Behavioural Animation - A Report

Shrinath Shanbhag
KRESIT, IIT Mumbai
{svs@it.iitb.ac.in}

Abstract— **This report presents a preliminary survey of behavioural animation techniques in computer graphics. It investigates and lists the challenges of traditional and interactive animation that can be solved through appropriate use of behavoiral animation techniques. It compares and contrasts behavioural animation with procedural animation. It discusses the requirements and challenges of behavioral animation and concludes with a discussion of important work in this field.**

Keywords— **Behavioural animation, Boids, Multi-layered direction, Autonomous Actors**

I. Introduction

CREATING computer animations for entertainment, interactive simulations and real-time virtual environments presents many challenges. The medium is rich and therefore the amount of detail is high. An author has to manually specify, process and manage huge quantities of information. Animations are specified at a very low level making the whole process mundane, time-consuming, repetitive and not amenable to rapid changes. Little support if any exists to specify computer animation at high level. Even lesser support exists for reusing existing animations designed by creative people in different scenarios. Interactive simulations and real-time virtual environments add an additional requirement - real-time interactivity - which cannot be handled by conventional linear processes.

Traditionally, researchers in computer graphics have been attracted towards realism and speed. Significant advances have been made in motion specification and animation through simulation of physically based models. Examples of the former include use of IK for interactive motion specification and that of later include automatic gait generation for articulated objects using motion controllers [2][7]. Researchers have used motion capture techniques to generate animation data. However, such data has been found to be too specific to a particular scenario and non-conducive to editing and modification. Methods for space-time edits, motion warping [12] and motion signal processing [11] are being researched actively by the graphics community. Reusability and complexity management, however, have received little or no attention. The need of the hour is to investigate techniques that address these issues. Behavioural animation techniques have the potential of handling these problems.

II. What is behavioural animation ?

Behavioural animation refers to the use of embodied autonomous objects to generate computer animation, by simulation of modelled behaviour. Behavioural animation is not restricted to living objects alone. Any object requiring animation, in addition to what can be specified reasonably well through rigid body dynamics, can benefit from the use of behavioural animation. Behavioural animation is especially suited to self-aware objects that possess internal motivations and/or need to react to external factors.

III. Why use behavioural animation ?

Behaviour modelling is context dependent. Objects with appropriately modelled behaviour can result in the following benefits.

1. Behavioural animation techniques can be used to rapidly specify and generate rich and complex animations, which would take orders of magnitude more time to produce using traditional techniques.
2. Can be used in a class of applications which require non-linearity and interactivity. Traditional techniques, which tend to be linear and non-interactive in nature, cannot be applied to this class of applications.
3. Behaviour when sufficiently parameterised can allow creative control and ease of specification for generating large variety of animations from a given behaviour-specification/framework.
4. Behavioural animation enables creating reusable characters. This kind of reuse has not been possible before.

IV. Behavioural Animation and Procedural Animation

Behavioural animation shares some characteristics of procedural animation. However the two differ in a number of areas. For most practical purposes they can be used in a manner so as to complement each other.

A. Similarities

1. As in procedural animation, animation is produced through simulation.
2. Description can be parameterised to generate variations of the basic animation.

B. Differences

1. Procedural and behavioural animations work at different levels. Procedural animation tends to control the geometric and visual attributes of an animated action. Behavioural animation is used to control the motivations governing the desire for a particular action.
2. In procedural animation, a single procedure is active at a time. Whereas in behavioural animation, many behaviours compete for control of the animated object.
3. In a procedural animation system, the animation procedure is invoked externally. The procedure itself has no knowledge of who invoked it or why it was invoked. A behavioural animaton system is autonomous. It activates certain behaviours based on internal motivations and external factors.
4. An animation procedure runs from start to finish uninterrupted. A behavioural animation rule can be prempted at any time by another competing rule.

V. What are the requirements of behavioural animation ?

One of the prime requirements of animation, irrespective of the method of generation, is believability. Believability does not imply realism. Believability depends on the context of animation. For the purposes of our discussion we limit ourselves to the realm of entertainment (as in movies), interactive fiction (as in computer games, muds) and networked virtual environments with avatars.

Behavioural animation requires autonomous creatures be augmented with sensors to observe their environments, and rules to react to stimulus. In addition they also need to represent internal motivations and preferences. There is a strong need to make these creatures directable. The creatures are embodied and hence they also need to be modelled for visualisation.

VI. Modelling autonomous animated objects

Autonomous animated objects are modelled in layers. The following figure depicts a general layering scheme

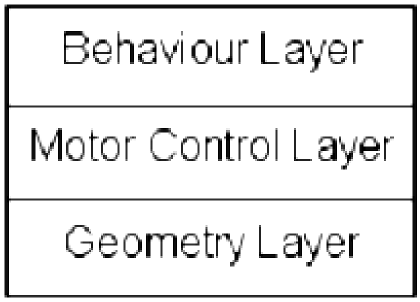

Fig. 1. Layered model for behavioural animation

Behaviour layer encapsulates the internal mental model associated with the creature. It comprises of motivation variables and action rules. Motivation variables hold the current value of the motivation. Motivation of the creature results in selection of one from a set of action rules, based on a selection criteria. Motivation variables can be affected by action rules, external factors and time.

The action rules determine the set of independent actions that a creature can perform, to satisfy its motivations. Actions are often organized in a loose hierarchy. Higher up in the hierarchy are top level goals. Task level goals appear lower down. There are multiple hierarchies and lower level nodes can be part of multiple hierarchies.

Associated with the mental model is a selection mechanism which determines a high level goal from this set of competing goals at each simulation cycle. Sub-goals are inferred from the specified high level goal. Sub-goals execute in parallel. Eg. The character can yawn as it walks.

Motor control layer implements the goals/sub-goals selected by the behavioural layer. It does so by generating and controlling the appropriate (geometric) parameters (dofs) required for generating the desired animation. Motor control layer needs to arbitrate between conflicting requirements posed by task-level behaviour's executing in parallel.

The geometry layer captures the visual and physical attributes of the creature. Detailed geometric models are required to create expressive characters.

The separation between behaviour and motor skills is important. It separates specification of motor skills from the task. The motor skill is free to implement the task as it sees fit. For e.g. if moving from point A to B is defined as a task, it may be accomplished by a simple translation or an animated walk, depending upon the creatures abilities. This allows similar behavioural models to be applied to different creatures.

A. Approaches to animating behaviour

Behaviour modelling techniques should be classified based on

1. Specification and control methods This refers to the way in which behaviours are specified. Behaviour may be specified procedurally or declaratively. A behaviour selection mechanism also needs to be determined. Control can be scripted or based on sensing the environment using virtual sensors.
2. Generality of the method The types of animations a technique can generate are good classifiers. Certain techniques are very specific and apply to one type of behaviour (eg flocking) or one type of creature (eg.

fishes).
3. Directability Directability refers to the degree to which an autonomous creature may be externally controlled. Though making an object autonomous and directable are conflicting goals, a high degree of directability is a desirable feature.
4. Ease of authoring Ease of authoring though related to ways of specification and control methods is not the same. The ease of authoring depends primarily on the type of primitives provided by the system, its user interface and extensibility mechanisms.

Modelling techniques can range from user friendly to arcane. User-friendly techniques may be amenable to an iterative development cycle and at the same time offer reduced set of options. Arcane techniques on the other hand can be quite powerful and may require significant investment of development efforts.

B. Challenges in behavioural modelling

The task of digitally cloning a living creature is probably unattainable. The challenge lies in modelling the creatures well enough, for specific environments, to respond in ways that makes them believable to their human observers.

In addition, the following goals need to be addressed.
- Design of high level primitives for ease of specification
- Creating an appropriate rule selection mechanism
- Modelling for interaction and cooperation with other virtual characters
- Providing hooks for external control (directability)

C. Challenges in top level rule selection

Deciding on the right action from a set of actions is complicated due to the following factors
- Problems inherent in sensing and perception make a creatures perception of the world incomplete at best and erroneous at worst
- Competing goals work at cross purposes resulting in an effect of the creature oscillating among competing activities. This phenomenon is known as dithering [10].
- Prevent starvation of lower priority goals. An important goal may be unattainable and pursuit of that goal may prevent the satisfaction of lower priority, attainable goals.

D. Challenges in motor skill design

- Behavioural characters often need to execute multiple motor skills in parallel. Designing to support parallelism is a challenge. Multiple actions can act on the same variable (dof).
- Transitions between poses are another challenging area. Simple linear interpolation of dof's does not always result in desired animation. The transitions should appear natural and should not take unreasonably long periods.
- Animations need to be consistent with physical constraints. Animations need to be believable and non-repetitive.

VII. Related Work

A. Boids (1987)

C.W. Reynolds [16] reported the use of behavioral animation, in his seminal paper, to control flocking behavior of bird like objects called boids. Though the behavior model used here was simple minded, the results produced were impressive. A set of three behaviors named flock centering, collision avoidance and velocity matching were used to control a geometrical flight model of boids resulting in animation resembling that of real life birds. These boids had the capability to navigate static obstacles. True to real life, flocks would split and re-merge if required to maneuver around objects in the environment.

In this approach, every boid had a local view of the whole environment. This view comprised of its neighbors and any obstacles within its view. Looking for objects within a sphere of influence simulated vision. The behaviors controlled the geometric flight model by computing a desired velocity vector at each simulation interval.

Even this simple model exhibited all characteristics of behavioral systems. Conflicting behaviors required arbitration. Allocating static priorities did this. For instance, collision avoidance had precedence over, flock centering which had precedence over velocity matching.

B. Fishes (1994)

Xiaoyuan Tu, Demetri Terzopoulos [13] simulated artificial fishes using behavioral animation. Their work was more involved than that of Reynolds. Fish movement was physically achieved using muscle based, hydrodynamic locomotion by simulating the forces of interaction of a deformable body in aquatic medium. The fishes were equipped with perception sensors. The behavior modeling system coordinated locomotion based on perception and other modeled internal motivations. Two kinds of sensors were modeled - the vision sensor and the temperature sensor. The temperature sensor sampled the ambient temperature of water at the center of the artifical fishes body. The vision sensor extracted important information related to color, sizes, distances and identities of objects in its neighbourhoods. It had access to the rendering database for this purpose and was not based on vision techniques. The fishes behavior

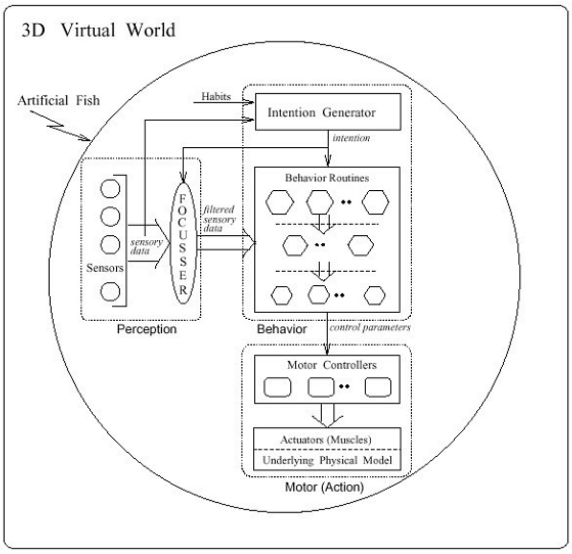

Fig. 2. Control and information flow in an artificial fish

model accounted for habits, mental state and incoming sensory information. Habits are parameters which store fish preferences such as whether it likes brightness, darkness, cold, warmth, schooling, is male or female etc. There is no equivalent to this in the boids model. Each fish contained three mental state variables - hunger, libido and fear. These were used to trigger top level behaviors. Eight behavior routines are modeled including avoiding-static-obstacles, avoiding-fish, eating-food, mating, leaving, wandering, escaping and schooling. The intention generator updates each of these variables during each update cycle taking into consideration things such as food consumed, digestion rate, elapsed time, time since last mating, distance to predator etc. In addition the intention generator also implements collision division. The intentions are prioritized in the following order - collision avoidance, fear from predators, hunger and libido. Dithering among behavior is avoided by building a memory model which remembers the last behavior chosen. Lower priority behaviors can be interrupted by higher priority behaviors. In such a case up to one previously selected behavior is remembered in an approximation to short term memory. Different types of fishes are modeled - predators, preys and pacifists.

C. Directable Autonomous Creatures (1995)

Most research thus far was aimed at defining only one specific object or behavior. B. Blumberg, T. Galyean [10], unlike their predecessors, defined a general framework for creating behaviour based characters. These authors brought out the importance of directability in behavioral animation. The requirements that characters be directable and yet autonomous are conflicting. The authors made three primary contributions regarding autonomous characters. They described

1. An approach to control which allows an external entity to direct a virtual character at a number of different levels.
2. A general behavioral model for perception and action-selection in autonomous animated creatures but which also supports external control.
3. A layered architecture which supports extensibility, reusability and multiple levels of direction.

They suggested that objects be modeled as shown in the following figure. Geometry modeled the autonomous object geometrically. DOF's are "knobs" that when used modify the underlying geometry. They are the mechanism by which creatures are repositioned and reshaped. Each DOF could be locked by a motor skill, restricting it by anyone else until unlocked. Motor skill layer is responsible for producing complicated motion. Motor skills rely heavily on degrees of freedom to do their work. It utilized one or more DOF's to produce coordinated movement.

The controller layer abstracts out the underlying motor skills. The primary job of the controller is to map commands such as "forward", "turn", "halt", "look at" etc. into calls to turn on or turn of the appropriate motor skills.

Behaviors implemented high level capabilities such as, "find food and eat", or "sit down and shake", as well as low level capabilities such as "move to" or "avoid obstacle" by issuing the appropriate motor commands to the controller.

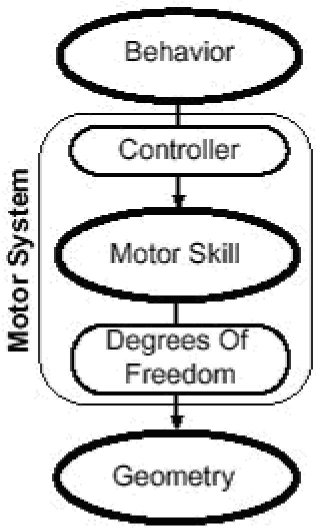

Fig. 3. Layered architecture for autonomous characters

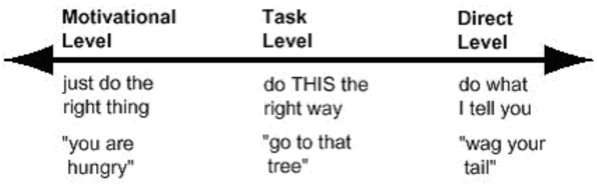

Fig. 4. Multiple levels of directability

The layered architecture provided for multiple levels of control as shown in the following figure

By providing the ability to "direct" the creature at multiple levels the animator or developer could choose the appropriate level of control for a given situation. The behavior module consisted of rules organized in a loose hierarchy and mechanisms for rule selection. Each rule competed against all other rules in its category for selection. Selection was based on a fitness value evaluated at each tick interval. The fitness value depended on inputs from external sensors, internal motivation variables and fitness values of other rules. An iterative procedure computed fitness values for all rules till one of them emerged the winner. The creature could be directed by changing its motivation or sensor variables and by directly scheduling tasks for execution. Any rule or motor action could be executed directly.

D. Improv (1996)

K. Perlin and A. Goldberg [8] created a system for the creation of real-time behavior-based animated actors called Improv. Improv provided tools to create actors that responded to users and to each other in real-time, with personalities and moods consistent with the author's goals and intentions. It consisted of two subsystems. The first, an animation engine, that used procedural techniques to enable authors to create layered, continuous, non-repetitive motions and smooth transitions between them. The second, a behavior engine, that enabled authors to create sophisticated rules governing how actors communicate, change and make decisions. The combined system provided an integrated set of tools for authoring the "minds" and "bodies" of interactive actors. The system used an English-style scripting language so that creative experts who were not primarily programmers could create powerful interactive applications.

The Improv system used a behavior model similar to that proposed by [10]. The animation engine allowed authors to specify actions in layers. The layering mechanism aided the authors in manually specifying parallelly executable actions. The "layers" metaphor was borrowed from run of the mill image processing applications. The layers were identified by the GROUP keyword of Improv's scripting language. The authors could place actions in different groups, and these groups would be organized into a "back-to-front" order. Also the author could "select" any action. The Improv system followed the following two action composition rules.

Actions which are in the same group compete with each other. At any moment, every action possess some weight. When an action is selected, its weight transitions smoothly from zero to one. Meanwhile, the weights of all other actions in the same group transition smoothly down to zero.

Actions in groups which are further forward obscure those in groups which are further back.

The following e.g. from [8] would make thing clear.

Suppose the author specifies the following action grouping.

```
GROUP Stances
ACTION Stand
ACTION Walk

GROUP Gestures
ACTION No_waving
ACTION Wave_left
ACTION Wave_right

GROUP Momentary
ACTION No_scratching
ACTION Scratch_head_left
```

Now, assuming the actions are selected in the following order

```
Stand
Walk
Wave_left
Scratch_head_left
No_scratching
Wave_right
```

The actor would start to walk. While continuing to walk he would wave his left hand. The he would scratch his head with his left hand and resume waving again. Finally he would switch over to waiving with his right hand.

The behavior engine allowed for a similar mechanism for specifying behavior scripts. Like actions, scripts were organized into groups. However unlike actions, when a script within a group was selected, any other script that was running in the same group stopped immediately.

```
Sample script
define SCRIPT "greeting"
{
{ "enter" }
```

```
{ wait 4 seconds }
{ "turn to camera" }
{ wait 1 second }
{ "wave" for 2 seconds
  "talk" for 6 seconds }
{ wait 3 seconds }
{ "sit" }
{ wait 5 seconds }
{ "bow" toward "Camera" }
{ wait 2 seconds }
{ "leave" }
}

Sample script layers

DAY_PLANS  Waking Morning Lunch
Afternoon Dinner Evening
...
ACTIVITIES Resting Working Dining
Conversing Performing
...
BEHAVIOR  Sleeping Eating Talking Joking
Arguing Listening Dancing
```

The author organized into the same group those scripts that represented alternative modes that an actor be in at some level of abstraction. In general, the author first specified those groups of scripts that controlled longer term goals and plans. The last scripts were those that were most physical.

In addition to commands that explicitly triggered specific actions, an author could specify that an actor choose randomly from a set of scripts. A probability could be associated with each of the actions in the script.

Information about an actor and his relationship to his environment were stored in actor properties. These properties could be used to describe aspects of an actors personality, mood and even his relationship to other actors. The properties could be updated by any script running in the system. In addition Improv supported the concept of decision rules that allowed selective execution of rules based on the state of the actors properties.

Improv was the most flexible and easy to use system compared to systems by other researchers before it.

E. Conclusion

Behavioral animation is a technique with potential to change the way animations are created. It is well suited to applications that demand interactivity. Researchers working in this area have already demonstrated the potential. A lot remains to be done.

References

[1] Andrew Gildfind, Evolving Performance Control Systems for Digital Puppetry, Ph.D. Thesis, September 2001, RMIT University http://goanna.cs.rmit.edu.au/ gildfind/thesis/

[2] Petros Floustos, Michiel van de Panne, Demetri Terzopoulos, Composable Controllers for Physics-Based Character Animation, Proceedings of SIGGRAPH'01, 2001, pp. 251-260

[3] Peter Jason Willemsen, Behaviour and Scenario Modelling for Real-Time Virtual Environments, Ph.D. Thesis, 2000, University of Iowa, http://www.cs.uiowa.edu/ willemsn/phd/thesis.pdf

[4] Daniel Thallman, Challenges for the Research in Virtual Humans, Agents 2000

[5] Z Popovic and A. Witkin, Physically based motion transformation, Proceedings of SIGGRAPH'99, 1999, pp. 11-20

[6] John D. Funge, Making Them Behave : Cognitive models for Computer Animation, Phd Thesis, University of Toronto, 1998

[7] Michiel van de Panne, Making them move : Motor control for animated humans and animals, European Control Conference, July 1-4, 1997, Brussels, pp. 191-210

[8] Ken Perlin and Athomas Goldberg, Improv: A System for Scripting Interactive Actors in Virtual Worlds, Proceedings of SIGGRAPH'96, 1996, pp. 205-216

[9] Daniel Thalmann, Physical, Behavioral and Sensor-Based Animaton, Graphicon'96

[10] B. Blumberg, T. Galyean, Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments, Proceedings of SIGGRAPH'95, 1995, pp. 47-54

[11] A Bruderlin and L Williams, Motion Signal Processing, Proceedings of SIGGRAPH'95, 1995, pp. 97-104

[12] A. Witkin and Z. Popovi'c, Motion Warping, Proceedings of SIGGRAPH'95, 1995, pp. 105-107

[13] Xiaoyuan Tu and Demetri Terzopoulos, Artificial Fishes : Physics, Locomotion, Perception, Behaviour, Proceedings of SIGGRAPH'94, 1994, pp. 42-48

[14] N. Magnenat Thalmann, D. Thalmann, Complex Models for Animating Synthetic Actors, IEEE Computer Graphics and Applications, 11(5), 1991, pp. 32-44

[15] N. Badler, B. Barsky, D. Zeltzer, Making Them Move: Mechanics, Control and Animation of Articulated Figures, Morgan Kaufmann Publishers, San Mateo, CA, 1991

[16] Craig W. Reynolds, Flocks, Herds and Schools: A Distributed Behavioural Model, Proceedings of SIGGRAPH'87, 1987, pp. 25-34

[17] John Lasseter, Principles of Traditional Animation Applied to 3D Computer Graphics, Proceedings of SIGGRAPH'87, 1987, pp. 35-44

Shrinath Shanbhag is a research scholar at the Kanwal Rekhi School for Information Technology (KRESIT), IIT Mumbai. He is working on the topic of behavioural animation. He is employed by the Graphics and CAD division of the National Centre for Software Technology, Mumbai as a Senior Staff Scientist. His interests lie in the fields of Computer Graphics, Realtime Rendering, Computer Gaming, Digital Story Telling and Indian Language Computing.