Integrating internal behavioural models with external expression

Marco Gillies,[*]
University College London in partnership with BTexact Technologies,
Ross Building pp1, Adastral Park, Ipswich IP5 3RE, UK, m.gillies@cs.ucl.ac.uk

Daniel Ballin
Radical Multimedia Lab, BTexact Technologies,
Ross Building pp1, Adastral Park, Ipswich IP5 3RE, UK, daniel.ballin@bt.com

Neil Dodsgon
Cambridge University Computer Laboratory,
William Gates Building, J. J. Thompson Avenue, Cambridge, UK,

m.gillies@cs.ucl.ac.uk, daniel.ballin@bt.com,Neil.Dodgson@cl.cam.ac.uk

Abstract

Users will believe in a virtual character more if they can empathise with it and understand what 'makes it tick'. This will be helped by making the motivations of the character, and other processes that go towards creating its behaviour, clear to the user. This paper proposes that this can be achieved by linking the behavioural or cognitive system of the character to expressive behaviour. This idea is discussed in general and then demonstrated with an implementation that links a simulation of perception to the animation of a character's eyes.

1 Introduction

Actors and animators are able to breath life into characters by understanding the inner thoughts, motivations and emotions that produce that character's actions and express them through their acting or animation. It has long been thought that understanding the inner motivations and personality of the character is vital to external expression. When an audience watches a character (whether it is in a play, film, animation or video game) they must be able to understand the character's feelings and empathize with that character. It is not good enough for the character to perform a sequence of actions if the audience is unable to relate these actions to the motivations, emotions and personality of the character; if they cannot understand why the character performs these actions. They must see the character's actions as the result of motivations and personality, a coherent internal state; not merely as an arbitrary set of events. For this to be the case the user must be able to understand the internal factors (perception, motivation, emotion, cognition etc.) that result in the character's actions. It is true that characters whose inner feelings and motivation are mysterious can often be appealing. However, this is largely because we are used to characters whose inner motivations are made clear to the audience. Also, mysterious characters often have a lot of effort put into their internal motivations and then into expressing them in a paradoxical or obscure way, rather than merely making their behaviour confusing.

When designing autonomous or semi-autonomous characters for computer animation it is vital to have a good model of the internal processes that produce the character's external behaviour. If the character is to be compelling and interesting it must also have a lot of expressive behaviour that gives a human-like impression of internal feelings. This paper will argue that these two aspects should be tightly coupled so that the audience/user can understand the motivations etc. of the character. We will elaborate this idea in section 3 and then describe two examples. Section 4 sketches a conceptual example of how this system might work implemented with a complex behavioural architecture. Section 5 describes a concrete implementation of such a system, focusing on one aspect, visual perception.

2 Previous Work

Numerous researchers have worked on behavioural architectures for controlling the behaviour of autonomous characters. Notable examples include Reynolds' flocking boyds[13]; Tu and Terzopoulos' artificial fishes[17]; Blumberg and Galyean's Silas T. Dog system[2], and Perlin and Goldberg's Improv system [11]. There is also a large body of work on animating expressive characters, including Vilhjálmsson and Cassell's

[*] Formerly Cambridge University Computer Laboratory

BodyChat[18]; Cassell, Vilhjálmsson and Bickmore's BEAT[3]; Chi and Badler's EMOTE[4]; Amaya, Bruderlin and Calvert's motion transformations[1], and Terzopoulos and Waters' facial animation system[15]. There has been also been work on combining the two and using behavioural architectures to drive expression, most notably in Blumberg and Galyean's system[2] the internal states of the character control various expressive behaviours such as tail wagging.

Our implementation concerns visual perception and eye gaze. Work on simulating visual perception includes Terzopoulos and Rabie's simulation of vision for fishes[14]; and Noser, Renault, Thalmann and Magnenat-Thalmann's work on simulating vision for navigation[10, 12]. There has been a large amount of interest in animating eye gaze for social interaction, for example, Thórisson[16]; Vilhjálmsson and Cassell[18], and Colburn, Cohen and Drucker[6]. These systems tend to focus either vision or the expressive aspects of gaze whereas we have designed our system both for modelling perception and animating eye gaze simultaneously. Our own implementation of eye gaze and visual perception bases some of its ideas on the work of Chopra-Khullar and Badler[5], who have produced a system for autonomously animating eye gaze. Our system improves on theirs in a number of ways that we shall discuss the the course of this paper.

3 Integrating behavioural controllers and expression

Actors and animators take great care to create coherent motivations and personality, a coherent internal state, around a character's actions and then to express it in the characters face, voice or body language. Autonomous characters do have a coherent internal state in the behavioural agents that control them, however, it is very important to communicate this to the viewer in an understandable way through the character's expressive behaviour. [1] In order to produce autonomous behaviour a character will typically have a control architecture that depends on a number of internal subsystems (e.g. perception, motivation, emotion and cognition) that result in a sequence of actions. Figure 1 gives a schematic of one possible system. In order to make the character's actions easier to understand it would help to communicate some of these internal processes to the viewer. We propose doing this by linking each of these internal subsystems directly to some form of expressive behaviour as shown in figure 2.

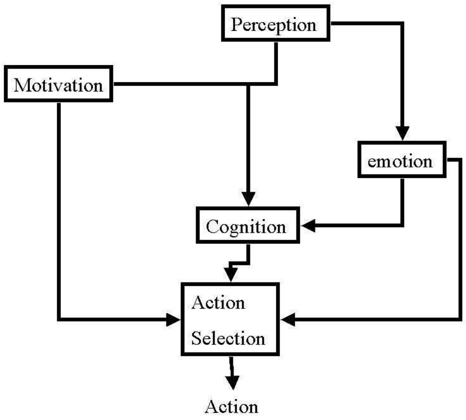

Figure 1: A possible architecture for a behavioural controller.

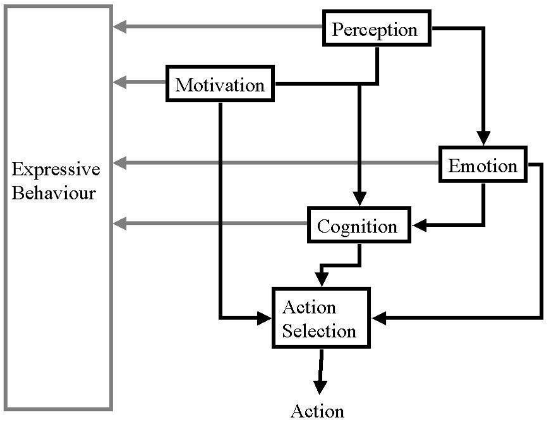

Figure 2: Each element of the behavioural controller should be directly linked to expressive behaviour.

One interesting result of tying expression directly to the internal behavioural model is that the model itself will have to be constructed in such a way that it can be expressed understandably to the user. This would probably mean that the model should be relatively easy to understand, at least at a high level of abstraction. This would also be a good thing for other reasons. Firstly, an incomprehensible behavioural system is more likely to produce incomprehensible behaviour. Secondly, we believe that in the future, behavioural architectures will be created as parameterised frameworks. The characters themselves will be created by non programmers (writers or animators) who will adjust the parameters of these frameworks to create the actual behaviour they want. In this sort of situation the harder the architecture is to understand the harder it will be for the character designer to create the characters they want.

[1] Note that the situation is rather different from real people, who can often be rather inexpressive and whose internal motivations can be rather hard to understand. However, this is a luxury that is not available to characters whether acted, animated or computer controlled. We are very ready to ascribe life to real people but it is much harder for characters to maintain the illusion of life and be compelling.

4 Conceptual Example

A king walks into a room carrying a sword. In the room there is a man and a broken lamp; the king stabs the man. This is an action based description of a story. However, there is a lot going on in the story besides these actions. If we take the king to be a virtual agent and look what happens in his behavioural controllers it might be something like this. The king has a motivation to get his magic lamp from his secret chamber. On arriving in his secret chamber his perceptual system registers a man and the broken lamp. As only the king should know about the secret chamber this triggers an expectation violation that results in an emotion of surprise. The king's cognitive system appraises the situation and decides that the man must be a thief. This triggers an emotion of anger. The cognitive and emotional systems produce a new goal, to kill the thief. The cognitive systems then determines that the best strategy is to stab the thief with the sword.

How might this be expressed? Each behavioural subsystem would trigger various expressive behaviours; possibly these would be combined by an arbitration mechanism. Firstly the perceptual system would trigger the king's eyes to look at the objects he is perceiving (more on this in the next section). When the emotion system registers surprise it would generate a surprised facial expression. As the cognitive system assesses the situation it might trigger looks towards important objects that it is considering, the lamp and the thief. The motivation system would back this up with looks at the lamp, its object of desire. The emotion system then registers anger triggering an angry facial expression, possibly combined with other expressive behaviour such as shaking with rage. Finally, as the cognitive and emotional systems focus on the thief the king's gaze would fix on him; and the king stabs him.

5 Implemented example

This sort of complete behavioural controller would be very sophisticated by todays standards but we have implemented the sort of system we have described for visual perception. In particular we have implemented a simulation of visual attention that simultaneously determines what the character is aware of and controls the animation of the character's eye gaze. Some of the material described below is covered in more detail in Gillies and Dodgson[9].

The behaviour of our character is generated by a set of communicating behavioural controllers, for example, one controller is in charge of the character's walking another detects whether it is about to have a collision with an object. One controller, the attention manager controls most of the simulation of attention/eye gaze. As shown in figure 3 this receives requests to look at objects from other behavioural controllers. These requests can be of two types, immediate requests cause the character to look at the target as soon as the manager receives the request while a monitor request just causes the character to look at the target occasionally until it is told to stop by another request. Immediate requests allow exact synchronization of eye gaze and behaviour, which is problematic in queue based systems such as Chopra-Khullar and Badler's. When no requests are sent the manager continues to produce gaze behaviour, generating requests itself, a process that we call *undirected attention*. The attention manager arbitrates between these requests to choose one that is active at a given moment of time. The attention manager processes the active request; animating the character's eyes and head so that it looks at the target of the request. At the same time it sends this target, the focus of attention, to other behavioural controllers so that they can use this information to direct the character's behaviour. If the character was not aware of the object beforehand, it is added to a list of objects that the character is aware of. The number of objects that behavioural tests have to be performed on can be reduced to those on this list, some times tests only need to be performed on the current focus of attention.

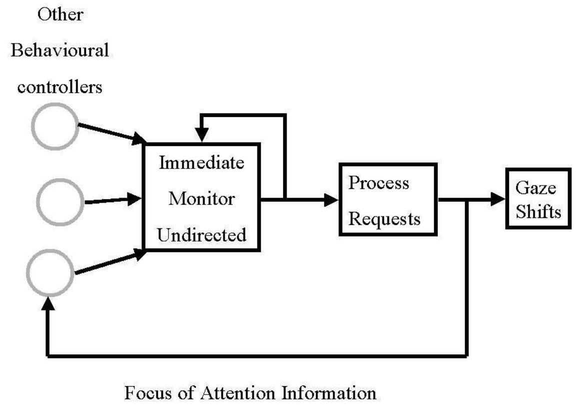

Figure 3: The Attention Manager receives requests of two different types: Immediate and Monitor requests. It chooses between these, or if none are present performs undirected attention to generate a request. When a request has been chosen it must be processed to turn it into an attention shift and eye movement. If necessary it is then sent to the gaze shift behaviour which moves the character's eyes.

When behavioural controllers receive the focus of attention from the attention manager they can extract information from it. Behavioural controllers can get two types of information from an object. Firstly they can query for a number of simple geometric properties that a person could reasonably know about an object that they see, for example its position and velocity. The other type of information is intended to model higher level features that some one could know about an object but that cannot be modelled in terms of the objects geometry or other information that is stored in the computer, for example, beauty or looking like a cup. This information is represented as a set of *object features*, these are simply textual tags with numerical values. The exact meaning of object features vary from feature to feature. For some the presence of a feature is enough to indicate that the object has that property, for example, if the feature "person" is present then the object is another character otherwise it is an inanimate object. Other features depend on the numerical value, for example, an object with a "beauty" feature of 0.7 is more beautiful than one with a value of 0.5. One use of object features is in undirected attention (see previous paragraph). Undirected attention attempts to find interesting objects for the character to look at. Objects are considered in turn. The character has a list of features and for each it has a number representing how interesting it is. This number is multiplied by the value of the feature in the object; the average of the results is taken and this is used as the probability of looking at the object.[2]

There is one set of behavioural controllers that is particularly important for making the character aware of its environment. These are the attention capture controllers. They detect objects in the periphery of the character's vision and make any important objects capture the character's attention by sending requests to the attention manager. Different attention capture controllers detect different types of object. One built-in attention capture controller detects moving objects, which often capture people's attention. Other attention capture controllers can be added dynamically to detect specific types of object, these detect objects having specific object features (as described above), section 5.2 describes a system that dynamically creates new attention capture controllers.

The next two sections describe two behavioural systems that use the attention mechanism described. The first describes an architecture for navigating an environment without walking into obstacles. The second describes a method for triggering simple actions based on objects that the character becomes aware of. Further uses of the attention mechanism are described in Gillies, Dodgson and Ballin[8].

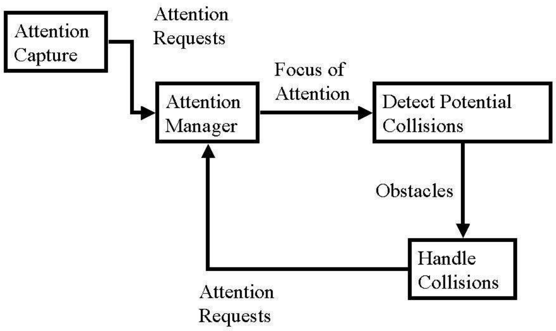

Figure 4: The ways in which information is passed between behavioural controllers.

5.1 Navigation

We have used this attention simulation to implement a character that can navigate an environment without colliding with obstacles. Cutting, Vishton and Barren[7] have produced evidence that people fixate objects (particularly moving objects) when deciding whether they are on a collision course with that object. To parallel this theory we only test objects for potential collisions when they are the focus of attention. This has two main advantages. Firstly only one object has to be tested at a time, greatly reducing the overhead of the test. Secondly, as the focus of attention is also the object that the character is looking at in the animation, the user/viewer can see that the character has seen the potential collision when it sees it. When a character is animated as looking at an object and then alters its course to avoid it, that character's awareness is made clear to the user. Thus the internal perception is linked to external expression as this paper argues. This makes the character's actions easier to understand, and they seem more purposeful.

When the navigation behaviour is active, attention capture controllers are activated to detect objects that the character might collide with, objects in the character's path and moving object. When the attention manager has a new focus of attention (from the attention capture controllers or another source) it is sent to any other behavioural controllers that request the information. One of these detects whether the character and object are on a collision course. If a potential collision is detected the object is then passed on to a set of behavioural controllers that avoid the collision. There are four possible strategies, for imminent side-on collisions with moving objects the character can stop and let the object go by; for less imminent side-on collisions the character can speed up to pass in front of the ob-

[2] Undirected attention is analogous to Chopra-Khullar and Badler's spontaneous looking. Our method is different, however. They use spacial frequency in an image of the object to determine whether the object is interesting. However, this is a rather simplistic method with little psychological basis (see Yarbus[19, p182] for a discussion) so we choose to add extra information to the system that allows animators some control over which object the character is likely to look at

ject; small static obstacles can be stepped over, and other types of collisions can be avoided by altering the character's path to move around the obstacle. Each of these methods is handled by a separate behavioural controller. When these controllers are handling a potential collision they send requests to look at the obstacle back to the attention manager. This is similar to the cognitive system having the character look at relevant objects in the conceptual example. Looking back at an object makes clear to the viewer why the character is altering its path or stopping etc. It also makes the character seem more involved with its environment. Figure 4 shows how objects are passed between behavioural controllers. Figure 5 shows a character walking on a street avoiding collisions with objects.

5.2 Triggering actions

We have also used our attention simulation to trigger actions in a character. The user specifies an action and an object feature that indicates which objects the action should be applied to. A new attention capture agent is created to detect these objects. When they are detected they can become the focus of attention. The action behavioural controllers can then react to them. Figure 6 shows a character reacting to an approaching ball by catching it. In this case the action controller first checks whether the ball is coming towards the character before activating the action. In our system actions can be created by the user based on a piece of pre-existing motion. The user can then specify how the action is invoked, it can only be applied to approaching object, or the character can actively search for a target.

6 Conclusion

We have developed a simulation of visual perception and eye gaze for behavioural animation. Though we have been unable to carry out formal evaluation we have received positive feedback to informal demonstrations. We believe that one of the main factors that make our animations compelling is the close coupling of the visual simulation used to generate behaviour and the expressive animation of eye gaze.

7 Acknowledgements

We would like to thank the UK Engineering and Physical Sciences research council for funding the bulk of this work and to BTexact technologies for funding the continuation of this research. We would also like to thank Professor Mel Slater and University College London for his help and support.

References

[1] Kenji Amaya, Armin Bruderlin, and Tom Calvert, *Emotion from motion*, Graphics Interface '96, Canadian Information Processing Society / Canadian Human-Computer Communications Society, May 1996, pp. 222–229.

[2] B. Blumberg and T. Galyean, *Multi-level direction of autonomous creatures for real-time virtual environments*, ACM SIGGRAPH, 1995, pp. 47–54.

[3] Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore, *Beat: the behavior expression animation toolkit*, ACM SIGGRAPH, 2001, pp. 477–486.

[4] Diane Chi, Monica Costa, Liwei Zhao, and Norman Badler, *The emote model for effort and shape*, ACM SIGGRAPH, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 173–182.

[5] Sonu Chopra-Khullar and Norman Badler, *Where to look? automating visual attending behaviors of virtual human characters*, Autonomous Agents Conference, 1999.

[6] Alex Colburn, Michael Cohen, and Steven Drucker, *The role of eye gaze in avatar mediated conversational interfaces*, Tech. report, Microsoft Research, 2000.

[7] James E. Cutting, Peter M. Vishton, and Paul A. Barren, *How do we avoid collision with stationary and moving obstacles*, Psychological Review **102** (1995), 627–651.

[8] Marco Gillies, Neil Dodgson, and Daniel Ballin, *Autonomous secondary gaze behaviours*, AISB workshop on Animating Expressive Characters for Social Interactions, April 2002.

[9] Marco Gillies and Neil Dodgson, *Eye movements and attention for behavioural animation*, Journal of Visualization and Computer Animation (2002), in press.

[10] H. Noser, O. Renault, D. Thalmann, and N. Magnenat-Thalman, *Navigation for digital actors based on synthetic vision, memory and learning*, Computers and Graphics **19** (1995), no. 1, 7–19.

[11] Ken Perlin and Athomas Goldberg, *Improv: a system for scripting interactive actors in virtual worlds*, Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM Press, 1996, pp. 205–216.

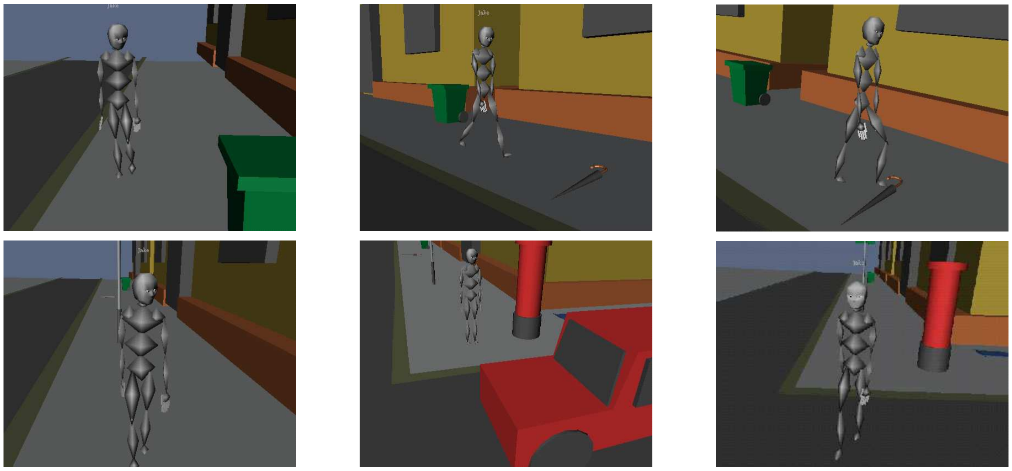

Figure 5: A character walking along a street. It changes its path to avoid colliding with the bin, steps over the umbrella and stops to let the car go past. What the character is aware of is determined by the attention mechanism and the character's gaze behaviour is appropriate to its behaviour.

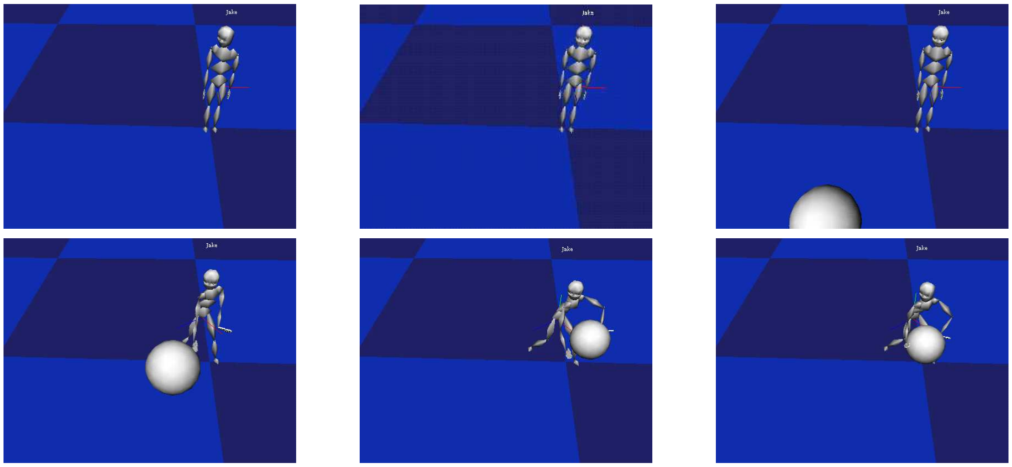

Figure 6: A character reacting to an approaching ball by catching it. The character looks around the environment (frame 1). It then spots the ball and watches it (frame 2). When the ball is close enough the catching is invoked.

[12] O. Renault, N. Magnenat-Thalmann, and D. Thalmann, *A vision-based approach to behavioral animation*, The Journal of Visualization and Computer Animation **1** (1990), no. 2, 18–21.

[13] Craig W. Reynolds, *Flocks, herds, and schools: A distributed behavioral model*, ACM SIGGRAPH, 1987, pp. 25–33.

[14] Demetri Terzopoulos and Tamer F. Rabie, *Animat vision: Active vision in artificial animals*, Fith International Conference on Computer Vision (ICCV'95), June 1995, pp. 801–808.

[15] Demetri Terzopoulos and K. Waters, *Physically-based facial modelling, analysis, and animation*, Journal of Visualization and Computer Animation **1** (1990), no. 2, 73–80.

[16] Kristin Thórisson, *Real-time decision making in multimodal face-to-face communication*, second ACM international conference on autonomous agents, 1998, pp. 16–23.

[17] Xiaoyuan Tu and Demitri Terzopoulos, *Artificial fishes: Physics, locomotion, perception, behavior*, ACM SIGGRAPH, 1994, pp. 43–49.

[18] Hannes Högni Vilhjálmsson and Justine Cassell, *Bodychat: Autonomous communicative behaviors in avatars*, second ACM international conference on autonomous agents, 1998.

[19] Alfred L. Yarbus, *Eye movements and vision*, Plenum Press, New York, 1967.