Real-Time Virtual Humans

Norman I. Badler
Center for Human Modeling and Simulation
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389
badler@central.cis.upenn.edu
1-215-898-5862 phone; 1-215-573-7453 fax

Abstract

The last few years have seen great maturation in the computation speed and control methods needed to portray 3D virtual humans suitable for real interactive applications. We first describe the state of the art, then focus on the particular approach taken at the University of Pennsylvania with the Jack *system. Various aspects of real-time virtual humans are considered, such as appearance and motion, interactive control, autonomous action, gesture, attention, locomotion, and multiple individuals. The underlying architecture consists of a sense-control-act structure that permits reactive behaviors to be locally adaptive to the environment, and a* **PaT-Net** *parallel finite-state machine controller that can be used to drive virtual humans through complex tasks. We then argue for a deep connection between language and animation and describe current efforts in linking them through two systems: the* Jack Presenter *and the* JackMOO *extension to lambdaMOO. Finally, we outline a* Parameterized Action Representation *for mediating between language instructions and animated actions.*

Keywords and Phrases: Virtual humans, human modeling, computer animation, virtual reality, autonomous agents, language and action, computer graphics.

1. Virtual Humans

Only fifty years ago, computers were barely able to compute useful mathematical functions. Twenty-five years ago, enthusiastic computer researchers were predicting that all sorts of human tasks from game-playing to automatic robots that travel and communicate with us would be in our future. Today's truth lies somewhere in-between. We have balanced our expectations of complete machine autonomy with a more rational view that machines should assist people to accomplish meaningful, difficult, and often enormously complex tasks. When those tasks involve human interaction with the physical world, computational representations of the human body can be used to escape the constraints of presence, safety, and even physicality.

Virtual humans are computer models of people that can be used

- as substitutes for "the real thing" in *ergonomic* evaluations of computer-based designs for vehicles, work areas, machine tools, assembly lines, etc., *prior to the actual construction of those spaces*;
- for *embedding real-time representations of ourselves or other live participants* into virtual environments.

Recent improvements in computation speed and control methods have allowed the portrayal of 3D humans suitable for interactive and real-time applications. There are many reasons to design specialized human models that individually optimize character, performance, intelligence, and so on. Many research and development efforts concentrate on one or two of these criteria.

In the efforts that we describe here, we cross several domains which in turn build from various interrelated facets of human beings (Fig. 1):

- Human Factors Analysis: Human size, capabilities, behavior, and performance affects work in and use of designed environments.
- Real-Time Agents and Avatars: People come from different cultures and have different personalities; this richness and diversity must be reflected in virtual humans since it influences appearance as well as reaction and choice.
- Instruction Understanding and Generation: Humans communicate with one another within a rich context

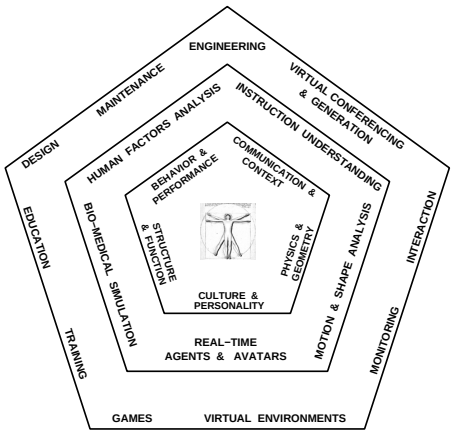

Figure 1. Virtual human applications, technology, and science.

of shared language, senses, and experience and this needs to be extended to computer-generated agents and avatars.

- Bio-Medical Simulation: The human machine is a complex of physical structures and functions; to understand human behavior, physiological responses, and injuries we need to represent biological systems.

- Motion and Shape Analysis: Understanding what we perceive when we see or sense the world leads to models of the physical world (physics) and the geometric shapes and deformations of objects.

In building models of virtual humans, there are varying notions of *virtual fidelity*. Understandably, these are application dependent. For example, fidelity to human size, capabilities, and joint and strength limits are essential to some applications such as design evaluation; whereas in games, training, and military simulations, temporal fidelity (real-time behavior) is essential. In our efforts we have attacked both.

Understanding that different applications require different sorts of virtual fidelity leads to the question of what makes a virtual human "right"?

- What do you want to do with it?

- What do you want it to look like?

- What characteristics are important to success of the application?

Unfortunately the state of research in virtual humans is not as advanced as to make the proper selection a matter of buying off-the-shelf systems. There are gradations of fidelity in the models: some models are very advanced in a narrow area but lack other desirable features.

In a very general way, we can characterize the state of virtual human modeling along at least five dimensions:

- **Appearance**:
  Cartoon shape $->$ Physiological model

- **Function**:
  Cartoon actions $->$ Human limitations

- **Time**:
  Off-line generation $->$ Real-time production

- **Autonomy**:
  Direct animation $->$ Intelligent

- **Individuality**:
  Specific person $->$ Varying personalities

The current state of Virtual Human technology is that we (and others) have proceeded rather far beyond the individual off-line rendering of still frames as realized by traditional hand animation or even computer assisted cartoon animation. If we need to invoke them, the *appearance* of increasingly accurate physiologically- and biomechanically-grounded human models may be obtained. We can create virtual humans with *functional* limitations that go beyond cartoons into instantiations of known human factors data. Animated virtual humans can be created in human *time* scales through motion capture or computer synthesis. Virtual humans are also beginning to exhibit the early stages of *autonomy* and intelligence as they react and make decisions in novel, changing environments rather than being forced into fixed movements. Finally, rather preliminary investigations are underway to create characters with *individuality* and personality who react to and interact with other real or virtual people [7, 8, 12, 33, 39, 45].

Virtual humans are different than simplified cartoon and game characters. What are the characteristics of this difference and why are virtual humans more difficult to construct? After all, anyone who goes to the movies can see marvelous synthetic characters (aliens, toys, dinosaurs, etc.), but they have been created typically for one scene or one movie and are not meant to be re-used (except possibly by the animator – and certainly not by the viewer). The difference lies in the *interactivity* and *autonomy* of virtual humans. What makes a virtual human *human* is not just a well-executed exterior design but movements, reactions, and decision-making which appear "natural," appropriate, and contextually-sensitive.

2. Agents and Avatars

We will consider an *agent* to be a virtual human figure representation that is created and controlled by computer

programs. An *avatar* is a virtual human controlled by a live participant. The principal issues roughly follow the dimensions cited above: *appearance and motion*, mechanisms of *control for interactivity and autonomy*, including gesture, attention, and locomotion, and multi-agent *interaction, cooperation, and coordination*.

2.1. Appearance and Motion

Avatars can be portrayed visually as 2D icons, cartoons [30], composited video, 3D shapes, or full 3D bodies [2, 48, 43]. We are mostly interested in portraying human-like motions, so naturally tend toward the more realistic surface and articulation structures. In general, we prefer to design motions for highly articulated models and then reduce both the model detail and the articulatory detail as demanded by the application [23].

Along the appearance dimension, the *Jack*® [3] figure has developed as a polygonal model with rigid segments and joint motions and limits accurate enough for ergonomics evaluations [3]. For real-time avatar purposes, simpler geometry can be used provided that the overall impression is one of a task-relevant figure. Thus a soldier model with 110 polygons is acceptable if drawn small enough and colored and/or texture mapped to be recognized as a soldier. On the other hand, a vehicle occupant model must show accurate and visually continuous joint geometry under typical motions. It must be both an acceptable occupant surrogate as well as a pleasing model for the non-technical viewer – who may be used to going to the movies to see the expensive special effects figures. Our "smooth body" [1] was developed using free-form deformation techniques [46] to aid in the portrayal of visually appealing virtual humans (Fig. 2).

The distinction between "synthesized" motions and the other types is roughly that the former generate transformations for more than one joint at a time. Thus, for example, we store a time series of joint angle changes (per joint) in *channelsets* so that specific motions can be re-played under real-time constraints [23]. No deviation from the pre-stored local transformations are allowed, although the whole body may be re-oriented or the playback speed varied. In a particularly effective modification of this technique, Perlin adds periodic noise to real-time joint transformations to achieve greater movement variability, animacy, and motion transitions [38].

In a motion synthesizer, a small number of parameters control a much greater number of joints, for example:

- end effector position and orientation can control joints along an articulated chain [53, 28, 50],
- a path or footsteps can control leg and foot rotations through a locomotion model [21, 27],

Figure 2. Smooth body *Jack* as virtual occupant in an Apache helicopter CAD model.

- a balance constraint can be superimposed on gross body motions [3, 27],
- dynamics calculations can move joints subject to arbitrary external and internal applied forces [29, 34],
- secondary motions can enhance a simpler form [38, 24].

The relative merits of pre-stored and synthesized motions must be considered when implementing virtual humans. The advantages to pre-stored motions are primarily speed of execution and algorithmic security (by minimizing computation). The major advantages to synthesis are the reduced parameter set size (and hence less information that needs to be acquired or communicated) and the concomitant generalized motion control: walk, reach, look-at, etc. The principal disadvantages to pre-stored motion are their lack of generality (since every joint must be controlled explicitly) and their lack of anthropometric extensibility (since changing joint-to-joint distances will change the computed locations of end effectors such as feet, making external constraints and contacts impossible to maintain). The disadvantages to synthesis are the difficulty of inventing natural-looking motions and the potential for positional disaster if the particular parameter set or code should have no solution, fail

to converge on a solution, or just compute a poor result. In particular, we note that inverse kinematics is not in itself an adequate model of human *motion* – it is just a local positioning aid [3, 28]. The issue of building adequate human motion synthesis models is a wide open and complex research topic.

Since accurate human motion is difficult to synthesize, motion capture is a popular alternative, but one must recognize its limited adaptability and subject specificity. Although a complex motion may be used as performed, say in a CD-ROM game or as the source material for a (non-human) character animation, the motions may be best utilized if segmented into motion "phrases" that can be named, stored, and executed separately, and possibly connected with each other via transitional (non-captured) motions [11, 44]. Several projects have used this technique to interleave "correct" human movements into simulations that control the order of the choices. While 2D game characters have been animated this way for years – using pre-recorded or hand animated sequences for the source material – recently the methods have graduated to 3D whole body controls suitable for 3D game characters, real-time avatars, and military simulations that include individual synthetic soldiers [40, 23, 9].

2.2. Control for Interactivity

Whichever motion generation technique is used, there must be a way of triggering the desired activity in the avatar. Specifying the motion can be as simple as direct sensor tracking (where each joint is driven by a corresponding sensor input), end effector tracking (where inverse kinematics or other behaviors generate the "missing" joint data), or external invocation via menu, speech, or button selection of the actions (whether then synthesized or interpreted from pre-stored data). The interesting observation is that the only mechanism available to an "unencumbered" participant is actually speech! Any other avatar control mechanism requires either a hands-on device (mouse, keyboard, glove input), or else external sensors and a limited field of movement. While there is considerable progress in using computer vision techniques to capture human motion [1, 20, 16, 25], both user mobility and movement generality are still in the future.

Our intention is not to promote speech input *per se*, but to use this observation to promote (in Section 3 a *language-centered* view of action "triggering" augmented and elaborated by parameters modifying lower-level motion synthesis or playback. (For example, this technique is used to great advantage in virtual environment applications such as the immersive interface to **MediSim** [49] and in the responsive characters in **Improv** [38, 39].) Although textual instructions can describe and trigger actions, details need not be explicited communicated. Thus the agent/avatar architecture must include semantic interpretation of instructions and even a lower reactive level within the movement generators that allows motion generality and environmental context-sensitivity.

2.3. Control for Autonomy

Providing a virtual human with human-like reactions and decision-making is more complicated than controlling its joint motions from captured or synthesized data. Here is where we engage the viewer with the character's personality and demonstrate its skill and intelligence in negotiating its environment, situation, and other agents. This level of performance requires significant investment in decision-making tools. We presently use a two level architecture:

- to optimize reactivity to the environment at the lower level (for example, in the choice of footsteps for locomotion through the space) [42, 27, 10];
- to execute parametrized scripts or plan complex task sequences at the higher level (for example, choosing which room to search in order to locate an object or another agent, or outlining the primary steps that must be followed to perform a particular task) [35, 4].

The architecture is built on Parallel Transition Networks **PaT-Nets** [3]. Nodes represent executable processes, edges contain conditions which when true cause transitions to another node (process), and a combination of message passing and global memory provide coordination and synchronization across multiple parallel processes. Elsewhere we have shown how this architecture can be applied to the game of "Hide and Seek" [4], two person animated conversation ("Gesture Jack") [12], and simulated emergency medical care (**MediSim**) [13]. Currently we are using this architecture to construct appropriate gestural responses from a synthetic agent, create appropriate visual attention during high-level task execution, manage locomotion tasks, and study multi-agent activity scheduling.

2.4. Gesture Control

Human arms serve (at least) two separate functions: they permit an agent/avatar to change the local environment through dextrous activities by reaching for and grasping (getting control over) objects [22, 18], and they serve social interaction functions by augmenting the speech channel with communicative emblems, gestures and beats [12].

For the first function, a consequence of human dexterity and experience is that we are rarely told how to approach and grasp an object. Rather than have our virtual humans learn – through direct experience and errors – how to grasp an object, we provide assistance through an object-specific

relational table (OSR). Developed from ideas about *object-specific reasoning* [31], the OSR has fields for each graspable site (in the *Jack* sense of an oriented coordinate triple) describing the appropriate handshape, grasp approach direction, and most importantly, its function or purpose. The OSR is manually created for graspable objects and allows an agent to look up an appropriate grasp site given a purpose, use the approach vector as guidance for the inverse kinematics directives that move the arm, and know which handshape is likely to result in reasonable finger placement. The hand itself is closed on the object through local geometry information and collision detection.

The second function of gestures is non-verbal communication. Thus gestures can be metaphors for actual objects, give indicators (via pointing) of location or participants in a virtual space around the speaker, or augment the speech signal with beats for added emphasis [12]. Currently we are working on embedding culture-specific and even individual personality gesture variations. The potential interference between practical and gestural functions is leading to a resource-based priority model to resolve conflicts.

Given that arm control for avatars requires fast position and orientation of the hands for either reaching or gestural function, fast computation of arm joint angles is essential. In recent work we have pushed beyond iterative inverse kinematics [53] to analytic formulas that can easily keep up with a live performance or a motion synthesizer outputting end effector position and orientation streams [50]. By extending this idea to the whole body, multiple individuals (3-10 on an SGI RE2) may be controlled in real-time by arbitrary end-effector and global body data alone [54].

2.5. Attention Control

A particularly promising connection is underway to connect **PaT-Nets** into other high level "AI-like" planning tools for improved cognitive performance of virtual humans. By interfacing *Jack* to OMAR (Operator Model Architecture) [17], we have shown how an autonomous agent can be controlled by a high level task modeler, and how some important human motor behaviors can be generated automatically from the action requests. As tasks are generated for the *Jack* figure, they are entered into a task queue. An *attention resource manager* [14] scans this queue for current and future visual sensing requirements, and directs *Jack*'s eye gaze (and hence head movement) accordingly. For example, if the agent is being told to "remove the power supply," parallel instructions are generated to locomote to the power supply area and attend to specific visual attention tasks such as searching for the power supply, scanning for potential moving objects, and periodically watching for obstacles near the feet. Note that normally none of this attentional information appears explicitly in the task-level instruction stream, yet attentional and sensing actions consume finite amounts of time and accordingly pace other actions.

2.6. Locomotion with anticipation

In order to interact with a target object, an agent must determine that it is not within a suitable distance and must therefore locomote to a task-dependent position and orientation prior to the initiation of the reach and grasp. Such a decision is readily made by embedding it in a **PaT-Net** representing potential actions that enable the specified action. Moreover, the locomotion process itself uses the two level architecture: at the lowest level the agent or avatar gets a goal and an explicit list of objects to be avoided; the other level encapsulates locomotion states and decisions about transitions. For example, the agent could be walking, hiding, searching, or chasing. If walking, then transitions can be based on evaluating the best position of the foot relative to the goal and avoidances. If hiding, then assessments about line of sight between virtual humans are computed. If searching, then a pattern for exhaustively checking the local geometry is invoked. Finally, if chasing, then the goal is the target object; but if the target goes out of sight, the last observed position is used as an interim goal. These sensing actions and resulting decisions are captured in the LocoNet [41].

2.7. Multi-agent task allocation

By encapsulating virtual human activities in **PaT-Nets**, we can interactively control the assignment of tasks to agents. A menu or program binds actions to individuals, who then execute the **PaT-Net** processes. Since the processes have the power to query the environment and other agents before starting to execute, multi-agent synchronization and coordination can be modeled. Thus an agent can start a task when another signals that the situation is ready, or one agent can lead another in a shared task. The latter would be especially useful when an avatar works with a simulated agent to perform a two-person task. One virtual human is designated as the "leader" (typically the avatar, so the live participant is in control) and the other the "follower." The follower's timing and motion are performed after each time-stepped motion of the leader. (The reverse situation, where the agent leads the avatar, may be needed for training and educational applications.) These are clearly the first steps toward a virtual social architecture.

Once we can generate and control multiple agents and avatars, many social and community issues arise including authentication of identity, capabilities, permissions, social customs, transference of object control, sharing behaviors, coordinating group tasks, etc. Underlying technology to

share interactive experience will depend on distributed system protocols and communication technology, client workstation performance, avatar graphics, and so on. Many of these issues are being addressed by other *ad hoc* groups, such as *Living Worlds* [32], *Open Community* [37], and *Universal Avatars* [51]. Having two avatars "shake hands" is considered the first stage of a social encounter requiring significant attention to the details of avatar interaction, body representation, and action synchronization. Assuming that the communications can be done fast enough (a big assumption), our avatars should be able to reach for each other's hand, detect a collision/connection, and then allow the follower avatar to position his/her hand according to the leader's spatial position. Indeed, such a demonstration has already been readily constructed by Stansfield at Sandia National Labs with *Jack* avatars, in-house network communication software, head-mounted displays, and end effector position/orientation sensors on the participants. Handshaking between virtual agents is discussed in the context of **Improv** [39]. Agent and avatar handshaking has also been considered in our *JackMOO* [47].

3. Connecting Language and Animation

Even with a powerful set of motion generators, a challenge remains to provide effective and easily learned user interfaces to control, manipulate and animate virtual humans. Interactive point and click systems such as *Jack* work now, but with a cost in user learning and menu traversal. Such interfaces decouple the human participant's instructions and actions from the avatar through a narrow and *ad hoc* communication channel of hand and finger motions. A direct programming interface, while powerful, must be rejected as as off-line method that moreover requires specialized computer programming understanding and expertise. The option that remains is a language-based interface.

Perhaps not surprisingly, instructions for people are given in natural language augmented with graphical diagrams and occasionally, animations. Recipes, instruction manuals, and interpersonal conversations use language as the medium for conveying process and action. While our historic interest in instructions has been on creating animations from instructions [5, 3, 52], we have recently begun to examine the inverse process, namely, generating text from the **PaT-Net** representations of animations. The purpose is primarily to help automate the production of aircraft maintenance instruction orders (manuals) in conjunction with the animation of the tasks themselves. The expectation is that the synthesized *text* material ought to reflect the proper execution of the tasks (which can be *visually* verified through the animation) and will have consistency across the entire document. By the same principles, being able to process the textual instructions will aid in discovering ambiguities, omitted steps, or inappropriate terminology.

The key to linking language and animation lies in constructing a semantic representation of actions, objects, and agents which is simultaneously suitable for execution (animation) as well as natural language expression. We have called this *implementable semantics*: the representation must have the power of a (parallel) programming language which drives a simulation (in a context of a given set of objects and agents), and yet supports the enormous range of expression, nuance, and manner offered by language. We consider three aspects of this problem in the remainder of this paper. The first part (Section 4) briefly describes Tsukasa Noma's *Jack Presenter* [36], the second (Section 5) considers a 3D avatar extension called *JackMOO* [47] to an existing lambdaMOO, and the third (Section 6) constructs a draft specification for a Parameterized Action Representation (PAR) which uses **PaT-Net** as an implementation language [6].

4. Jack Presenter

[1]During his sabbatical stay at the University of Pennsylvania, Tsukasa Noma[2] created a virtual human "presenter." Based on extensions to *Jack*, the inputs to the presenter system are in the form of speech texts with user- or program-generated embedded commands, most of which relate to the virtual presenter's body language. As the text is processed, the Jack presenter acts out the speech with the requested gestures to both a texture-mapped "white-board" or image plane as well as to the listener (Figs. 3 and 4). Important components of this system include:

- Proper inputs for representing presentation scenarios.
- Natural motion with presentation skills.
- Real-time motion generation synchronized with speech.

We will examine each of these briefly.

The input consists of text to be spoken through a speech synthesizer and commands to affect the presentation. The text may be created in advance and manually annotated with commands to load an image onto the board, point at a site on the image, or gesture towards the board or audience. A socket interface permits the on-line generation of the text and commands from another program. In this mode, a sophisticated control program such as designed for Gesture Jack [12] could pass instructions to the presenter in real-time.

[1]The content of this Section is strongly based on work by Tsukasa Noma [36] and is included with the permission of the author.

[2]on a Japanese Ministry of Education, Science, Sports and Culture overseas research fellowship.

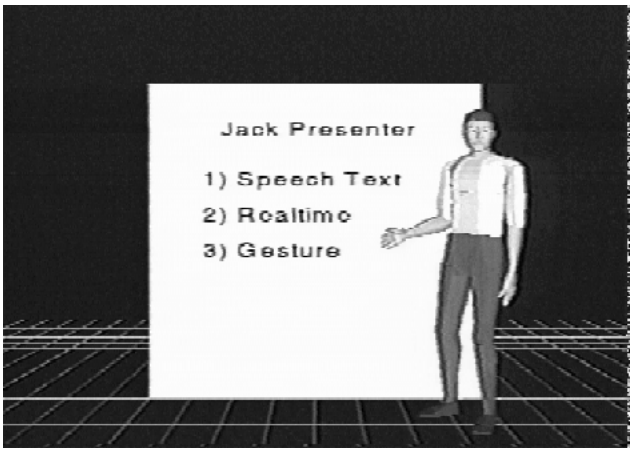

Figure 3. The *Jack presenter* points to the slide and looks at the audience.

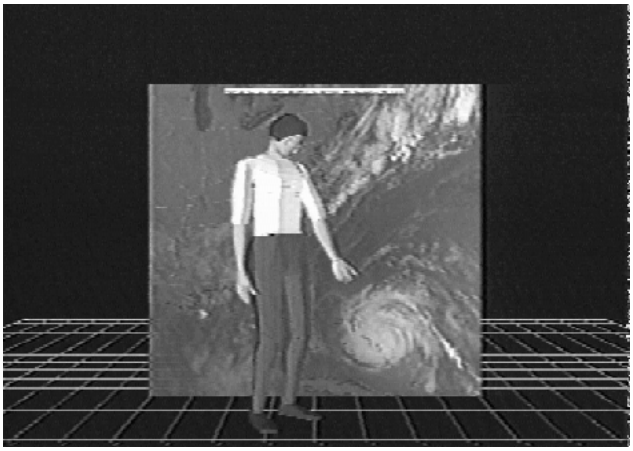

Figure 4. *Jack presenter* points to and looks at the hurricane on the weather map.

The presenter's motions are designed to make him look like a teacher with a visual aid board. The board can display any text or texture. As a pre-processing step, the presentation designer associates sites (coordinate systems) with interesting features on the image, such as the text position or the eye of the hurricane. (We do not yet deal with the hard problem of automatically identifying the interesting features from the image itself.) As the presenter talks, he has to utilize three skills:

- Where to point and with which hand.
- What to gesture to the audience.
- Where to look: the board or the audience.
- Where to place himself to maximize visibility of the image feature and line of regard from a pointing gesture.

The execution involves a combination of user selections (e.g. what to point at) and decision-making (when to step over to the other side of the board, when to use the other hand). Eye gaze, for example, is towards the pointed-at site during a pointing gesture but toward the audience otherwise. The "rules of presentation" are coded in **PaT-Nets** (in fact, a **C++** version called LWNets (Light Weight PaT-Nets) especially written by Noma in order to maximize real-time control).

The presenter's actions are controlled in real-time (on an SGI RE2) as the **PaT-Net** executes the input stream. The text and command stream, however, *contains no timing information*. There are only two sources for such information, and these are used implicitly to schedule and synchronize the animation. The first is the text stream itself, which is sent to a text-to-speech system – Entropic Research Laboratory's TrueTalk^{TM} TTS (Text-To-Speech) system[19] running on a SGI Indigo2. Basically, the motion specified by a command is to coincide with the utterance of a word following the command in the input. The second source is the performance of gesture or locomotion by the presenter. Since these actions are determined by **PaT-Nets**, transitions to interpret the next command in the input stream can be delayed unti, the gesture or locomotion action has completed. Since the latter is context-dependent, a useful level of autonomy and synchrony results.

Extensions planned for this system input better facial expressions to correlate with the text stream and the presenter/audience interaction in the style of Gesture Jack, more flexible presentation style rules, and increased autonomy in generating the pointing and gesturing commands from the text itself. The latter, of course, will require a deep understanding of the board contents and the text itself – for example, to extract emphasis or affect. Finally, creating a presenter that talks to another simulated individual will be a useful exercise in shared communication, dialogue structure, and environment- and object-sensitive interactions.

5. JackMOO

[3]We have prototyped a prototype system, *JackMOO*, which combines *Jack* and LambdaMOO [15], a multi-user, network-accessible, programmable, interactive system which has been used for the construction of text-based conferencing, educational/training, and other collaborative software. *JackMOO* allows us to store the richer semantic information necessitated by the scope and range of human actions that an avatar must portray, to express those actions in the form of natural, imperative sentences. *JackMOO*

[3]Adapted from [47].

therefore provides us with an testbed for language control of avatar animation. This section describes *JackMOO* and its components, especially a *JackMOO* client program that mediates the flow of control between *Jack* and the lambdaMOO server and provides the primary user interface to the system.

Of central importance to *JackMOO* is the association of human action verbs with possibly several **PaT-Nets** that realize the action on the virtual human on the *Jack* display. Actions as *step-forward*, *turn-around*, and *look-at*, are specified in the *Jack* environment in the form of executable programs, providing the level of interface necessary for the control of a virtual human avatar in a virtual world. **PaT-Nets** thus function as a high-level API accessing underlying *Jack* behavior and functionality.

As an action programming interface, **PaT-Nets** provide the author with too many choices – essentially an unconstrained parallel language. To facilitate human action authoring through more syntactically and semantically structured forms, we are designing a *Parameterized Action Representation*, outlined below.

6. Parameterized Action Representation

[4]It is convenient to graphically present processes as *nodes* in which some action, change, or function takes place, and *arcs* which link one process (node) to another that temporally follows either by virtue of culmination (completion) of the first or other circumstances. A process can be recursively defined as a network (or graph) of process nodes (possibly disconnected, i.e. parallel). Thus, a hierarchy of processes can exist, grounding out at single process nodes for the simplest types of processes.

An *action* is just a particular kind of process which involves a volitional agent acting in the world. We call our representations of actions *Parameterized Action Representations* (PARs) and they contain a necessary slot for an agent. A generic process representation is a PAR with an optional agent slot. Our representation is a modified version of the representation used by Kalita and Lee [26], expanded to include culmination conditions, agent/object representations, as well as more detail about the specifics of actions.

The top-level type in the representation is the *parameterized action*; an action depends on its participants (agent and objects) for the details of how it to be accomplished. For instance, opening a door and opening a window will involve very different behaviors on the part of the agent (e.g., [31, 18]). The subparts of a parameterized action can refer to particular aspects of the agent and objects as part of their meaning.

In order to produce animation, actions represented in the PAR must be converted into **PaT-Nets**. All the actions of an agent which correspond to a given set of instructions are referred to as the top-level actions and are maintained at the highest level in a queue tree. Each of these high level actions might have subactions. All these subactions are now maintained in a queue at the next level. Sensing actions are considered as finite duration processes, and so are also considered during action execution. For every action, a **PaT-Net** is spawned. For every high level action, the subactions form the children and the higher level action is assumed completed only after all the children's actions are completed. An action is also considered completed if the culmination conditions of some higher level **PaT-Net** are satisfied. A sequence of actions is maintained as children from left to right, the leftmost child being executed first. Once an action is completed, the action on its right is then considered. Further details can be found in the draft report [6].

[4]This section is adopted from [6].

7. Conclusions

This paper has described the current status of virtual human modeling and control, with an emphasis on real-time motion and language-based interfaces. In particular, we discussed such issues as appearance and motion, interactive control, autonomous action, gesture, attention, locomotion, and multiple individuals. The underlying *Jack* architecture consists of a sense-control-act structure that permits reactive behaviors to be locally adaptive to the environment, and a **PaT-Net** parallel finite-state machine controller that can be used to drive virtual humans through complex tasks.

A real-time *Jack Presenter* demonstrated the feasibility of controlling pointing gestures, attention, body motion, and speech through a uniform interface processed by **PaT-Nets**. An important component of this study was the computation of movements not directly specified in the text nor its annotation. In addition, actions were synchronized with the text and the execution of other unspecified actions such as locomotion to a better presentation position.

The *JackMOO* is a virtual world environment combining *Jack* with an existing multi-user technology LambdaMOO. The *JackMOO* hybrid focuses on 3D human-like avatars and employs an English-like language interface (imperative sentences) to control them. *JackMOO* provides a flexible environment in which pilot/drone and leader/follower roles may be specified and used to advantage in training and educational 3-dimensional scenarios.

We next described a the top level of a Parameterized Action Representation. The PAR is meant to be the intermediate structure between simple natural language imperative sentences with complex semantics and task execution by a virtual human agent. There are many dimensions to the PAR, including slots for the agent, participating objects, applicability conditions, culmination conditions, spatiotemporal

descriptions, agent manner, and suggested subactions. An algorithm for interpreting PARs within an object-oriented system has been designed, based on the *JackMOO* framework.

The future holds great promise for the virtual humans who will populate our virtual worlds. They will provide economic benefits by helping designers early in the product design phases to produce more human-centered vehicles, equipment, assembly lines, manufacturing plants, and interactive systems. Virtual humans will enhance the presentation of information through training aids, virtual experiences, and even teaching and mentoring. And Virtual humans will help save lives by providing surrogates for medical training, surgical planning, and remote telemedicine. They will be our avatars on the Internet and will portray ourselves to others, perhaps as we are or perhaps as we wish to be. They may help turn cyberspace into a real, or rather virtual, community.

Acknowledgments

The many students, staff, and colleagues in the Center for Human Modeling and Simulation make this effort possible. In particular, special thanks go to Rama Bindiganavale, Diane Chi, Tsukasa Noma, Ken Noble, Sean Sheridan, and Bond-Jay Ting for the illustrations. The satellite image of Hurricane Bertha in Fig. 4 is from NOAA/National Climatic Data Center. Additional information and contributors may be found through **http://www.cis.upenn.edu/~hms**.

This research is partially supported by DARPA DAMD17-94-J-4486; U.S. Air Force DEPTH through Hughes Missile Systems F33615-91-C-0001; U.S. Air Force through BBN F33615-91-D-0009/0008; U.S. Air Force DAAH04-95-1-0151; DMSO DAAH04-94-G-0402; ONR through Univ. of Houston K-5-55043/3916-1552793; ARO DURIP DAAH04-95-1-0023; DARPA SB-MDA-97-2951001 through the Franklin Institute; Army AASERT DAAH04-94-G-0220; DARPA AASERT DAAH04-94-G-0362; NSF IRI95-04372; National Library of Medicine N01LM-43551; National Institute of Standards and Technology 60 NANB6D0149; and JustSystem Japan.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation nor any other sponsoring organization.

References

[1] F. Azuola, N. Badler, P.-H. Ho, I. Kakadiaris, D. Metaxas, and B. Ting. Building anthropometry-based virtual human models. In *Proc. IMAGE VII Conf.*, 1994.

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

[3] N. Badler, C. Phillips, and B. Webber. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, New York, NY, 1993.

[4] N. Badler, B. Webber, W. Becket, C. Geib, M. Moore, C. Pelachaud, B. Reich, and M. Stone. Planning for animation. In N. Magnenat-Thalmann and D. Thalmann, editors, *Computer Animation*. Prentice-Hall, 1996. To appear.

[5] N. Badler, B. Webber, J. Kalita, and J. Esakov. Animation from instructions. In N. Badler, B. Barsky, and D. Zeltzer, editors, *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*, pages 51–93. Morgan-Kaufmann, 1990.

[6] N. Badler, B. Webber, M. Palmer, T. Noma, M. Stone, J. Rosenzweig, S. Chopra, K. Stanley, J. Bourne, and B. Di Eugenio. Final report to Air Force HRGA regarding feasibility of natural language text generation from task networks for use in automatic generation of Technical Orders from DEPTH simulations. Technical report, CIS, University of Pennsylvania, 1997.

[7] J. Bates. The role of emotion in believable agents. *Comm. of the ACM*, 37(7):122–125, 1994.

[8] J. Bates, A. Loyall, and W. Reilly. Integrating reactivity, goals, and emotion in a broad agent. In *Proc. of the 14th Annual Conf. of the Cognitive Science Society*, pages 696–701, Hillsdale, NJ, 1992. Lawrence Erlbaum.

[9] BDI-Guy, 1996. Boston Dynamics Inc., Cambridge, MA.

[10] W. Becket. *Reinforcement Learning of Reactive Navigation for Computer Animation of Simulated Agents*. PhD thesis, CIS, University of Pennsylvania, 1997.

[11] A. Bruderlin and L. Williams. Motion signal processing. In *Computer Graphics, Annual Conf. Series*, pages 97–104. ACM, 1995.

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

[13] D. Chi, B. Webber, J. Clarke, and N. Badler. Casualty modeling for real-time medical training. *Presence*, 5(4):359–366, 1995.

[14] S. Chopra. Strategies for simulating direction of gaze and attention. Technical report, Center for Human Modeling and Simulation, University of Pennsylvania, 1995.

[15] P. Curtis. LambdaMOO, 1997. Xerox PARC Ftp site: parcftp.xerox.com/pub/MOO.

[16] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Proc. CVPR*, pages 231–238. IEEE Press, 1996.

[17] S. Deutsch, J. MacMillan, N. Cramer, and S. Chopra. Operator Model Architecture (OMAR) support. Technical Report 8179, BBN, Cambridge, MA, 1997.

[18] B. Douville, L. Levison, and N. N. Badler. Task level object grasping for simulated agents. *Presence*, 5(4):416–430, 1996.

[19] TrueTalk Programmer's Manual, 1995. Entropic Research Laboratory.

[20] I. Essa and A. Pentland. Facial expression recognition using a dynamic model and motion energy. In *Proc. of the International Conf. on Computer Vision*, Cambridge, MA, 1995.

[21] M. Girard and A. Maciejewski. Computational modeling for the computer animation of legged figures. *ACM Computer Graphics*, 19(3):263–270, 1985.

[22] J.-P. Gourret, N. Magnenat-Thalmann, and D. Thalmann. Simulation of object and human skin deformations in a grasping task. *ACM Computer Graphics*, 23(3):21–30, 1989.

[23] J. Granieri, J. Crabtree, and N. Badler. Off-line production and real-time playback of human figure motion for 3D virtual environments. In *VRAIS Conf.* IEEE Press, 1995.

[24] J. Hodgins, W. Wooten, D. Brogan, and J. O'Brien. Animating human athletics. In *ACM Computer Graphics, Annual Conf. Series*, pages 71–78, 1995.

[25] I. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion with occlusion based on active multiviewpoint selection. In *Proc. of the Conf. on Computer Vision and Pattern Recognition*, pages 81–87. IEEE Computer Society, June 1996.

[26] J. Kalita and J. Lee. An informatl semantic analysis of motion verbs based on physical primitives. *Computational Intelligence*, 1996.

[27] H. Ko and N. Badler. Animating human locomotion in real-time using inverse dynamics, balance and comfort control. *IEEE Computer Graphics and Applications*, 16(2):50–59, March 1996.

[28] Y. Koga, K. Kondo, J. Kuffner, and J.-C. Latombe. Planning motions with intentions. In *ACM Computer Graphics, Annual Conf. Series*, pages 395–408, 1994.

[29] E. Kokkevis, D. Metaxas, and N. Badler. User-controlled physics-based animation for articulated figures. In *Computer Animation Conf. Proc.*, 1996.

[30] D. Kurlander, T. Skelly, , and D. Salesin. Comic chat. In *ACM Computer Graphics, Annual Conf. Series*, pages 225–236, 1996.

[31] L. Levison. *Connecting planning and acting via object-specific reasoning*. PhD thesis, CIS, University of Pennsylvania, 1996.

[32] Living Worlds, 1997. http://www.livingworlds.com/draft_1/index.htm.

[33] P. Maes, T. Darrell, B. Blumberg, and A. Pentland. The ALIVE system: Full-body interaction with autonomous agents. In N. Magnenat-Thalmann and D. Thalmann, editors, *Computer Animation*, pages 11–18. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[34] D. Metaxas. *Physics-Based Deformable Models: Applications to Computer Vision, Graphics and Medical Imaging*. Kluwer, Boston, MA, 1996.

[35] M. Moore, C. Geib, and B. Reich. Planning and terrain reasoning. Technical Report MS-CIS-94-63, CIS, University of Pennsylvania, Philadelphia, PA, 1994.

[36] T. Noma and N. Badler. A virtual human presenter. In *IJCAI '97 Workshop on Animated Interface Agents*, Nagoya, Japan, 1997.

[37] Open Community, 1997. http://www.merl.com/opencom/opencom.htm.

[38] K. Perlin. Real time responsive animation with personality. *IEEE Trans. on Visualization and Computer Graphics*, 1(1):5–15, 1995.

[39] K. Perlin and A. Goldberg. Improv: A system for scripting interactive actors in virtual worlds. In *ACM Computer Graphics, Annual Conf. Series*, pages 205–216, 1996.

[40] D. Pratt, P. Barham, J. Locke, M. Zyda, B. Eastman, T. Moore, K. Biggers, R. Douglass, S. Jacobsen, M. Hollick, J. Granieri, H. Ko, and N. Badler. Insertion of an articulated human into a networked virtual environment. In *Proc. of the Conf. on AI, Simulation and Planning in High Autonomy Systems*. University of Florida, Gainesville, 1994.

[41] B. Reich. *An architecture for behavioral locomotion*. PhD thesis, CIS, University of Pennsylvania, 1997.

[42] B. Reich, H. Ko, W. Becket, and N. Badler. Terrain reasoning for human locomotion. In *Proc. Computer Animation '94*, pages 996–1005. IEEE Computer Society Press, 1994.

[43] B. Robertson. Best behaviors. Digital magic. *Computer Graphics World (Supplement)*, pages S12–S19, 1996.

[44] C. Rose, B. Guenter, B. Bodenheimer, and M. Cohen. Efficient generation of motion transitions using spacetime constraints. In *ACM Computer Graphics, Annual Conf. Series*, pages 147–154, 1996.

[45] D. Rousseau and B. Hayes-Roth. Personality in synthetic agents. Technical Report KSL-96-21, Stanford Knowledge Systems Laboratory, 1996.

[46] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *ACM Computer Graphics*, 20(4):151–160, 1986.

[47] T. Smith, J. Shi, J. Granieri, and N. Badler. Jackmoo: A prototype system for natural language avatar control. In *Pacific Graphics*, 1997. Submitted.

[48] S. Stansfield. Distributed virtual reality simulation system for situational training. *Presence*, 3(4):360–366, 1994.

[49] S. Stansfield, D. Carlson, R. Hightower, , and A. Sobel. A prototype VR system for training medics. In *MMVR Proc.*, 1997. (To appear).

[50] D. Tolani and N. Badler. Real-time inverse kinematics for the human arm. *Presence*, 5(4):393–401, 1996.

[51] Universal Avatars, 1996. http://www.chaco.com/avatar.

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

[53] J. Zhao and N. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, 13(4):313–336, 1994.

[54] X. Zhao. *Kinematic control of human postures for task simulation*. PhD thesis, CIS, University of Pennsylvania, 1996.