University of Pennsylvania

ScholarlyCommons

Technical Reports (CIS)

Department of Computer & Information Science

12-6-1995

Towards Personalities for Animated Agents With Reactive and Planning Behaviors

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

Barry D. Reich
University of Pennsylvania

Bonnie L. Webber
University of Pennsylvania, bonnie@inf.ed.ac.uk

Follow this and additional works at: https://repository.upenn.edu/cis_reports

Towards Personalities for Animated Agents With Reactive and Planning Behaviors

Abstract

We describe a framework for creating animated simulations of virtual human agents. The framework allows us to capture flexible patterns of activity, reactivity to a changing environment, and certain aspects of an agent personality model. Each leads to variation in how an animated simulation will be realized. As different parts of an activity make different demands oil an agent's resources and decision-making, our framework allows special-purpose reasoners and planners to be associated with only those phases of an activity where they are needed. Personality is reflected in locomotion choices which are guided by an agent model that interacts with the other components of the framework.

Comments

Towards Personalities for Animated Agents with Reactive and Planning Behaviors

MS-CIS-95-38

Norman I. Badler
Barry D. Reich
Bonnie L. Webber
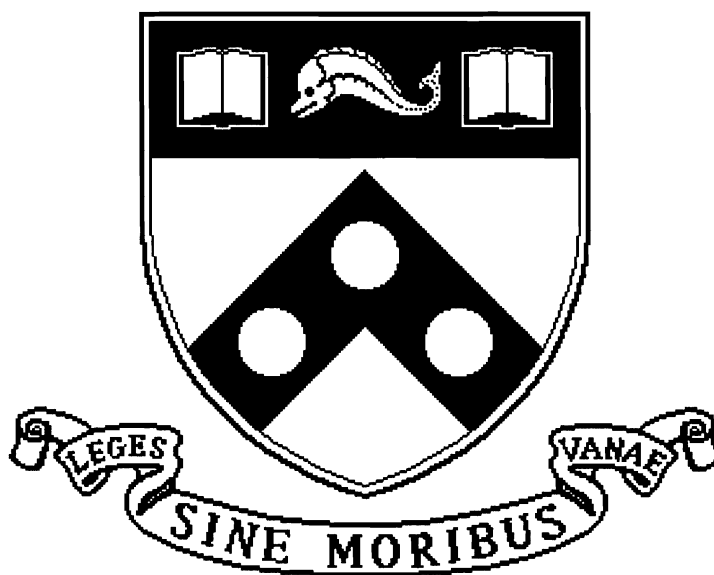
1995

Towards Personalities for Animated Agents with Reactive and Planning Behaviors

Norman I. Badler   Barry D. Reich   Bonnie L. Webber
Center for Human Modeling and Simulation
Department of Computer & Information Science
University of Pennsylvania*

December 6, 1995

Abstract

We describe a framework for creating animated simulations of virtual human agents. The framework allows us to capture flexible patterns of activity, reactivity to a changing environment, and certain aspects of an agent personality model. Each leads to variation in how an animated simulation will be realized. As different parts of an activity make different demands on an agent's resources and decision-making, our framework allows special-purpose reasoners and planners to be associated with only those phases of an activity where they are needed. Personality is reflected in locomotion choices which are guided by an agent model that interacts with the other components of the framework.

1   Introduction

Conventional animations often seek to re-create "life" through the artistic skills of an animator who, by drawing and painting, externalizes observations, experience, and intuition into the images, shapes, and movements that make for believable characters [Thom81]. For three-dimensional computer animation, one has a more complex toolkit and a commensurately harder

---

1

task defining exactly what makes an *object* into a *character*. Part of the problem is the control of multiple degrees of freedom: not only must the character do whatever is desired, it must also convey a feeling of liveliness, animacy, and engagement with the environment. Being able to simply walk from here to there is not enough; there should be purpose, reactivity, and attitude. We explore some of these issues in a discussion of a synthetic human agent architecture and a specific implementation in a system called *Jack*® [Badl93b].

Three primary mechanisms exist for specifying motion in a synthetic computer-generated character:

1. Direct manipulation of the body parts to the poses desired;

2. Actually perform the actions, so that the virtual agent mimics the participant's actual performance;

3. Instructing an agent in what to do, so that its behavior follows, in part, from the goals it has adopted.

Recent automated techniques for animation aim to ease the animator's burden, but it appears that *personality* is typically established through skillful direct manipulations. As such, it is *ad hoc* and probably difficult to quantify and reproduce. Incorporating mathematical techniques for motion interpolation, and even using physics-based models, the agents tend to look personality-free at best and hapless mechanical mannequins at worst.

Producing animated people seems to require more than the existing physical or manual toolset. One response to this difficulty is *performance animation*: live actors perform while sensing systems monitor various body parts such as hands [Burd94], faces [Will90], or landmarks [Badl93a, Robe94]. While this provides motion data of unquestioned realism, it is only one instance of a performance: different conditions can lead to different motions because people adapt their behavior to circumstances through their unconscious reactions and conscious decision-making. Moreover, unless the performer is additionally a good actor, the *personality* of the performer is captured as well. This bias sometimes may be exactly what was desired, but at other times it is an unwelcome feature that must be suppressed by manual post-processing [Brud95]. This may be possible for an agent's *physical* style, but behavioral characteristics that follow from their *decision-making* (or *cognitive*) style are another story.

We start by briefly reviewing our historical approaches to personality for animated humans. Then we will present a two-level architecture for

2

intelligent agents, including computational approaches to patterns of human activity and how PaT-Nets capture such patterns for the production of animated simulations. Finally we will describe how agent personality variations can affect the animated outcomes, both in terms of physical style and decision making/cognitive style.

2 Approaches to Animated Agent Personalities

Above we outlined three modes of controlling animation:

1. Manipulating the human agent directly to position and move it as desired;

2. Showing the human agent what to do, so it moves by imitation;

3. Telling the human agent what to do, so that its behavior follows, in part, from the goals it has adopted.

We believe future *real-time* animations must converge to techniques (2) and (3): basically "do what I do" and "do what I tell you to do". Performance-based systems for virtual environments are forced to adopt technique (2). The economy of expression in verbalized commands requires considerable ongoing study of technique (3). But the techniques developed for direct manipulation (1) will emerge as the facilitator for the other two. Thus we consider the study of algorithmic techniques for human movement control to be an important endeavor for animating personalities for real-time agents.

2.1 Physical Style

About twenty years ago we began studying human movement notations in order to gain insight into what qualities and properties human movement observers and choreographics abstracted from the continuum of motion. Briefly, our first "lesson learned" was that the formal (and even computationally tractable) structures of Labanotation [Hutc70, Webe78] captured the *form* of movement but not its *essence* — that is, its character, style, or emotion. We learned that even Laban was aware of its limitations, and towards the end of his career developed another system called "Effort-Shape" [Dell70]. This system sought to describe the "qualities" or shape of a movement without trying to bind it to specific spatial locations or orientations.

3

In more physical terms, we saw Effort-Shape as seeking to characterize the first and second derivatives of position, and we began to explore some of the possible relationships between Effort-Shape notation and computer animation techniques [Badl89]. We felt that if we could characterize Effort-Shape qualities in terms of computational realizations, we could generate characters with various personalities or emotional content. While there were tantalizing beginnings to this process, we were stymied by other problems (such as modeling a decent human figure in the first place) and so directed our attentions elsewhere. Only much later did some of this work re-surface under the guise of locomotion "styles" [Ko94]: when the *Jack* figure was animated walking with bent torso and smooth motion, the walk conveyed a distinct pensive or sad look; Walking with upright torso, raised shoulders, and an abrupt gait conveyed a "macho" or aggressive look. We have not researched this specific connection more thoroughly, but the underlying motion implementations are now sufficiently developed to make the prospect both interesting and feasible.

2.2   Cognitive Style

Most animation tools provide manual control over images, shapes, and movements. Recently, more automated techniques for animation have been developed, often to ease some burden or other on the animator. For example, dynamics can be used to animate particles or objects responding to physical forces [Hahn88, Wilh90], and "flocking" can be used to constrain interactions between figures in a scene [Reyn87, Tu84]. Partial success can be judged from the various physics-based techniques that use "real-world" mathematics to get the motions "right" [Cohe92, Sims94, Witk88].

Unfortunately, getting animated figures to appear human-like seems to require more than the existing physical or manual toolset. One reaction to this difficulty has been the "performance animation" noted earlier, where actors go through the necessary motions while sensing systems monitor various body landmarks. While this provides motion data of unquestioned realism, it is only a specific instance of a performance and might still need massaging by an expert. For example, it may not be directly usable for a character of markedly different body size/shape than the original actor.

Moreover, while performance animation generally guarantees that the physics is correct — without building and evaluating the formulas – it still misses something. Consider the following scenario:

A pedestrian stands on a street corner, waiting for the light

4

to change so that he can safely cross the street. Meanwhile a car is approaching the same intersection. What happens when the light changes?

In this scenario, the performance-based data might be useful for animating the character's walk, though it could also be simulated through a locomotion generator [Ko94]. In a scripted animation, the animator would be responsible for initiating the walk when the light changed and would also be controlling the car motions. Suppose we then removed the animator: A pedestrian completely driven by physics would be propelled across the street by his forward force. The car would also be moved by physics. If they arrived at the same place at the same time, the animation might be exciting but it would not be fun for either the car or the pedestrian. So what happens when we remove the animator is that we remove the pedestrian's decisions: *human movement realism reflects decision-making in context*. For realistic animation, *synthetic humans must engage in such decision-making if we want them to share human qualities*. Sensed human motions are insufficient because they reflect only decisions that have already been made. Physics is insufficient because there are no decisions outside the outcome of the mathematical laws. Conscious humans are neither puppets nor mannequins. They continually assess their surroundings (to validate expectations, avoid obstacles, minimize surprises *etc.* [Webb95]), make choices, and plan for the future.

Different cognitive "styles" of sensing, decision-making and planning (including, for example, a agent's degree of commitment to his/her current goals [Brat88]) can make animated human agents behave differently in response to the same environment and symbolically-specified activity. Below we will describe several personality variants that are apropos to our project of having a set of animated humans play games of Hide and Seek [Badl95]. These include styles of seeking a hiding place, styles of responding to inferred intentions of other agents, and styles of pursuit.

3   The Agent Architecture

Allowing behavior to follow from decisions made in context appears to be a prime motivator for human-like agents. In this section we outline our agent architecture as a two-level structure. The lower level functions as a Sense-Control-Act (SCA) loop, while a higher level executes a pre-defined (but general) schema.

5

An *agent* can take action by virtue of having an SCA loop to produce locally-adaptive (reactive) behavior. In general, behaviors are considered "low level" capabilities of an agent, such as being able to locomote [to], reach [for], look [at], *etc.* In this discussion, we shall concentrate primarily on the walking behavior, as it is clearly influenced by the local structure of the environment, the presence of sensed obstacles, distance to the goal, *etc.*

An agent also manifests high-level patterns of activity and deliberation, which in turn can affect the immediate formulation and parameters of an SCA loop. Such patterns are captured in our framework through parallel state-machines we call *Parallel Transition Networks*, or PaT-Nets. PaT-Nets can sequence actions based on the current state of the environment, of the goal, or of the system itself, and represent the tasks in progress, conditions to be monitored, resources used, and temporal synchronization. An agent's deliberations both prior to and during action can be captured through special purpose reasoners and planners associated with specific states of a network.

In this framework, the agent can instantiate PaT-Nets to accomplish goals (*e.g.*, go to the supply depot and pick up a new motor), while low-level control is mediated through direct sensing and action couplings in the SCA loop (*e.g.*, controlling where the agent's feet step and making sure that s/he doesn't run into or trip over any obstacles). By linking numerical feedback streams (SCA loops) and state controllers (PaT-Nets supported by special-purpose reasoners and planners), we believe it is possible to obtain maximum flexibility and maintain appropriate levels of specification in the animated simulation of virtual human agents [Badl93b, Beck93].

The rest of this section briefly describes features of SCA loops and PaT-Nets. For more detail, see [Badl95].

3.1 Low-Level Control: SCA Loops

The *behavioral loop* is a continuous stream of floating point numbers from the simulated environment. Simulated sensors map these data to the abstract results of perception and route them through control processes, each of which is independently attempting to solve a minimization problem. The results go to simulated effectors or motor actions that enact changes on the agent or the world. This loop operates continuously.

The behavioral loop is modeled as a network of interacting *sense, control,* and *action* (SCA) processes, connected by arcs across which only floating point messages travel. An individual path from sensors to effectors is referred to as a *behavioral net*. It is analogous to a complete behavior in

6

an "emergent behavior" architecture such as Brooks' *subsumption architecture* [Broo86], except that nodes may be shared between behaviors, and arbitration (competition for effector resources) may occur throughout the behavioral path and not just at the end-effector level. The behavioral loop is modeled as a network with floating point connections in order to allow the application of low-level, unsupervised, reinforcement learning in the behavioral design process [Beck95]. Here we briefly describe the components of an SCA loop.

Sensory Nodes Sensory nodes model or approximate the abstract, geometric results of object perception. They continuously generate signals describing the polar coordinate position (relative to the agent) of a particular object or of all objects of a certain type within a specified distance and field of view.

Control Nodes Control nodes model the lowest level influences on behavior. For example, control of locomotion is loosely based on Braitenberg's *love* and *hate* behaviors [Brai84], here called *attract* and *avoid*. Control nodes are formulated as explicit minimizations using outputs to drive inputs to a desired value (similar to Wilhelms' [Wilh90] use of Braitenberg's behaviors). They typically receive input signals directly from sensory nodes, and send outputs directly to action nodes, though they could be used in more abstract control situations.

Action Nodes Action nodes connect to the underlying human body model and directly execute routines defined on the model (such as walking, balance, hand position, and torso orientation) and arbitrate among inputs, either by selecting one set of incoming signals or averaging all incoming signals. An example is the *walk controller*, which decides where to place the agent's next footstep and then connects to the locomotion generator [Badl93b, Moor95] to achieve the step.

Our main use of SCA loops to date has been in locomotion reasoning.

3.2 High-Level Control: PaT-Nets

PaT-Nets are finite state machines with message passing and semaphore capabilities [Beck94, Douv95]. Nodes are associated with processes that can invoke executable behaviors, other PaT-Nets, or specialized reasoners or planners. Invocation occurs when a node is entered. An arc transition

between nodes in a PaT-Net may check a local condition evaluated within the PaT-Net or a global condition evaluated in an external environment. Arcs are prioritized, and a transition is made to a new node by selecting the first arc with a true condition. Nodes may also support probabilistic transitions, reflecting the probability of transitioning to another node at a given clock tick. *Monitors* associated with a PaT-Net will execute an action if a general condition evaluates to true, regardless of the current state. In addition, a PaT-Net may have local state variables available to all processes and conditions, and may also take parameters on instantiation.

A running network is created by making an instance of the PaT-Net class. All running PaT-Net instances are embedded in a Lisp operating system that time-slices them into the overall simulation. While running, PaT-Nets can spawn new nets, communicate with each other, kill other nets, and/or wait (sleep) until a condition is met. Running nets can, for example, spawn new nets and then wait for them to exit (effectively a subroutine call), or run in parallel with the new net, while maintaining communication with it. Because PaT-Nets are embedded in an object-oriented structure, new nets can be defined that override, blend, or extend the functionality of existing nets.

4 Patterns of Activity

It has long been recognized that much of everyday human activity falls into patterns. In early work on computer-based story understanding, such patterns were captured in structures called "scripts" or "schemata" and were used to fill in what hadn't been said in a story as well as what had been said [Scha77]. Scripts/schemata have also be used in generating behavior. For example, in "hierarchical planning" [Sace77, Wilk88], a plan operator specifies a partially ordered pattern of actions or sub-goals that can be used to achieve a particular goal.

Here we want to make three points about patterns of activity:

1. Patterns have different sources: they may be idiosyncratic, peculiar to an individual; they may be cultural, simplifying interactions between people [Cass94]; they may be occupational, as in a medic's initial assessment of a trauma patient [Chi95]; or they may be recreational, as in the pattern of a game.

2. Patterns vary in their rigidity. They range from low-level fixed pat-

8

terns one may do daily without thinking – for example, putting one's left shoe on first and then one's right – to high-level patterns of engagement where the particulars are subject to variation such as in having breakfast, making dinner, going out for dinner, playing golf, *etc*. The variation may reflect low-level reactions to circumstances, high-level decisions, or current state.

3. People may be engaged simultaneously in multiple patterns of activity. Their resulting behavior may therefore reflect their allocating different resources to different patterns, using one pattern in support of another, time-slicing patterns or blending patterns.

The next section shows how PaT-Nets can be used to capture patterns of activity in the game of Hide and Seek.

5   An Example of PaT-Net Agent Control

While there are many different sets of rules for Hide and Seek, most involve one player (who is "it" or the "seeker") first averting his/her eyes while the other players hide, then setting out to find at least one of them, then engaging in some competition with whomever is found, which may then lead to that other player becoming "it" in the next round. Thus Hide and Seek manifests a pattern of behavior whose realization may vary from instance to instance as players hide in different places, as different players are found, and as the competition between the player who is "it" and the player who is found yields different results. Here we show how this kind of pattern is easily supported in PaT-Nets.

The high-level controller, or *PlayNet*, for simulating one common version of Hide and Seek is illustrated in Figure 1. In this version, a hider, once hidden, does not move until seen by the seeker, at which time the hider attempts to run home without being tagged by the seeker. In Figure 1, the **Sync** node causes players to wait until all the players are "home", at which point the seeker may begin counting (**Count**) and the other players start to hide (**Hide**). The **Evade** node has a player running to home base, while avoiding the seeker. (Currently this sort of multi-goal behavior is hand-coded while a more general approach to multiple goal integration is worked out.) Dashed transitions in the network represent a change in role from hider to seeker or *vice-versa*. The **at home** condition becomes true when

9

the agent is at home-base, while the **safe** condition becomes true when the seeker notices that the hider he is pursuing has made it home.
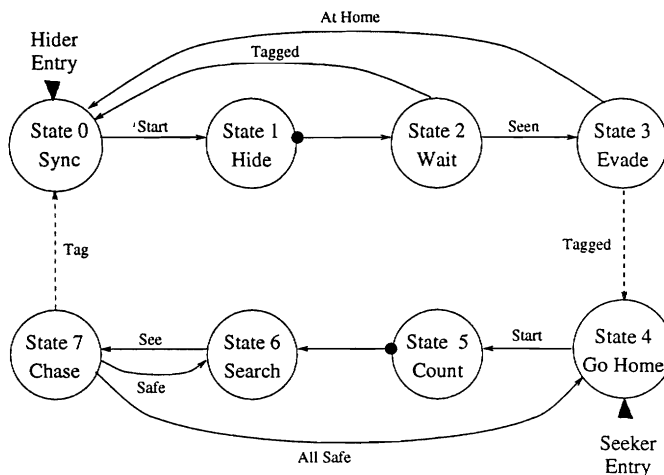

Figure 1: PlayNet for Hide and Seek

While a *PlayNet* specifies an entire game of Hide and Seek, each player a personal copy, which records features of the game specific to that player (*e.g.*, whether the player has been seen by the seeker or made it home safely), as well as features of the global state of the game (*e.g.*, whether all hiders have made it home safely).

To support agent coordination, two types of communicative mechanisms have been used in Hide and Seek. First, the environment itself is used: Players have limited (simulated) perceptual abilities which allow them to see unobstructed nearby things in their environment. This supports the coordination involved in **see/seen** in Figure 1. Second, coordination of all agents is handled through semaphores. For example, in Figure 1, count **done** is broadcast to all agents, as are **tag/tagged** and **at home/safe**, since such information is relevant to all players but cannot be insured known through the players' limited perception.

Focusing now on the processes associated with individual nodes of a *PlayNet*, these may either be intrinsically terminating, as in **count** (count to 10), indicated by a black dot on the arc, or externally terminating by virtue of some event in the environment. For example, **search** is only terminated when circumstances provide the seeker with a hider. Note that intrinsically

10

terminating processes may also be interrupted by external events, stopping them before completion. For example, **evade** (*i.e.*, run home while evading the seeker) is intrinsically terminated by the hider reaching home, but may be interrupted by the hider being tagged. For PaT-Nets to support this, actions must be cleanly preemptable, so that the simulation programmer can pass in a condition for premature termination.

The use of special purpose reasoners and planners in the context of specific processes is illustrated in two of the nodes of Figure 1 – **hide** and **search**. Currently during **hide**, a special purpose reasoner is invoked when the node is entered, that evaluates the set of hiding places the agent knows about and chooses one that maximizes the agent's preference criteria. Each agent has different preferences regarding a hiding place – differing by the importance of its distance from the seeker, the number of available exits, *etc.* Once a hiding place is chosen, the agent moves towards it. One can imagine, though, a more sophisticated reasoner for choosing hiding places that would be able to work in previously "unknown" environments and that would provide more room for the expression of cognitive style. Such a reasoner might carry out noticing and evaluating hiding places in parallel, while moving away from "home". Since all hiders are simultaneously looking for hiding places, agent personality can be expressed in their response to contention for a hiding place. An non-combative agent might acquiesce and seek another spot as soon as it "realizes" there may be contention for its current chosen target. A confrontational agent might respond quite differently.

Another special purpose reasoner is illustrated in the **search** node. Searching is viewed as choosing a place to look for a hider, and then doing what one needs to get there ([Geib95, Moor93]). Choosing a place to look involves keeping track of where one has looked already and reasoning about what remaining spaces can contain an agent that the seeker cannot see. Unlike hiding currently, seeking does not assume the seeker knows about all possible hiding places *a priori*. More effort has been put into a model of realistic seeking than hiding.

6 Behavioral Control of Human Locomotion Based on Personality

The aim is to design and build a system in which an animator has control over the agent model. This includes personality traits such as curiosity

and cautiousness, and state information such as the agent's energy level and alertness. The animator sets high-level locomotion goals for the agent. The system generates locomotion automatically based on the agent model, shaping the locomotion to reflect and convey to an observer the agent state and personality chosen by the animator.

As an example, consider an animator who is given the task of modeling several agents playing hide and seek in an outdoor environment. We will look at two in particular: Marge and Homer. In the scene to be animated, Homer is in his hiding place and Marge is seeking. The animator opens the agent model interface window and configures the agents beginning with Homer. She clicks on the "Speed" button with the mouse and a slider-bar appears. She chooses a relatively fast speed of 9.0 (out of 10.0). Then she sets Homer's "Awareness" of his surroundings to 9.5.

The animator configures Marge in a similar way, also setting her "Awareness" to 9.0, but setting her "Speed" to 6.0. She wants to find a hider quickly, but not to move so quickly that she misses someone. Finally, that animator wants to allow Marge to notice and react to hiders as they become visible. She defines a LISP condition which evaluates to TRUE when Marge can see a hider. The probability of reacting to this condition when it becomes TRUE She relates to Marge's "Awareness" level. If this happens, a new schema is adopted with the goal "go to hider", followed by schemata that cause Marge to tag the hider and bring everyone home, ending the game. When the animator tests the configurations by starting the locomotion control systems, she sees the following.

> Marge begins exploring the environment. She is walking somewhat quickly, entering and exiting small structures and avoiding obstacles. As she walks from one building to another, she notices Homer hiding in the alley between them. She stops, turns toward Homer, and begins chasing him. Homer notices Marge immediately, turns away from her, and runs away.

The animator shows the animation to a colleague. After a few minutes of discussion and a small amount of parameter adjustment, she is satisfied that the agents appear to have the desired personalities. What follows is a description of our multi-level locomotion control system that supports this embodiment of a character's personality traits in their locomotion choices and characteristics.

This system controls locomotion at three levels. At the lowest level, a behavioral simulation system controls agent locomotion directly based on a

behavioral loop associated with the agent (Sec. 3.1). This level makes final choices about where the agent will place his feet at each step. The second level consists of PaT-Nets (Sec. 3.2) that control agent locomotion indirectly by scheduling and controlling reactive behaviors. Above this is an "agent model" to configure the state-machine and parameterize the behaviors so that the agent's locomotion style reflects its personality, and physical and mental state.

The behavioral simulation system [Beck93] communicates directly with a human locomotion system [Ko94] choosing each footstep for the agent based on a set of behaviors associated with that agent. The behaviors are variations of attraction and avoidance and are completely parameterizable. The agent's observed behavior is the result of the choice of behaviors and parameters which are themselves the results of choices made at higher levels.

There are limits to what low-level reactive behaviors can achieve. When the desired complexity of an agent's behavior increases beyond this limit, higher-level locomotion reasoning must be introduced. Locomotion reasoning determines the characteristics of an agent's locomotion: *i.e.*, what types of attractions and avoidances with what choice of parameters will achieve the goal. This is done via PaT-Nets.

PaT-Nets and their special purpose reasoners support complex behaviors such as *chasing* and *path-following* where behaviors and parameters change over time (Sec. 5). In addition to changing behaviors and parameters, the state-machine structure allows us to evaluate, in parallel, arbitrary conditions, passed as arguments. If any of these conditions ever evaluate to TRUE, the state-machine takes an appropriate action. These conditions allow the agent to interrupt the current plan when an unexpected situation occurs, *e.g.*, when something of interest enters the agent's field-of-view. An appropriate message is passed up to the calling system allowing it to replan.

The agent model, a set of agent characteristics or attributes, is a global structure of attributes that can affect the configuration of the state-machine which in turn affects the agent's perceived behavior.

Figure 2 shows a subset of the characteristics that make up the agent model and some of the ways each one affects the agent's behavior. **Speed**, or walking rate, is one of the factors regulated by the "rushed" and "fatigued" slider bars. A fatigued or laden agent walks slowly while an agent who is in a rush walks quickly or runs. An inebriated agent walks at an inconsistent rate, with a velocity that varies in an amount proportional to the level of intoxication.

The combination of several behaviors, as is necessary with behavioral

Characteristic	Locomotion System Parameters Affected				
	Speed	Inertia	Anticipation	Condition	Probability
Fatigued	X				
Laden	X				
Rushed	X .				X
Inebriated	X	X	X		X
Aware			X		X
Alert to Danger				X	
Curious				X	X
Distracted					X

Figure 2: How the agent model affects the agent

control, typically yields excessively wandering paths. In an attempt to straighten locomotion paths, we add an "inertia" behavior to an agent's behavior set which attracts the agent to the forward direction. An inebriated agent is given a low **Inertia** value and tends to meander as a result.

When we walk around in the presence of other people we attempt to avoid them based not only on their current location, but also on a guess of where they will be one or more steps in the future. Consciously or unconsciously, we anticipate their potential locations in the near future and base our avoidance on this prediction. An inebriated agent has a lower **Anticipation** value than normal, while an agent who is particularly aware of his surroundings will anticipate more.

Condition is a special system feature implemented at the state-machine level. It is a *hook* which allows the animator to implement behaviors, particularly those involving sub-goals, and make decisions that are unsupported by the architecture. When an arbitrary condition evaluates to TRUE, the state-machine takes an appropriate action, possibly stopping the agent and exiting. This allows a higher-level system to replan, taking advantage of the opportunity. A condition, for example, might evaluate to TRUE when an enemy is in the agent's field-of-view. Consider this scenario. We see a burglar walk past a police officer. The burglar notices the officer and starts running (his sense of urgency increases). He tries to find a good hiding place while the officer pursues him. Our system is capable of simulating this scenario.

Although the burglar passes the officer, in real life, there is no guarantee

14

that they would notice each other. **Probability** refers to the probability that an agent will react to the condition evaluating to TRUE. The probability is high when the agent is aware of his surroundings or curious, but low when he is distracted, inebriated, or rushed, for example.

7 Conclusion

We have just begun to explore the power of this agent architecture. Many questions have been identified for further study. They include:

- the viability of confining high-level reasoning and planning to particular nodes of a PaT-Net. It may be that in other activities, reasoning and planning need a more global purview or need to be interleaved in as yet unforeseen ways.

- the range of activities worth framing in a net-based approach: the greater the number of nodes, the greater the connectivity between them, and the greater the complexity of the arc conditions, the less a net-based approach seems to make sense. In other words, we would like to identify criteria we can use to decide when to employ a first-principles planner and when we can use a network of pre-determined decision points. It is possible that learning a task or becoming more expert at something entails a migration from planning as a node to more focused networks of choices.

- a clear regimen for mapping instructions for a task or game (including warnings and constraints) to one or more communicating PaT-Nets.

- the mechanism for incorporating other personality traits, especially non-locomotor ones. Preliminary investigations into two-person synthetic communication has already yielded insights into personality influences on speech, dialogue, turn-taking, and gesture [Cass94].

We have noted here that *synthetic humans must also be able to adapt to circumstances if we want them to share human qualities.* Motion performance is not sufficient because we do not know yet how to readily adapt it to other circumstances. Even when behavior essentially follows a pattern, as in going to a supermarket, playing a game, or trouble-shooting and/or repairing equipment, people sense the world around them, react and make choices in ways that adapt the pattern to their personalities and circumstances. It is supporting flexible patterns of activity that motivates much

15

of our research. We see animation as an integration of a rich collection of interacting techniques, organized in a principled, structured representation. Here we have discussed its use in representing flexible patterns of human activity influenced by personality.

Our experiments so far indicate that the representational efficacy of this architecture of behavioral reaction, transition networks, symbolic planning, and certain personality attributes is necessary for modeling actions of human-like agents. Whether it is sufficient is a question for the future.

References

[Badl89] Badler, N. A representation for natural human movement. In J. Gray (ed.), *Dance Technology I*. AAHPERD Publications, Reston, VA (1989), 23–44.

[Badl93a] Badler, N. I., Hollick, M. J. and Granieri, J. Real-time control of a virtual human using minimal sensors. *Presence* **2**(1) (1993), 82–86.

[Badl93b] Badler, N. I., Phillips, C. W. and Webber, B. L. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, New York, NY (1993).

[Badl95] Badler, N. I., Webber, B. L., Becket, W., Geib, C., Moore, M., Pelachaud, C., Reich, B. and Stone, M. Planning for animation. To appear in *Computer Animation*, ed. N. Magnenat-Thalmann and D. Thalmann, Prentice-Hall (1995).

[Badl91] Badler, N. I., Webber, B. L., Kalita, J. and Esakov, J. Animation from instructions. In N. Badler, B. Barsky, & D. Zeltzer (eds.), *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*. Morgan-Kaufmann, San Mateo, CA (1991) 51–93.

[Beck94] Becket, W. The *Jack* Lisp API. Technical Report MS-CIS-94-01, University of Pennsylvania, Philadelphia, PA (1994).

[Beck95] Becket, W. *Reinforcement Learning for Reactive Navigation of Simulated Autonomous Bipeds*. PhD thesis, University of Pennsylvania (1995).

[Beck93] Becket, W. M. and Badler, N.I. Integrated behavioral agent architecture. In *The Third Conference on Computer Generated Forces and Behavior Representation*, Orlando, FL (1993).

[Brai84] Braitenberg, V. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, Cambridge, MA (1984).

[Brat88] Bratman, M., Israel, D. and Pollack, M. Plans and resource-bounded practical reasoning. *Computational Intelligence* 4(4) (1988), 349–355.

[Broo86] Brooks, R. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* (1986), 14–23.

[Brud95] Bruderlin, A. and Williams, L. Motion signal processing. *Computer Graphics*, Annual Conference Series, ACM (1995), 97–104.

[Burd94] Burdea, G. and Coiffet, P. *Virtual Reality Technology.* Wiley, NY (1994).

[Cass94] Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, W., Douville, B., Prevost, S. and Stone, M. Animated Conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. *Computer Graphics*, Annual Conference Series, ACM (1994), 413–420.

[Chi95] Chi, D., Webber, B., Clarke, J. and Badler, N. Casualty modeling for real-time medical training. To appear, *Presence* (1996), Special issue on human modelling.

[Cohe92] Cohen, M. F. Interactive spacetime control for animation. *Computer Graphics* 26(2) (1992), 293–302.

[Dell70] Dell, C. *A Primer for Movement Description.* Dance Notation Bureau, New York, NY (1970).

[Douv95] Douville, B. PaT-Net User's Guide. Technical Report. Department of Computer & Information Science, University of Pennsylvania (1995).

[Geib95] Geib, C. *The Intentional Planning System: ItPlanS.* PhD thesis, Dept of Computer & Information Science, University of Pennsylvania (1995).

[Hahn88] Hahn, J. K. Realistic animation of rigid bodies. *Computer Graphics* 22(4) (1988), 299–308.

[Hutc70] Hutchinson, A. *Labanotation*. Theatre Arts Books, New York, NY (1970).

[Ko94] Ko, H. *Kinematic and Dynamic Techniques for Analyzing, Predicting, and Animating Human Locomotion*. PhD Dissertation, Department of Computer & Information Science, University of Pennsylvania (1994).

[Moor93] Moore, M. B. Search Plans. (PhD Dissertation Proposal) Technical Report MS-CIS-93-56/LINC LAB 250/IRCS-93-29. Department of Computer & Information Science, University of Pennsylvania (1993).

[Moor95] Moore, M.B., Geib, C.W. and Reich, B.D. Planning and terrain reasoning. *AAAI Spring Symposium on Integrated Planning Applications*, Stanford CA (March 1995). (Also appears as Technical Report MS-CIS-94-63, University of Pennsylvania).

[Reic94] Reich, B.D., Ko, H., Becket, W. and Badler, N. Terrain reasoning for human locomotion. *Proceedings of Computer Animation '94*, Geneva, IEEE Computer Society Press (1994), 996–1005.

[Rena90] Renault, O., Magnenat-Thalmann, N. and Thalmann, D. A vision-based approach to behavioral animation. *The Journal of Visualization and Computer Animation* 1(1) (1990), 18–21.

[Reyn87] Reynolds, C.W. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21(4) (1987), 25–34.

[Reyn88] Reynolds, C.W. Not bumping into things. *SIGGRAPH Course 27 Notes: Developments in Physically-Based Modeling*, ACM SIGGRAPH (1988), G1–G13.

[Robe94] Robertson, B. Caught in the act. *Computer Graphics World* 17(9) (1994), 23–28.

[Sace77] Sacerdoti, E. *A Structure for Plans and Behavior*. American Elsevier, New York, NY (1977).

[Scha77] Schank, R. and Abelson, R. *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ (1977).

[Sims94] Sims, K. Evolving virtual creatures. *Computer Graphics*, Annual Conference Series, ACM (1994), 15–22.

[Thom81] Thomas, F. and Johnson, O. *Disney Animation: The Illusion of Life*. Abbeville Press, New York, NY (1981).

[Tu84] Tu, X. and Terzopoulos, D. Artificial fishes: Physics, locomotion, perception, and behavior. *Computer Graphics*, Annual Conference Series, ACM (1994), 43–50.

[Webb95] Webber, B., Badler, N., Di Eugenio, B., Geib, C., Levison, L. and Moore, M. Instructions, intentions and expectations. *Artificial Intelligence Journal* **73** (1995), 253–269.

[Webe78] Weber, L., Smoliar, S. W. and Badler, N. I. An architecture for the simulation of human movement. Proc. ACM Annual Conf., Washington, DC (1978), 737–745.

[Wilh90] Wilhelms, J. and Skinner, R. A 'notion' for interactive behavioral animation control. *IEEE Computer Graphics and Applications* **10(3)** (1990), 14–22.

[Wilk88] Wilkins, D.E. *Practical Planning*. Morgan Kaufmann, San Mateo, CA (1988).

[Will90] Williams, L. Performance-driven animation. *Computer Graphics*, 24(4) (1990), 235–242.

[Witk88] Witkin, A. and Kass, M. Spacetime constraints. *Computer Graphics* **22(4)** (1988), 159–168.