RealActor: Character Animation and Multimodal Behavior Realization System

Aleksandra Cerekovic[1], Tomislav Pejsa, and Igor S. Pandzic

University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia, {aleksandra.cerekovic, tomislav.pejsa, igor.pandzic}@fer.hr

1 Overview

In this paper we present RealActor, a character behavior realization system for embodied conversational agents based on the Behavior Markup Language (BML). Developed several years ago as part of the SAIBA framework, BML is an XML dialect for describing physical realizations of multimodal human behaviors. It allows modeling of complex communicative utterances which include both verbal and non-verbal behavior. BML elements represent various primitive actions (e.g. speech, facial and body gestures) and multimodal behavior can be modeled by specifying temporal relationships between these elements. Our BML-based character animation system has the following features:

- specification of character behaviors using BML scripts
- start, stop, schedule or merge behaviors via high-level, BML-compliant API
- database of annotated animations shared between multiple characters
- motion playback system based on MPEG-4 FBA standard
- visual text-to-speech synthesis based on industry-standard Microsoft SAPI
- lip synchronization
- integration with any graphics engine via a minimal scene wrapper
- flexible art pipeline based on universal file formats (FBX, COLLADA, VRML) that includes tools for automated production of face models and morph targets for facial animation

The BML realizer (Fig. 1) is the core module of our system and consists of three central components: BML Parser, Behavior Planner and Behavior Scheduler. BML Parser parses BML scripts, generates appropriate behavior blocks and adds them to a list. Behavior Planner prepares each block for execution by adding timing information needed for multimodal behavior synchronization. For primitive animations this timing data is retrieved from the animation database, where each animation is annotated with time constraint and type information. Speech is handled using lip-sync or text-to-speech synthesis - if lip-sync is used, speech must be prerecorded and manually annotated with timing data, while timings for synthesized speech can be inferred at run-time. Finally, BML Scheduler is responsible for execution of prepared behaviors and uses timing information to decide which behaviors will execute and when.

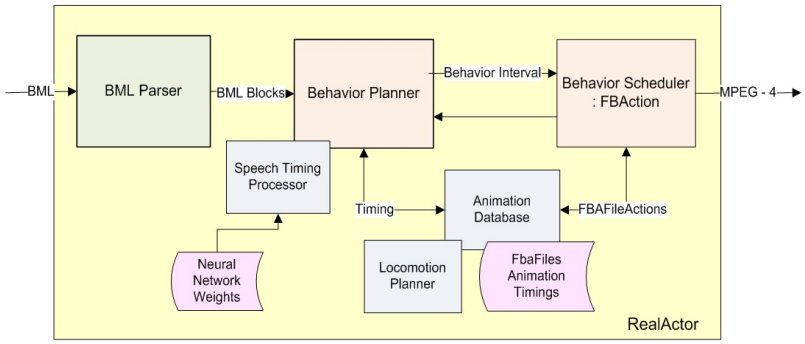

Fig. 1. Architecture of the BML realizer

Most TTS systems (including Microsoft SAPI used in RealActor) do not provide a priori phoneme and word timing information necessary for synchronization of synthesized text with non-verbal behavior. To address this issue, we utilize backpropagation neural networks (BNNs) to estimate word durations and align speech with animation in real-time. Our system is able to correctly align 92,26% of words for a short time interval (500 ms) and 73,26% of words for a long interval (1500 ms) with alignment error not exceeding the 80 ms threshold (which is the shortest discrepancy between audio and video perceptible to human beings). Furthermore, the system achieved 79,03% and 56,59% alignment rates with no measurable alignment error.

2 Results and Future Work

We have developed RealActor, an open-source character animation and multimodal behavior system based on Behavior Markup Language (BML). To our knowledge, it is one of only three BML-compliant animation systems in existence. We have developed a universal, language-independent solution to the issue of multimodal behavior synchronization that utilizes neural networks to estimate word durations. Furthermore, our system is designed in such a manner that it can be integrated with existing graphics engines and application frameworks with minimal effort, which we demonstrated by integrating RealActor with open-source engines such as OGRE, Horde3D and Irrlicht. While testing RealActor, we observed that manual authoring of BML scripts can be time-consuming and counter-intuitive and plan to address this by providing a graphical BML authoring tool or by deveping a higher-level component that would generate behaviors automatically. On lower level, we will introduce a new animation system that will utilize parametric motion graphs to synthesize long sequences of visually pleasing and interactively controllable motion.