Virtual Humans for Animation, Ergonomics, and Simulation

Norman 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

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. Finally, we argue for a deep connection between language and animation and describe current efforts in linking them through the JackMOO extension to lambdaMOO.*

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 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.

From these virtual humans research areas, many current, emergent, or future major applications are enabled:

- Engineering: Analysis and simulation for virtual prototyping and simulation-based design.
- Virtual-Conferencing: Efficient tele-conferencing using virtual representations of participants to reduce transmission bandwidth requirements.
- Interaction: Agents and avatars that insert real-time humans into virtual worlds with virtual reality.
- Monitoring: Acquiring, interpreting, and understanding shape and motion data on human movement, performance, activities, or intent.
- Virtual Environments: Living and working in a virtual place for visualization, analysis, training, or just the experience.
- Games: Real-time characters with actions and personality for fun and profit.
- Training: Skill development, team coordination, and decision-making.
- Education: Distance mentoring, interactive assistance, and personalized instruction.
- Military: Battlefield simulation with individual participants, team training, and peace-keeping operations.
- Design/Maintenance: Design for access, ease of repair, safety, tool clearance, visibility, and hazard avoidance.

Besides general industry-driven improvements in the underlying computer and graphical display technologies themselves, virtual humans will enable quantum leaps in applications requiring personal and live participation.

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?

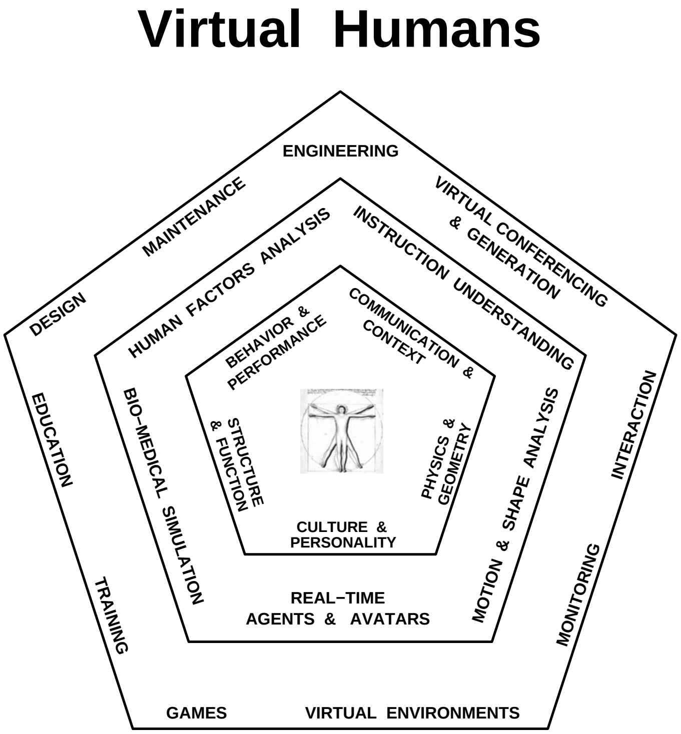

Figure 1. Virtual human applications, technology, and science.

- 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 − − − − − + −− > Physiologically accurate model
- Function: Cartoon actions − − − − − + − > Human limitations
- Time: Off-line generation − − − − − + − > Real-time production
- Autonomy: Direct animation − − − − − + −− > Intelligent
- Individuality: Specific person − − − + − − − −− > Varying personalities

The arrows and hash marks are meant to be qualitative indicators of where we think usable technology exists today. Understanding that the arrows can actually extend an undetermined distance to the right, the idea is nonetheless being conveyed that we (and others) have proceeded rather far beyond the individual 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 biomechically-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 *automony* 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 [21, 22, 9, 29, 34, 40].

The University of Pennsylvania has been very actively engaged in research and development of human-like simulated figures. Our interest in human simulation is not unique, but the complex of activities surrounding our approach is. The framework for our research is a software system called *Jack* [3]. *Jack* is an interactive system for definition, manipulation, animation, and performance analysis of virtual human figures. Our philosophy has led to a particular realization of a virtual human model that pushes the above five dimensions toward the right:

- can be substituted for live individuals for workspace or cockpit evaluation.
- demonstrates various (useful) human limitations, constraints, and capabilities.
- may be moved live (in real-time) by position and orientation information or other motion generators such as *walk-to* or *look-at*.
- may have its actions synthesized by a program so that it can make its own decisions, navigate spaces, and so on.
- represents "anyone" rather than a single specific person or character.

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 [27], composited video, 3D shapes, or full 3D bodies [2, 42, 38]. 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 [18].

Along the appearance dimension, the *Jack* 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 [41] to aid in the portrayal of visually appealing virtual humans (Fig. 2.1).

The motions manifest in the avatar may arise from various sources:

- Motion capture from direct live video
- Motion capture from sensors
- Pre-stored motion data
  - as 2D sprites
  - as 3D global transformations

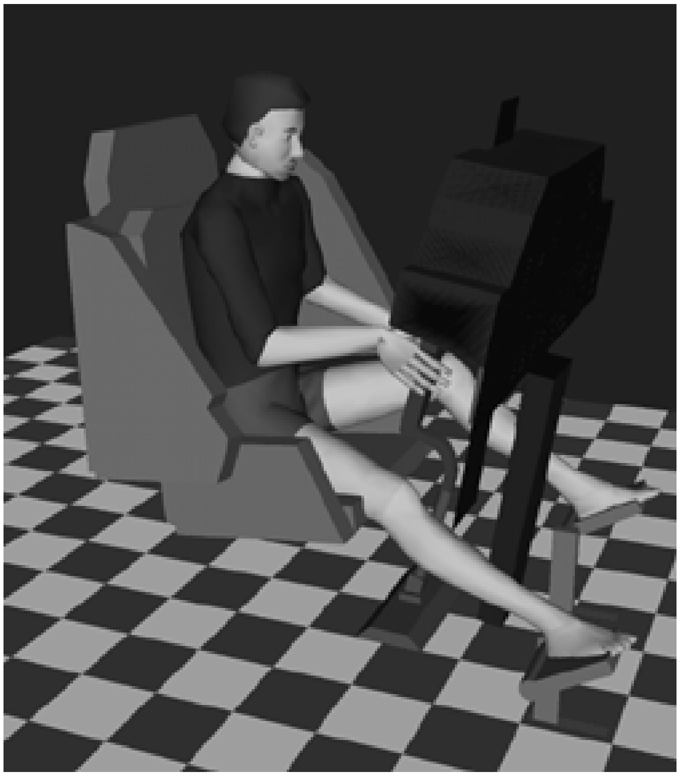

Figure 2. Smooth body *Jack* as virtual occupant in an Apache helicopter CAD model.

  - as 3D local (joint) transformations

- Motion synthesis
  - joint angle interpolation
  - inverse kinematics
  - dynamics
  - other generators (e.g. locomotion, faces)

In general, we will not consider 2D or purely video presentations of avatars, rather we will concentrate on avatars that more-or-less mimic human structure.

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 [18]. 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 [33].

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 [46, 25, 44],
- a path or footsteps can control leg and foot rotations through a locomotion model [16, 24],
- a balance constraint can be superimposed on gross body motions [3, 24],
- dynamics calculations can move joints subject to arbitrary external and internal applied forces [26, 30],
- secondary motions can enhance a simpler form [33, 19].

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, 25]. 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 [8, 39]. 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 [35, 18, 20].

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, 15, 12, 23], 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 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 [43] and in the responsive characters in Improv [33, 34].) 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) [37, 24, 7];

- 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) [31, 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], to two person animated conversation [9], or to simulated emergency medical care [10]. 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 [17, 14], and they serve social interaction functions by augmenting the speech channel with communicative emblems, gestures and beats [9].

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* [28], 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 [9]. 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

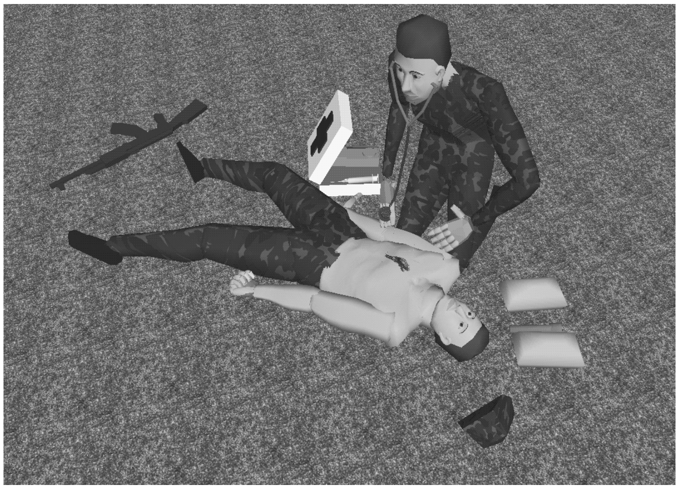

Figure 3. *Jack* **as virtual casualty and medic for training scenario.**

function, fast computation of arm joint angles is essential. In recent work we have pushed beyond iterative inverse kinematics [46] to analytic formulas that can easily keep up with a live performance or a motion synthesizer outputting end effector position and orientation streams [44]. 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 [47].

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) [13], 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* [11] 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.

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 [36].

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.

We developed a prototype system for agent task assignment to evaluate a multi-function aircraft maintenance equipment cart ("MASS"). The user specifies tasks for an agent, and the agent accepts tasks for which it is both qualified and responsible. The tasks can be queued in advance, and are executed as prior tasks are completed or other agent or environment conditions obtain.

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*, *Open Community*, and *Universal Avatars*. 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.

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, 45], 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. The details of this Parameterized Action Representation (PAR) – which involves PaT-Nets as an implementation language – are being developed in a companion document [6].

As a prototype implementation of this language–animation connection, we are constructing *JackMOO*: a multi-user environment mediated by an existing system called *lambdaMOO* [32]. The lambdaMOO is a text-based multi-user world. By adding a *Jack* system and an additional dialog box to the user interface, the user can instruct his or her avatar to take steps, go into a neighboring room, turn on a TV set, and so on. While the text input is quite constrained by the limited "verb-object-modifier" syntax of lambdaMOO, it is nonetheless very efficient for specifying the avatar's actions. The JackMOO updates a local client's 3D animated view of the instruction executed on the *pilot* avatar[1]. Simultaneously, the lambdaMOO updates the persistent world view (on the lambdaMOO server) and informs (textually) other users occupying the same virtual room of the avatar's actions. Should one of the other users have a JackMOO interface, the avatar's actions will be mirrored by the user's drone on that client's display. In effect, the actions are specified in *either* a textual or graphical fashion and executed as *both* a textual and database update and (if possible) an animated display. It is thus essential that discrete (text-based) instructions have continuous (animation) consequences: the conversion is based on action ("verb") semantics embedded into object-specific methods stored in the lambdaMOO objects. We are extending this object-oriented system to store the richer semantic information necessitated by the scope and range of human actions that an avatar must portray.

4. Conclusions

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

[1] In VRML terminolgy, the *pilot* is the agent or avatar associated with the movement source (user or program); a drone is a remote (networked) copy of the pilot.

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 Diane Chi, Rama Bindiganavale, Sean Sheridan, Ken Noble, and Bond-Jay Ting for the illustrations. 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.

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

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

[9] 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.

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

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

[12] 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.

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

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

[15] 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.

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

[17] 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.

[18] 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.

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

[20] B. D. Inc. BDI-Guy, 1996. Software system.

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

[22] B. J., L. A., and R. W. Integrating reactivity, goals, and emotion in a broad agent. In *Proc. of the* 14^{th} *Annual Conf. of the Cognitive Science Society*, pages 696–701, Hillsdale, NJ, 1992. Lawrence Erlbaum.

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

[24] 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.

[25] 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.

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

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

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

[29] 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.

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

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

[32] X. PARC. LambdaMOO, 1997. Ftp site: parcftp.xerox.com/pub/MOO.

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

[34] 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.

[35] 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.

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

[37] 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.

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

[39] 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.

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

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

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

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

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

[45] 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.

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

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