Authoring an Interactive, Functional, Purposeful, Virtual Populace

Submission ID:002

No Institute Given

Abstract. This paper presents a robust and flexible simulation architecture with an activity authoring system for building complex inhabited interactive virtual environments for military training applications. The presence of an interactive virtual populace adds to the immersion and provides a potentially richer training experience for subjects. Events authored using Parameterized Behavior Trees (PBTs), involving multiple actors in specific roles, enable mission directors to author complex missions requiring a deeper engagement of the populace by defining events which are triggered based on spatial and temporal conditions, or user input. We demonstrate our framework by authoring and simulating a variety of missions including cordon and search, object retrieval, crowd dispersal, and evacuation in a Middle Eastern marketplace environment.

Keywords: serious games, interactive virtual world, human-robot interaction, functional populace

1 Introduction

The provision of an interactive, functional, purposeful virtual populace is of paramount importance to completely immerse and engage human subjects in military training simulations and other serious game applications. Autonomous virtual characters must act and interact with each other in meaningful ways, and elicit appropriate behavior in response to stimuli from the player controlled soldier or other members of a soldier-robot team. Furthermore, different missions may require the virtual populace to follow narrative arcs, or participate in certain events that are triggered in a dynamic fashion.

This paper presents our ongoing development of a virtual Middle Eastern marketplace environment to serve as a testbed for military training simulations, and human soldier-robot team experiments. We extend the ADAPT platform [19] for simulating and animating a virtual populace, with heterogeneity injected at both the visual and behavior layers. Variety in visual appearance is provided by procedurally generating virtual human models from a small library of art assets and different behaviors are authored using Parameterized Behavior Trees (PBTs) [18]. The hierarchical and parametric nature of PBTs facilitates ease of authoring, modularity, and reuse where existing tree definitions can be attached as a sub-tree to author more complex behaviors, involving multiple interacting actors. Using our platform, we author an interactive simulation where villagers roam the marketplace, engage in conversations with each other, and haggle with vendors to purchase different goods. The villagers respond to the presence of the soldiers and robot by either cautiously keeping a safe distance, waving to the soldiers as they pass by, or even following the robot out of curiosity.

The virtual populace injects life into the simulation, providing a greater degree of immersion, and a richer training experience for the users. Missions requiring a deeper engagement of the populace, and more complex interactions with the soldier-robot team are designed using an event-centric authoring paradigm. Events are PBT definitions involving multiple actors, which can be triggered based on spatial and temporal conditions, or user input. For example, a group of suspicious villagers engaged in a conversation will disperse or even run away when the soldier or robot approaches them. A mission director can easily author a variety of mission directives by defining a library of events and conditions that trigger them. We demonstrate our framework by authoring and simulating a variety of missions in a Middle Eastern marketplace environment.

2 Related Work

Authoring interactive virtual worlds has been studied extensively from many different perspectives [1]. These methods represent different tradeoffs between the ease of authoring, autonomy of generation, and level of interaction between the user and the virtual avatars.

Scripted approaches [9, 11] describe avatar behaviors as predefined action sequences, requiring authors to define scripts for each fore-seeable interaction. Improv [15] and LIVE [12] use rule-based systems where behavior is governed on the satisfaction of a certain set of preconditions. These systems are reactive in nature and are not designed to generate complicated agent interactions which have meaning over the entire course of a lengthy simulation. Crowd approaches [2, 3] provide interfaces to specify goals or map parameters to personality traits and examine the emergent behaviors in crowds. These techniques don't provide fine grained control required to author complex multi-actor interactions. Cognitive approaches [4, 20] use complex models such as decision and neural networks to author monolithic agent architectures equipped with knowledge bases and mechanisms for dynamic action selection. These approaches are not easy to author, and often require domain specialists or machine learning tools to automate the authoring process.

Domain-independent planning [5, 7, 14] is a promising direction for automating behavior generation in well-defined problem domains [14]. However, complex multi-actor interactions require the overhead of a centralized scheme or the modeling of agent communication. Furthermore, player input may invalidate current plans, requiring an efficient replanning scheme. Hence, current applications that use planning for behavior generation are limited to simple problem domains with a small number of interacting agents. Commercial platforms [8, 10] provide end-to-end solutions for character animation, pathfinding and steering, and behavior authoring. SmartBody [16] and ADAPT [19] are open-source solutions that enable end users to author and generate interactive virtual environments populated with functional, purposeful virtual humans. The use of simulation for military training has gained importance in recent years as authoring and simulation platforms have grown more mature and are capable of supporting military requirements. However, current platforms [6, 8, 13] focus on recreating photo-realistic environments and have limited capabilities for interactions between autonomous virtual characters and the soldier trainees.

Comparison to Related Work. We extend the ADAPT simulation platform [19] to author a functional, purposeful virtual marketplace where human-robot soldier teams can perform a variety of user-authored missions that involve complex interactions with the inhabitants. We use parameterized behavior trees [18] and an event-centric authoring paradigm [17] to provide a flexible, intuitive interface for authoring complex multi-actor interactions that involve human controlled actors and follow specific mission directives.

3 Framework

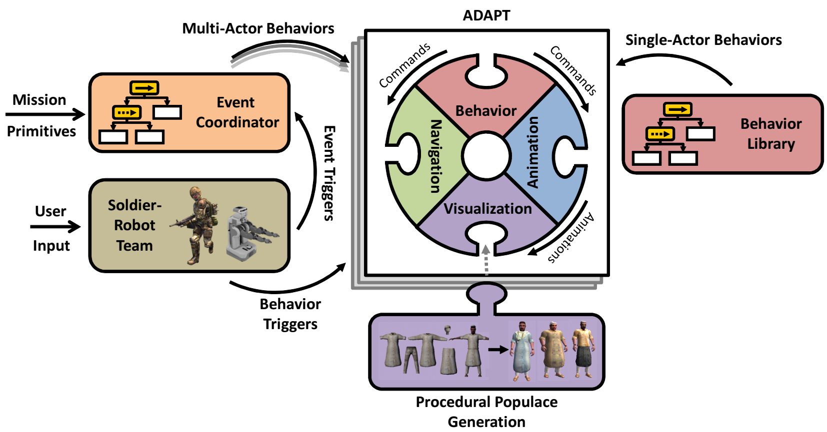

Fig. 1: Framework Overview.

We build on the ADAPT simulation platform by procedurally generating a diverse virtual populace and author complex avatar behaviors using Parameterized Behavior Trees (PBTs). An event-centric authoring paradigm facilitates complex multi-actor interactions without the need for heavy scripting. A user-controlled soldier-robot team is introduced into the simulation which elicits appropriate behavioral responses in the populace and triggers events which are authored to define specific training missions. Figure 1 presents an overview of our authoring framework, which is described below.

3.1 Procedural Populace Generation

Diversity in the visual appearance of a virtual populace is essential for a realistic simulation. The burden of manually creating a set of different characters is prohibitive and does not scale well for large crowds. To offset this limitation, we provide the ability to procedurally generate customizable virtual humans using only a small library of art assets. The input to the system is a set of assets representing textured meshes for different body parts of a virtual character. During runtime, different combinations of these assets are used to procedurally build a complete character skeleton, while satisfying any user constraints on the mesh selection.

Asset Specification. Each body part asset is associated with an XML file providing information that is used by the procedural character generator to combine different assets to create the complete human character. The XML files contain information about the geometry, UV texture atlas, atlas offsets, the hierarchical list of bones to which the geometry is attached, and other semantic tags used for specifying constraints on selection of assets. Figure 2 illustrates the XML specification for an asset and Figure 3 illustrates some assets that were used in our framework.

```
<BodyPartMetaData>
    <geomName> name of geometry</geomName>
    <uvName> texture atlas name </uvName>
    <offsetV> texture atlas offset range </offsetV>
    <tags>
        <string> skeleton root </string>
        <string> semantic tag 1 </string>
        <string> semantic tag 2 </string>
        ....
    </tags>
    <clothVert> number of vertices </clothVert>
</BodyPartMetaData>
```

Fig. 2: XML Specification for body part assets.

A crowd generation module generates groups of characters, while satisfying constraints on the spatial distribution of different character types. This provides a simple, yet intuitive interface for specifying and automatically generating a heterogeneous virtual populace, with sufficient control on the appearance of the characters both at the single character and crowd scale. The steps for procedural crowd generation are described below.

Fig. 3: Body part assets used for procedural character generation, and procedurally generated Middle Eastern characters.

1. **Asset Import.** Import a library of art assets for different body parts of a virtual character. Generate an XML specification file for each asset describing its geometry, which body joint(s) it is associated with, and any other semantic tags used for constraint specification.
2. **Procedural Character Generation.** For each body part of the character (e.g., head, torso, legs, arms, etc.), randomly select an asset from the library while satisfying any user-specified constraints. Build the character by combining the assets and binding to the skeleton. Dynamic texturing facilitates swapping materials on each asset for greater variety. Figure 3 illustrates example models that were generated using our framework.
3. **Procedural Populace Generation.** Repeatedly create characters to generate a crowd of the desired size while satisfying given constraints on their location and distribution in the virtual environment. A GUI can be used to edit crowd or individual character parameters such as body part scaling, asset category, number of characters at given location, etc.

3.2 Virtual Human Simulation

The ADAPT platform [19] provides the tools for authoring functional, purposeful human characters in a virtual environment by combining character animation, steering, pathfinding, and behavior authoring in a single framework. We extend ADAPT with the procedural generation of characters and provide a framework for authoring and simulating a variety of missions for a soldier-robot team to perform in a Middle Eastern marketplace scenario, involving complex interactions between the soldiers, robot, and the virtual populace.

Navigation and Steering. ADAPT provides a navigation mesh approach for pathfinding. Given an arbitrarily complex virtual scene, we can automatically precompute a navigation mesh which stores a triangulated representation of free space to support efficient navigation queries. In addition, goal-directed collision avoidance is supported using a predictive steering algorithm.

Full-body Character Animation. The ADAPT animation system supports full-body character animation and supports multiple control tasks such as locomotion, gaze tracking, and reaching using independent control modules. Additional controllers can be easily added without requiring any change to the rest of the system.

3.3 Behavior Authoring

The animation and navigation system provides a set of capabilities that each character can perform, such as `ReachFor()`, `Goto()`, and `GazeAt()`. In order to invoke sequences and combinations of these capabilities in a context-specific fashion (e.g., maintain a safe distance when the robot is near), ADAPT provides a graphical programming paradigm for behavior authoring using Parameterized Behavior Trees (PBTs). PBTs support authoring multi-actor behaviors in a single tree structure, which greatly reduces the authoring burden for more complex interactions between multiple actors. For more information on the ADAPT system, please refer to [19].

3.4 Event Coordination

In order to author complex missions for the soldier-robot team to perform in the marketplace, we need a mechanism to trigger event trees in a dynamic fashion based on the presence and interaction of the soldier-robot team with the populace. To achieve this, we define a centralized *event coordinator* which automatically instantiates behaviors and events in the populace to conform to the specific mission directives which are user-defined. A mission director defines a mapping of events which must occur when certain conditions in the simulation are satisfied. These conditions include: (1) user input , (2) the spatial location of the soldier and robot, (3) when the soldier or robot performs a particular interaction (e.g., investigate suspicious vendor stall), (4) temporal conditions, or (5) the success or completion of ongoing events.

Figure 4 illustrates a generic mission definition. The event coordinator monitors the simulation and instantiates events when a particular condition is satisfied by

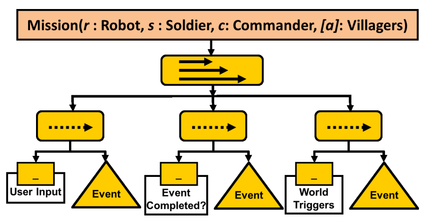

Fig. 4: Generic mission definition.

selecting one or more actors to participate in the event which satisfy the designated role. It is important to note that events are are hierarchical control structures rather than scripted action sequences, and may succeed or fail based on the actions of the user controlled soldier-robot team. This ensures that every run of the same mission produces a different simulation where the outcome is contingent upon user input, facilitating a dynamic simulation environment which exercises the soldier-robot interactions in a variety of different situations, without the need for heavy authoring.

4 Middle Eastern Marketplace

4.1 Environment

We designed and authored a Middle Eastern marketplace environment, populated with a native virtual populace. We used information from online sources and military personnel as the basis of the environment design. The market is present in the heart of a village which consists of small buildings, an outdoor seating area or restaurant, a cemetery and a roadway that runs through it. The marketplace consists of a series of stalls that are connected to the buildings and occupy a portion of the main street. A navigation mesh is precomputed to facilitate efficient pathfinding and steering for the virtual characters. In addition, objects in the virtual environment are semantically annotated to enable the authoring of behaviors which require actors to interact with the environment.

4.2 Soldier-Robot Team

The soldier-robot team comprises a team captain, a robot, and one or more other soldiers, who are tasked with carrying out a specific mission in the given environment.

Team Captain. The team captain is controlled by a trainee subject using a traditional keyboard and mouse interface. More complex control interfaces such as gestures or voice commands can be readily integrated into this framework. The subject controls the movement of the team captain using a traditional WASD control scheme, or by clicking on a point in the environment. Commands are issued by first selecting the robot or teammate, and then triggering which command to issue by pressing a predefined button. For certain commands such as `Goto`, an intermediate step is required where the user clicks on a position or object in the environment which is required as a parameter to the behavior. The user can additionally interact with the virtual populace by clicking on them and selecting a mode of interaction. The commands issued by the user result in appropriate animations on the subject's avatar: the team captain performs a gesture to indicate a particular command or points at a specific location or object to go to. The user can choose from a first person view, a third person view, or a bird's eye view depending on the mission specification and type of training that is to be carried out.

Robot. The robot is a semi-autonomous agent. It monitors commands issued by the team captain which then trigger appropriate behaviors. A directional command interface where the user has fine-grained control of the robot is very burdensome, preventing the user from carrying out additional tasks simultaneously. Instead, we provide a forward-looking control interface where the user can issue high-level commands such as `Follow`, `Goto(object)`, and `Investigate` and the robot is equipped with autonomous capabilities to independently carry out these behaviors. This system allows us to lay down a strong foundation for choosing the right set of robot behaviors, on top of which one can define an executable tactical vocabulary. Table 1 defines a set of behaviors that we defined for demonstrating the missions described in Section 5. Additional behaviors can be easily added using the authoring interface described earlier.

Other soldiers. The other soldiers can be controlled (1) directly by other users using a similar control interface, (2) indirectly by defining an extended set of behaviors that can be triggered by the team captain's commands, or (3) by conditions in the environment.

Behavior	**Description**
`ComeHere`	The team captain(user) instructs the soldier or robot to arrive at his location.
`Follow`	The soldier or robot is instructed to follow the team captain.
`Stay`	The soldier or robot is instructed to stay at their current position.
`Goto (pos/obj)`	The soldier or robot is instructed to goto a particular spatial location or an object in the environment.
`TakePictures (pos/obj)`	The robot is instructed to goto a particular object or location and take pictures from different vantage points.
`Pick (object)`	The robot is instructed to goto a particular object and pick it up.
`Drop (pos)`	The robot is instructed to drop the picked object at a location.
`Investigate (object)`	The soldier or robot is instructed to goto a particular object and investigate it by picking it up, taking pictures, and retrieving it.
`Patrol (pos1,pos2)`	The soldier or robot is instructed to patrol between two specified positions.

Table 1: Individual behaviors defined for the robot and soldier teammate, which can be triggered by the user.

4.3 Virtual Populace

A semantically labeled Middle Eastern marketplace environment provides a contextually relevant scenario to carry out different training missions pertaining to the region. The presence of a functional, purposeful populace that behaves in an appropriate manner and responds to the soldier-robot team is of paramount importance to a realistic interactive training experience. Figure 6(a) shows a virtual marketplace populated with villagers and vendors. Villagers wander about in the marketplace and stop by the vendor stalls to look at the goods that are being sold. If something catches their fancy (or is in their shopping list), they initiate a `Haggle` behavior to interact with the vendor in order to purchase a particular product. If the haggle succeeds, they pick up the product and resume their previous behavior. The vendor mans the stalls and waves at oncoming villagers, urging them to purchase his goods. `Conversation` behaviors between nearby villagers are triggered to bring the selected villagers at a meeting point, face each other, and play appropriate estures and head motions. Filtering the gestures that are performed results in more specific interactions that showcase agreement, disagreement, or even anger between the participating actors.

Additional specializations can be easily introduced by defining new behavior trees that are instantiated when their trigger condition is satisfied. The villagers are made to react to the presence of soldiers and robot by introducing a set of appropriate behaviors that are triggered when the villager is near them. For example, we introduce an `Avoid` behavior where villagers maintain a safe distance from the passing robot or soldier. Alternatively, villagers might be happy to see the soldiers and approach them and initiate a conversation. The presence of the robot might invoke the curiosity of the villagers by capturing their attention or even having them follow the robot for a short distance. Figure 5 illustrates the behavioral response of villagers to the presence of the soldier-robot team, and Figure 6(b) shows different reactions of the villagers in response to the soldier-robot team. Table 2 enumerates some behaviors that we authored for villagers in the marketplace. Mission-specific behaviors where the villagers act in a particular manner (e.g., a crowd gathering or a procession) can be easily added without requiring any change to the existing behavior library. The supplementary document provides details of additional authored behaviors.

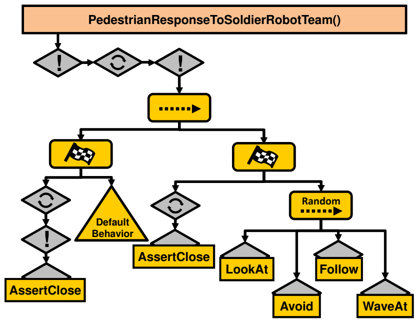

Fig. 5: Example PBT to author pedestrian response to the presence of the soldier-robot team.

Behavior	Description
`Wander`	Roam the marketplace, stop at stalls and look at goods of interest.
`Conversation`	Arrive at a common meeting point and have a conversation with other villagers. Different conversation behaviors can be authored to convey concord, or disagreement.
`Haggle`	Interaction between vendor and villager to agree upon a price to purchase goods.
`Keep Safe Distance`	Cautiously observe soldier or robot as it passes by and keep a safe distance.
`Interact`	Wave to passing soldier and strike a conversation.
`Follow`	Follow soldier or robot.
`Disperse`	Disperse from a specific area in the marketplace.

Table 2: Behaviors defined for the villagers in the marketplace.

(a) (b) (c)

Fig. 6: Screenshots of simulation. (a) Villagers in a Middle Eastern marketplace. (b) The virtual populace responding to the presence of the soldier-robot team. (c) The robot carrying out a command from the human-controlled team captain to investigate suspicious contraband, by taking pictures from different vantage points.

5 Missions

To demonstrate the effectiveness of our framework, we author and demonstrate the following missions, exercising different types of interactions between the team captain, his soldier and robot team, and the populace.

Cordon and Search. The soldier-robot team is required to cordon off a particular area in the marketplace, instruct the villagers to clear the region, and search the premises for any suspicious goods such as firearms. The soldier-robot team arrive at the designated location and the team captain cordons off the region by instructing the villagers to clear the area. Noticing the presence of suspicious contraband, he asks his teammate (computer controlled) to guard the location and instructs the robot to inspect the goods. The robot transmits images of weapons to the team captain, who upon inspecting the images, asks the robot to retrieve the weapons. The emergent nature of the interactions between the villagers and the soldier-robot team ensures that each run of missions provides a different experience to the user. Figure 6(c) shows the robot investigating suspicious contraband while the team captain and other soldier cordon off the area.

Clear Crowd. This mission requires the team to clear a gathering of villagers in the center of the marketplace and is authored by defining a `CrowdGathering` event where a crowd of villagers convene at the center of a marketplace to engage in a discussion or procession. The team arrives at the scene and disbands the crowd by instructing the villagers to disperse. The response of the villagers may not always be amicable where certain members of the crowd might express discontent and be reluctant to follow orders from the soldiers.

Monitor Suspicious Villagers. This mission requires the SR team to monitor the activity of one or more suspicious villagers in the marketplace. In one run of the mission, the team captain instructs the robot to investigate by covertly following the villager. The villager, unaware of the robot, leads him to a deserted alleyway with hidden weapons. The robot transmits proof of the discovery by sending pictures to the team captain, enabling the team to take further action.

The missions described here and shown in the supplementary video demonstrate the flexibility of our platform in authoring and simulating a large variety of training scenarios just by defining new events involving any of the inhabitant, soldier and robotic agents. Existing missions can be modified by changing the event definition or the conditions when events are triggered, providing an entirely new training experience. The supplementary document provides additional details for the authored missions.

6 Conclusion and Future Work

This paper describes a platform for authoring functional, purposeful, virtual humans and demonstrates its application to military training simulations. We author a virtual marketplace scenario with an interactive populace that exhibits meaningful interactions with each other and a player controlled human-robot soldier team. Different missions can be easily authored and customized to create a large variety of training scenarios where the simulation is open-ended and the outcome of the mission is entirely dependent on user input and inherent inhabitant interactions. Our framework provides a strong foundation for military training simulations and human soldier-robot team experiments, and can be easily adapted to other interactive applications.

For future work, we will introduce social and cultural parameters to control the simulation, animation, and behavior of the populace. This will allow us to author scenarios that are an accurate representation of a specific region's populace, enhancing immersion and training experience. Currently, only one user can participate in a training exercise: the other members have programmed responses to user commands. This platform can be easily extended to support multi-user training. We will also provide a graphical user interface to simplify the behavior authoring process and make it more accessible to subject matter experts. The underlying ADAPT simulation platform is released open-source for the community to develop their own applications and can be downloaded here: http://cg.cis.upenn.edu/hms/research/ADAPT/.

Our system makes it possible to investigate important research directions including the selection of the right set of robot behaviors, and defining an appropriate tactical command vocabulary for human-robot interaction. We currently support a traditional keyboard-mouse control interface which will be extended using multi-modal control schemes such as gestures and voice commands.

References

1. Badler, N.: Virtual Crowds: Methods, Simulation, and Control (Synthesis Lectures on Computer Graphics and Animation). Morgan and Claypool (2008)
2. Braun, A., Musse, S.R., de Oliveira, L.P.L., Bodmann, B.E.J.: Modeling individual behaviors in crowd simulation. CASA 0, 143 (2003)
3. Durupinar, F., Allbeck, J., Pelechano, N., Badler, N.: Creating crowd variation with the ocean personality model. In: AAMAS'08. pp. 1217–1220 (2008)
4. Fleischman, M., Roy, D.: Representing intentions in a cognitive model of language acquisition: Effects of phrase structure on situated verb learning. In: AAAI '07. pp. 7–12
5. Funge, J., Tu, X., Terzopoulos, D.: Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In: ACM SIGGRAPH. pp. 29–38 (1999)
6. Gonzalez, J.P., Lacaze, A., Childers, M.: Using RIVET for Parametric Analysis of Robotic Systems. In: GVSETS (2009)
7. Kapadia, M., Singh, S., Reinman, G., Faloutsos, P.: A behavior-authoring framework for multiactor simulations. IEEE CGA 31(6), 45–55 (Nov 2011)
8. Koechling, J., Crane, A., Raibert, M.: Applications of realistic human entities using DI-Guy. In: Proceedings of Spring Simulation Interoperability Workshop (1998)
9. Loyall, A.B.: Believable agents: building interactive personalities. Ph.D. thesis, Pittsburgh, PA, USA (1997)
10. Massive Software Inc.: Massive: Simulating life (2010), www.massivesofware.com
11. Mateas, M.: Interactive drama, art and artificial intelligence. Ph.D. thesis, Pittsburgh, PA, USA (2002)
12. Menou, E.: Real-time character animation using multi-layered scripts and space-time optimization. In: ICVS '01. pp. 135–144 (2001)
13. MetaVR Inc.: Metavr (2012), www.metavr.com
14. Orkin, J.: Agent architecture considerations for real-time planning in games. In: AIIDE. pp. 105–110 (2005)
15. Perlin, K., Goldberg, A.: Improv: a system for scripting interactive actors in virtual worlds. In: ACM SIGGRAPH. pp. 205–216 (1996)
16. Shapiro, A.: Building a character animation system. pp. 98–109. MIG (2011)
17. Shoulson, A., Badler, N.I.: Event-centric control for background agents. In: ICIDS. pp. 193–198 (2011)
18. Shoulson, A., Garcia, F., Jones, M., Mead, R., Badler, N.I.: Parameterizing Behavior Trees. In: MIG. pp. 144–155. Springer (2011)
19. Shoulson, A., Marshak, N., Kapadia, M., Badler, N.I.: Adapt: the agent development and prototyping testbed. In: ACM SIGGRAPH I3D. pp. 9–18 (2013)
20. Yu, Q., Terzopoulos, D.: A decision network framework for the behavioral animation of virtual humans. In: ACM SIGGRAPH/EG SCA. pp. 119–128 (2007)