement simply a set of parallel/sequential movements of its elementary parts. Therefore, each facial gesture can be defined as a composition of movement of its elementary control units (joints or morphed shapes). For an instance, a smile can be defined as a final, predefined gesture (e.g. simple smile), or as an ‘‘online’’ gesture (e.g. smile with slightly open mouth) composited out of simple smile and jaw movement. Since most facial gestures are closely related to FAP, the elementary control units of EVA framework are defined based on MPEG-4 FAP. ‘‘Online’’ gestures are within EVA XML described as a set of <FAP> tags, encapsulated within <sequence> or <parallel> tags. The <sequence> and <parallel> tags define the flow of animation, and whether the sets of FAPs will be animated sequentially or simultaneously. The idea of nesting the control units into sequences and parallels enables the animation of any type of gesture with any degree of complexity (assuming complexity relates to the control unit structure of the gesture). The gestures in EVA framework can therefore vary in duration (time domain), degree of presentation (space domain) and in composition.

In order to animate any XML event (behavior description) efficiently, the XML event data are processed by separate XML event processor and mapped into a set of sequences, or parallel intervals of movement (translation/rotation), consisting of one or more control joints. The XML event processor firstly checks the behavioral type (speech, body or facial gesture), and then the type of animation the gesture uses (either parallel or sequence). Proper movement intervals are formed based on the description of each element of the XML event. Each formed interval is then added to the animation as a sequence, or in parallel. Each of the movement type is processed individually, and by using animation blending it is inserted in overall ECA animation according to its behavior description. The internal synchronization process presented in section 3.2 ensures that each of gesture types is always ‘‘in-sync’’ with its predecessors, and that the animation flow is always continuous and free of erratic/’’jerky’’ movements.

### 3.2 Animating the behavior

Each bone/morphed target in 3D model presents a control joint (control point) that can move or influence certain amount of vertices on the polygonal mesh. A continuous sequence (or parallel) of different influences can be regarded as a gesture or movement. When generating animated sequences, different animation segments (animation states) must transit between each other, without transition to a neutral state. On transition between animations states the corresponding control point moves to the desired position. When animating behavior, the animation engine simply acts as a finite state machine (FSM). Figure 4 presents the concept of such internal synchronization and generation of animation strings realized in the Animation realization layer of the EVA’s animation engine. As an example, let’s say that we have 2 animated sequences that we wish to visualize. The first animated sequence presents happy face and the second disgust.

Body gestures (Figure 5) represent movements that relate to the head, hand and other movements, not contained within speech sequences or facial gestures. The definition of body gesture is similar as the definition of facial gestures. The animation with body gestures is, however, assumed to be used for animating movements consisted of bone based control units.

---

**Figure 4:** Example of facial gesture description

```
<gesture type="speech" name="happy" durationUp="1.0" durationDown="1.0" persistend="true"></gesture>
```

**Figure 5:** Example of body gesture description

```
<sequence><parallel>
  <UNIT name="LeftJoint" type="RKP" state="0.02;0.05;22.00" duration="10.0" persistend="true">< DurantDown="10.0"></ DurantDown>
  <UNIT name="RightJoint" type="LKP" state="20.0" durationUp="10.0" persistend="true">< DurantDown="20.0"></ DurantDown>
</parallel></sequence>
```

**Figure 6:** Animation generation and synchronization
As follows from Figure 6, animation sequence I is described by 3 animation states and two transitions. The animation state A1 is used for neutral state (corresponds to image A1), the animation state A2 is used for transitional state, and the animation state A3 (image A3) completes the animation sequence I: a happy face with open smile. The animation sequence II is described by only animation state B1 (image B1), where animation state B1 also completes the animation sequence II. As already mentioned before, the animation of control units occur on FSM transitions between different animation states. Within the animation transition phase, only control units related to current animation state Ai, will be lowered (moved to their neutral position), control units related to animation state Ai+1 will be raised, and control units related to both Ai and Ai+1 states, will transit from the position defined by state Ai to new position, as defined by state Ai+1. Mathematically, the transitions within given time interval t can be defined as follows:

- Raised control units: StiR(t) = Ai+1\((Ai \cap Ai +1)\\/control units exclusively described by Ai+1
- Neutralized control units  StiL(t) = Ai\((Ai \cap Ai +1)\\/control units exclusively described by Ai
- Transiting control units    StiT(t) = (Ai \cap Ai +1)\\/control units by both Ai and Ai+1

Therefore, the animation sequence can be defined as parallel movement of control units within sets StiR(t), StiL(t) and StiT(t).

For an instance, the animation sequence I defines movement of following control units: Mouth-Corners, Lips, Jaw and Teeth. The animation sequence II defines movement of LipsSneer and Jaw control unit. The transition between states A3 and B1 (Figure 6) will in this case neutralize the influence of Mouth-Corners, LipsPart, and Teeth control units, raise LipsSneer control unit and transit the Jaw control unit.

The result of the synchronization and animation generation process displayed in Figure 6 is described as animation string, as a sequence of control unit movements, or as a sequence of parallel movements of control units, and is realized as desktop or video stream. Depending on the usage mode (service or local) the ECA EVA was initiated.

4 Conclusion and future research

ECA EVA and underlying EVA framework present a novel engine for generating multimodal interfaces, used for natural human-machine interaction. EVA framework enables synthesis of both verbal and non-verbal behavior. It is based on distributive concept and physically separates the behavior generation and behavior realization (animation generation and realization) phases. Therefore, it enables the usage of any type of source that can provide animation parameters in the form of behavioral events; any source contained within DATA service cloud. The DATA framework architecture also enables the EVA to run in either, service or local mode, and support different (even resource limited) user devices.

Different sources can either provide combined behavior (e.g. behavioral modeling), or each one its own behavioral events that are processed in parallel into common animated behavior. The behavioral events provide the descriptions of desired animated behavior, and the synchronization process ensures that all sequential segments of an animation (animation states) are always continuous. By using animation blending technique, different animation segments (different body part animations) are combined into smooth and continuous animated movement. By using multi-part based 3D models of an ECA, the framework enables online personalization of the ECA. Each body part can be modeled and animated independently. ECA EVA presented in the article is also capable of animating expressive behavior. Figure 5 shows an example of expressive behavior, the expressive speech.

The expressivity of an ECA plays a central role in defining its personality, its emotional state, and can further explain the context of the spoken dialogue (e.g. what parts of the dialog are important – emphasis, visualization of the spoken word etc.). The expressivity basically defines “how” information is presented through
physically based behavior (movement), and plays a central role in the perception of verbal and nonverbal dialogue. Our next steps will be directed towards context-oriented behavioral modeling. Since EVA framework can render and animate any behavior provided by behavioral events, our main focus will be set on those sources that can provide such events, e.g. from video processing techniques to behavior modeling systems that are capable to learn and adapt.

Acknowledgements

Operation part financed by the European Union, European Social Fund.

References:
[7] Alexis Heloir, Marc Schröder, Patrick Gebhard Realizing Multimodal Behavior Closing the gap between behavior planning and embodied agent presentation
[10] Y Fu, R Li, TS Huang, M Danielsen, Real-Time Multimodal Human–Avatar Interaction, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 18, NO. 4, APRIL 2008

1 Autodesk Maya 3D – http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&ikid=13577897
2 VLC - http://wiki.videolan.org/LibVLC
3 G-streamer - http://www.gstreamer.net/