

Animating Virtual Humans in Intelligent Multimedia Storytelling

Minhua Ma and Paul Mc Kevitt
School of Computing & Intelligent Systems
Faculty of Engineering, University of Ulster, Magee
Derry/Londonderry, BT48 7JL, Northern Ireland
{m.ma, p.mckevitt}@ulster.ac.uk

Abstract—There are a wide range of languages and technologies for modeling and animation of virtual humans. Here, we review current state-of-the-art virtual human animation standards and present our work on language visualisation (animation) in our intelligent multimodal storytelling system, CONFUCIUS. We provide an overview of the system and the technologies employed in the animation engine. We discuss several issues in human animation, such as multiple animation channels, space sites of virtual humans, and object manipulation. Finally, we conclude that introducing linguistic knowledge provides more intelligent multimedia storytelling.

Keywords: multimedia content, 3D animation generation, virtual human, language visualization, multimodal storytelling

I. INTRODUCTION

3D graphics web applications such as online games, virtual environments, and intelligent agents, are more and more demanding 3D graphics modelling languages [1] that represent not only virtual objects but virtual humans and their animation. Existing virtual humans and animation on the Web are created by various authoring tools (e.g. 3D Studio Max, Maya, Poser, and motion capture devices) and in different formats. Current 3D human standards (e.g. VRML, MPEG-4) aim for various levels of abstraction, especially for lower level representations. Here, we review state-of-the-art virtual human modelling and animation languages and presents our work on language visualisation (animation) in our intelligent multimodal storytelling system, CONFUCIUS. It shows that the integration of linguistic knowledge with these modelling languages can achieve higher level representation of virtual human modelling and animation: it leads to automatic 3D animation generation from natural language, and in the long run, to more intelligent multimedia storytelling which includes animation and speech.

First, in section II we review various virtual human representation languages and group them into four levels of abstraction, starting from 3D geometry modelling to language animation. Next in section III, the intelligent multimedia storytelling system, CONFUCIUS, is introduced and its architecture is described. We also discuss several issues in CONFUCIUS' virtual character animation generation. Then we compare our work to related research on virtual human animation in section IV. Finally, section V summarizes the work with a discussion of possible future work on multiple character coordination and synchronization.

II. VIRTUAL HUMANS AT VARIOUS LEVELS OF ABSTRACTION

We investigated current virtual human representation languages and found that they can be classified to four groups according to the levels of abstraction, starting from 3D geometry modelling to language animation. Fig.1 illustrates these four levels of virtual human representation. Most work on virtual human modelling and animation focuses on the lower levels.

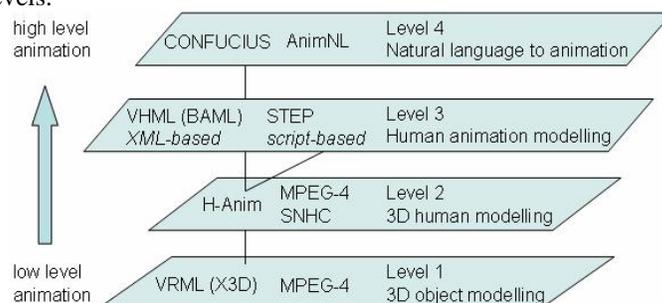


Fig. 1. Four levels of virtual human representation

A. VRML (X3D) and MPEG-4 for Object Modelling

The first level is for 3D object modelling. VRML (X3D) and MPEG-4 are two leading standards of 3D content for Web applications. VRML (Virtual Reality Modelling Language), developed by the Web3D Consortium (originally the VRML Consortium), is a hierarchical scene description language that defines the geometry and behaviour of a 3D scene and the way in which it is navigated by the user. X3D is the successor to VRML. It extends VRML with new features, advanced APIs, additional data encoding formats (VRML97 and XML), and a component-based architecture that permits a modular approach. VRML (X3D) is the standard used most widely on the Internet to describe 3D objects/humans and users' interaction with them.

Unlike VRML, MPEG-4 uses BIFS (Binary Format for Scenes) for real-time streaming, i.e. a scene does not need to be downloaded in full before it can be played, but can be built up on the fly. BIFS borrows many concepts from VRML. BIFS and VRML can be seen as different representations of the same data. In VRML, the objects and their actions are described in text, but BIFS code is binary, and thus is shorter for the same content—typically 10 to 15 times.

B. H-Anim and MPEG-4 SNHC for Humanoid Modelling

The second level is for 3D human modelling. H-anim [2] is a standard VRML97 representation for humanoids. It defines standard human *Joints* articulation (e.g. knee and ankle), *Segments* dimensions (e.g. thigh, calf, and foot), and *Sites* (e.g. hand_tip, foot_tip) for “end effector” and attachment points for clothing. An H-Anim file contains a joint-segment hierarchy as shown in Fig. 2. Each joint node may contain other joint nodes and a segment node that describes the body part associated with the joint. Each segment is a normal VRML transform node describing the body part's geometry and texture. H-Anim humanoids can be animated using keyframing, inverse kinematics (IK), and other animation techniques. Table 1 lists H-Anim models available on the Internet.

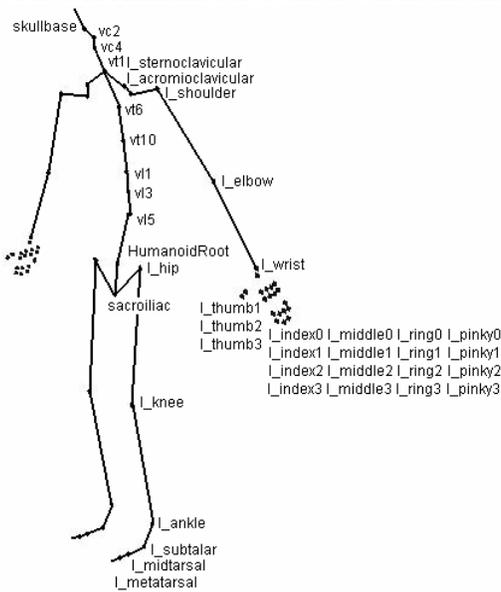


Fig. 2. H-Anim joint hierarchy

MPEG-4 SNHC (Synthetic/Natural Hybrid Coding) is concerned with the compression of media streams, such as geometry, animation parameters, or text-to-speech, beyond traditional audio and video, the representation and coding of synthetic objects as well as their natural A/V counterparts, and the spatial-temporal composition of these natural and synthetic objects. MPEG-4 SNHC offers an appropriate framework for 3D virtual human animation, gesture synthesis and efficient compression/transmission of these animations.

SNHC incorporates H-Anim and provides an efficient way to animate virtual human bodies and tools for the efficient compression of the animation parameters associated with the H-Anim articulated human model. It defines two sets of parameters: body definition and animation parameters. Body definition includes face deformation parameters (FDPs) and body deformation parameters (BDPs). These let the decoder specify shape and texture of a model. Animation parameters include face animation parameters (FAPs) and body animation parameters (BAPs). BAPs are a set of rotation angles of body parts to specify posture.

TABLE 1.
H-ANIM MODELS ON THE WEB

<i>H-anim models</i>	<i>Names</i>	<i>Authors, URLs</i>
	Nancy	Cindy Ballreich http://www.ballreich.net/vrml/h-anim/nancy_h-anim.wrl
	Baxter Nana	Christian Babski http://ligwww.epfl.ch/~babski/StandardBody
	Y.T. Hiro	Matt Beitler http://www.cis.upenn.edu/~beitler/H-Anim/Models/H-Anim1.1/
	Dilbert	Matt Beitler http://www.cis.upenn.edu/~beitler/H-Anim/Models/H-Anim1.1/dilbert/
	Max	Matt Beitler http://www.cis.upenn.edu/~beitler/vrml/human/max/
	Jake	Matt Beitler http://www.cis.upenn.edu/~beitler/H-Anim/Models/H-Anim1.1/jake/
	Dork	Michael Miller http://students.cs.tamu.edu/mmiller/hanim/proto/dork-proto.wrl

C. VHML and STEP for Human Animation Modelling

The third level is for human animation modelling. Following the lead of W3C's SMIL (Synchronized Multimedia Integration Language) [3], VHML (Virtual Human Mark-up Language) [4] is an XML-based language which provides an intuitive way to define virtual human animation. The H-Anim specification describes the geometry and structure of a virtual human, however it doesn't provide a way to specify animation. VHML is composed of several sub-languages: DMML (Dialogue Manager Markup Language), FAML (Facial Animation Markup Language), BAML (Body Animation Markup Language) [5], SML (Speech Markup Language), and EML (Emotion Markup Language). Fig.6A shows a VHML exam-

ple. With this human animation language it will be easy to specify generic animations for virtual humans in a wide variety of applications. Currently VHML is under development. Like other XML-based markup languages, VHML is declarative and require a Java or other XML consumers to really get things done, e.g. taking a Java interpretation of XML-based markup with H-Anim or MPEG-4 formats.

There are several other languages on the third level. STEP [6] is a scripting language for human actions. It has a Prolog-like syntax, which makes it compatible with most standard logic programming languages. The formal semantics of STEP is based on dynamic logic. Fig. 6B is an example of STEP script. RRL (Rich Representation Language) [7] is an SMIL influenced markup language used in the NECA system, which generates interactions between two or more animated characters. RRL focuses on representations of agent behaviour in dialogue and supports the integrated representation of various types of information, even including some linguistic information (e.g. pragmatic, semantic, syntactic, and prosodic). AML (Avatar Markup Language) [8] is a VRML influenced markup language for describing avatar animation. It encapsulates Text To Speech content, Facial Animation and Body Animation in a unified manner with appropriate synchronization information.

D. Natural Language to 3D Animation

The fourth level includes high level animation applications which convert natural language to virtual human animation. Little research on virtual human animation focuses on this level. One of the first projects is the AnimNL project [9] that aims to enable people to use natural language instructions to tell virtual humans what to do. In the next section, we will introduce our CONFUCIUS project [10] which also deals with language animation. We believe that research on this level will lead to powerful web-based applications.

III. CONFUCIUS

We are developing an intelligent multimedia storytelling interpretation and presentation system called CONFUCIUS. It automatically generates 3D animation and speech from natural language input as shown in Fig. 3. The dashed part in the figure is the knowledge base including language knowledge (lexicons and a syntax parser) which is used in the Natural Language Processing (NLP) module, and visual knowledge such as 3D models of characters, props, and animations of actions, which is used in animation engine. The surface transformer takes natural language sentences as input and manipulates surface text. The NLP module uses language knowledge to parse sentences and analyse their semantics. The media allocator then generates an XML-based specification of the desired multimodal presentation and assigns content to three different media: animation, characters' speech, and narration, e.g. it sends the parts bracketed in quotation marks near a communication verb to the text-to-speech engine. The animation engine takes semantic representation and use visual

knowledge to generate 3D animations. The animation engine and Text-to-Speech (TTS) operate in parallel. Their outputs are combined in the synchronizing module, which outputs a holistic 3D virtual world including animation and speech in VRML. Finally the narration integration module integrates the VRML file with the presentation agent, Merlin the Narrator, to complete a multimedia story presentation.

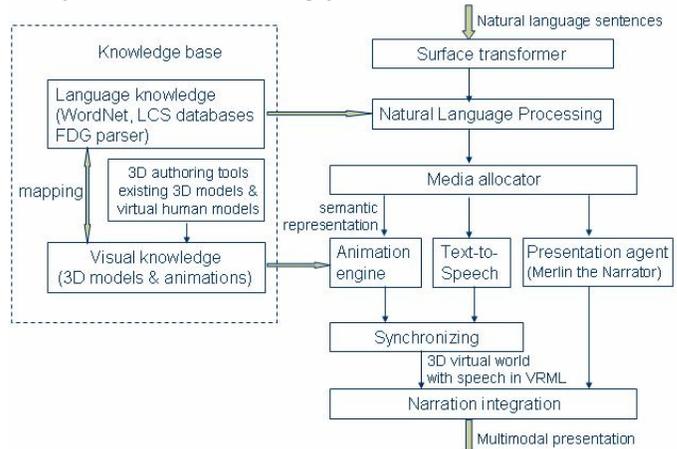


Fig. 3. Architecture of CONFUCIUS

A. Humanoid Animation in CONFUCIUS

Since our task of language animation in CONFUCIUS focuses on off-line generation, and real-time interaction is never our concern, we adopt the H-anim standard to model the virtual characters in our storytelling. H-anim provides four Levels of Articulation (LOA) for applications which require different *levels of detail*. Some applications such as medical simulation and design evaluation require high fidelity to anthropogeometry and human capabilities, whereas games, training and visualized living communities are more concerned with real-time performance. Storytelling is not usually concerned with accurate simulation of humans. We use the Level 2 of Articulation (LOA2) of H-anim in character modelling for CONFUCIUS. This level ensures enough joints for human movements in storytelling, e.g. it includes enough hand joints for grasp postures. Fig. 2 illustrates the joints of LOA2.

Fig. 4 shows the flowchart of the *animation generator*. *Motion decomposition*, *environment placement*, and *camera controller* are optional processes. Motion decomposition translates the EVENTS in semantic representation into a set of simple executable actions which are included in the *animation library*. *Animation controller* then instantiates keyframing information in the animation library to the motion bearer (an object) or a human and schedules the execution of the sequence of basic actions (i.e. timing). Motion instantiation also deals with applying path to the keyframing information of motions. For instance, besides joints' rotation the motion *climb* in "climb a tree" is a vertical upward movement whereas in "climb the mountain" or "climb through the tube" (climb + PP) is a slope upward movement and the slant depends on the surface feature of the object. In the animation library, only the joint rotations of *climb* are defined. The animation controller

needs to add an appropriate *upward* positionInterpolator and rotate the whole body of the character to suit to the slope if necessary.

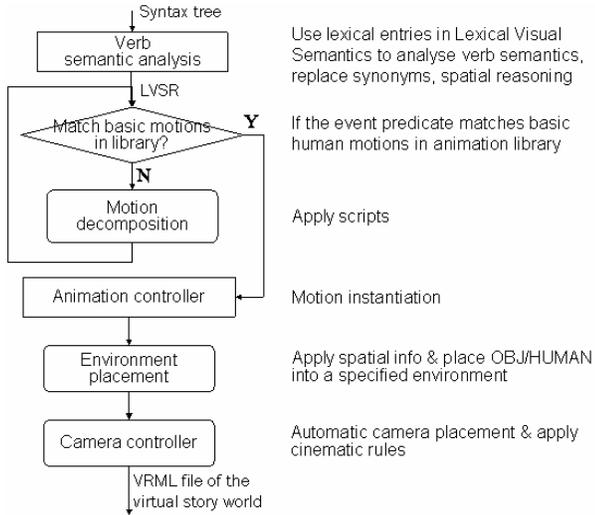


Fig. 4. Flowchart of animation generator

Automated camera placement and control may be useful in virtual environments, since there is no available director or editor in real-time. The *camera controller* uses cinematic rules to place camera and control camera behaviour. Some of the cinematic rules, which describe the relation between the action and the camera behavior, are action-dependent. For instance, the over-the-shoulder shot emerges from conversation rules (see Fig. 5), presenting verbs of communication. Other rules are action-independent, i.e., cinematic constraints on combinations of shots.



Fig. 5. Over-the-shoulder shot for presenting verbs of communication

B. Multiple Animation Channels

Performing simultaneous animations is not a problem for the third level human animation modeling languages, i.e. VHML and STEP in Fig. 1, since they provide a facility to specify both sequential and parallel temporal relations. Fig. 6 shows how VHML and STEP represent the parallel temporal relation. However, simultaneous animations cause the Dining Philosopher's problem for higher level animation using pre-defined animation data, i.e. multiple animations may request to access same body parts at the same time. In order to solve this problem, we introduce the approach of multiple animation channels to control simultaneous animations.

```
<left-calf-flex amount="medium">
<right-calf-flex amount="medium">
  <left-arm-front amount="medium">
  <right-arm-front amount="medium">
Standing on my knees I beg you pardon
  </right-arm-front></left-arm-front>
</right-calf-flex></left-calf-flex>
```

A. A VHML example

```
script(walk_forward_step(Agent),ActionList):-
  ActionList=[parallel(
    [script_action(walk_pose(Agent),
      move(Agent,front,fast))]]).
```

B. A STEP example

Fig. 6. Representing parallel temporal relation

A character that plays only one animation at a time has only a single channel, while a character with upper and lower body channels will have two animations playing at the same time. Multiple animation channels allow characters to run multiple animations at the same time (such as walking with the lower body while waving with the upper body). Multiple animation channels often need to disable one channel when a specific animation is playing on another channel to avoid conflicts with another animation.

We use an animation table as shown in Table 2 to implement multiple animation channels. Every pre-defined animations must register in the animation table and specify which joints are used for the animation. In Table 2, each row represents one animation, and each column represents one joint. 0 indicates that the joint is not used for the animation; 1 indicates that it is used and can be disabled when playing simultaneous animations; and 2 means that the joint is used and cannot be disabled. When simultaneous animations are requested, the animation engine checks the animation table and finds if the involved joints of these animations conflict, i.e. if there is any joint whose values for both animations are 2, these animations conflict and they cannot be played at the same time. If two animations do not conflict (for example, "run" and "throw"), the animation engine merges their keyframes information, i.e. interpolators, and creates a new animation file which will be applied to the virtual character.

TABLE 2.

THE ANIMATION TABLE

Involved joints / Animations	sacroiliac	l_hip	r_hip	...	r_shoulder
walk	2	2	2	...	1
jump	2	2	2	...	1
wave	0	0	0	...	2
run	2	2	2	...	1
scratch head	0	0	0	...	2
sit	2	2	2	...	1
...

C. Space Sites of Virtual Humans

In VRML files, there are a list of grasp sites and their purposes, and intrinsic directions (top, front, etc.) defined with respect to an object, and a list of sites for manipulating and

placing/attaching objects defined with respect to a virtual human. We classify three types of objects as follows:

Small props which are usually manipulated by hands or feet, e.g. cup, box, hat, ball.

Big props which are usually source or targets (goals) of actions, e.g. table, chair, tree.

Stage props which have internal structure, e.g. house, restaurant, chapel.

To figure out where to place these three types of props around virtual human bodies, we create corresponding site tags for virtual humans using H-Anim Site nodes.

1. For manipulation of small props, a virtual human has six sites on the hands (three sites for each hand, `l_metacarpal pha2`, `l_metacarpal pha5`, `l_index_distal_tip`, `r_metacarpal pha2`, `r_metacarpal pha5`, `r_index_distal_tip`), one site on the head (`hanim_skull_tip`), and one site for each foot tip (`l_forefoot_tip`, `r_forefoot_tip`). The sites `metacarpal pha2` are used for grip and pincer grip; `metacarpal pha5` are for pushing; and `index_distal_tip` are for pointing. The sites `forefoot_tip` are for kicking. See Fig. 7 for the position of these sites.

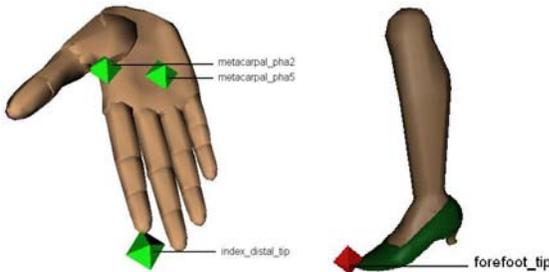


Fig.7. Site nodes on hands and feet (H-Anim)

2. For big props placement, we use five sites indicating five directions around the human body: `x_front`, `x_back`, `x_left`, `x_right`, `x_bottom`. We leave out `x_top` because there is already a site node, `hanim_skull_tip`, defined on the head of every virtual human for attaching headdress. Big props like a table or chairs are usually placed at these positions.

3. For stage props setting, we have five more space tags besides those in (2) around a virtual human to indicate further places: `x_far_front`, `x_far_back`, `x_far_left`, `x_far_right`, `x_far_top`. Fig. 8 shows the positions of these sites. Stage props such as a house often locate at these far sites of virtual humans.

D. Object Manipulation

To manipulate 3D objects, we need to study hand movements and postures. Table 3 lists the classification of hand movements of object manipulation according to physical characteristics, i.e. change effectuated and indirection level, or their function (either prehensile or nonprehensile). Non-prehensile movements include pushing, lifting, tapping and punching.

To illustrates how complex it can be to perform a simple task of hand movement, let's consider the example of picking up a mug: walking to approach the mug, deciding which hand

to use, searching for the graspable site (i.e. the handle), moving body limbs to reach the handle, deciding which hand posture to use, adjusting hand orientation and the approaching aperture, grasping, close the grip, and finally lifting the mug.

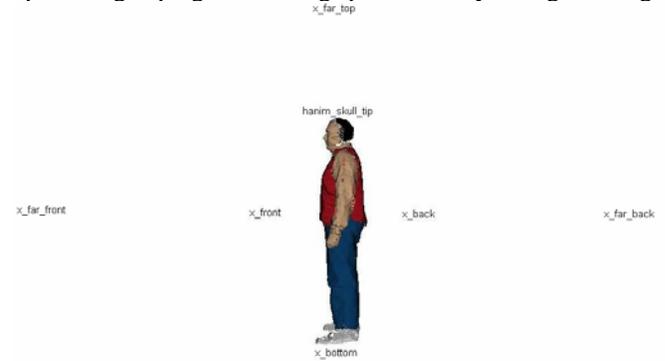


Fig. 8. Site nodes around a virtual human's body

TABLE 3.
CLASSIFICATION OF HAND MOVEMENTS

Classification standards	Physical characteristics	Functions
Classes of hand movements and postures	<ul style="list-style-type: none"> • Change effectuated: position, orientation, shape • How many hands are involved: one or two • Indirection level: direct manipulation or through another object or tool 	<ul style="list-style-type: none"> • Prehensile • Nonprehensile

There are two approaches to organizing the knowledge required in the above task to achieve “intelligence” for successful grasping. One is to store applicable objects in the animation file of an action and using lexical knowledge of nouns to infer hypernymy relations between objects. For instance, one animation file of “pick-up” specifies the applicable objects are cups. The hand posture and movement of picking up a cup are stored in the animation file. From the lexical knowledge of the noun “mug” the system knows that a “mug” is a kind of “cup” and its meronymy¹ relations, and the system then accesses the mug's geometric file to find its grasp site, i.e. the location of the handle. The system then combines the “pick up” animation for a cup object with the virtual human and uses it on the mug.

The other approach includes the manipulation hand postures and movements within the object description, besides its intrinsic object properties. Kallmann and Thalmann [11] call these objects “smart objects” because they have the ability to describe in details their functionality and their possible interactions with virtual humans, and are able to give all the expected low-level manipulation actions. This approach decentralizes the animation control since object interaction information is stored in the objects, and hence most object-specific computation is released from the main animation control. The idea comes from the object-oriented programming

¹ The “parts of” relationship. The meronyms of “mug”, for example, are handle, stem, brim, and base.

paradigm, in the sense that each object encapsulates data and provides methods for data access.

Robotics techniques can be employed for virtual hand simulation of hand movements, as for automatic grasping of geometrical primitives. They suggest three parameters to describe hand movements for grasping: hand position, orientation and grip aperture. We distinguish four stored hand postures and movement (Fig. 9) for moving, touching and interacting with 3D objects: index pointing (Fig. 9A, e.g. press a button), grip (Fig. 9B, e.g. hold cup handle, knob, or a cylinder type object), pincer grip (Fig. 9C, i.e. use thumb and index finger to pick up small objects), and palm push (Fig. 9D, e.g. push big things like a piece of furniture). They use different hand sites to attach objects. Hand postures and movements are defined as the motions of fingers and hands in virtual humans' VRML files. Different kinematic properties, such as movement velocity and grip aperture are fixed since further precision might involve significant costs in terms of processing time and system complexity but the result is only a little more realistic.

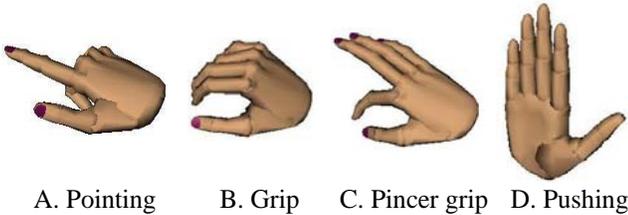


Fig. 9. Four hand postures for physical manipulation of objects

IV. RELATION TO OTHER WORK

A number of projects are currently based on virtual human animation, exploring a variety of applications in different domains such as medical [12], art [13], training and maintenance [14], interface agents [15], and virtual reality [16]. However, few of these systems takes the modern NLP approach that a high level human animation system should be based on. CONFUCIUS is an overall framework of intelligent multimedia storytelling, which makes use of state-of-the-art techniques of 3D modelling and animation with the addition of the natural language understanding technologies to achieve higher level virtual human animation.

V. CONCLUSION

We have investigated current virtual human representation languages and classified them into four levels of abstraction. Virtual human animation generation in the intelligent multimedia storytelling system CONFUCIUS, which creates human character animation from natural language, was described. We have discussed several issues such as multiple animation channels, space sites of virtual characters, and physical manipulation of 3D objects. The value of CONFUCIUS lies in generation of 3D animation from natural language by automating the processes of language parsing, semantic representation and animation production. We believe these techniques have the potential to have an impact on various areas such as com-

puter games, movie/animation production and direction, multimedia presentation, and shared virtual worlds. Future research may address coordination and synchronization of multiple virtual humans.

REFERENCES

- [1] R. W. H. Lau, F. Li, T. L. Kunii, B. Guo, B. Zhang, N. M. Thalmann, S. Kshirsagar, D. Thalmann, M. Gutierrez, "Emerging web graphics standards and technologies", *IEEE Computer Graphics and Applications*, January/February 2003, Vol. 23(1), 66-75.
- [2] H-Anim, Humanoid animation working group, <http://www.h-anim.org>
- [3] SMIL, Synchronized Multimedia Integration Language, <http://www.w3.org/AudioVideo/>
- [4] VHML, Virtual Human Modelling Language, <http://www.vhml.org>
- [5] BAML, Body Animation Markup Language, http://vrlab.epfl.ch/research/S_BAML.PDF
- [6] Z. Huang, A. Eliens and C. Visser, "STEP: A Scripting Language for Embodied Agents", *Proceedings of the Workshop on Lifelike Animated Agents*, Tokyo, 46-51, 2002.
- [7] P. Piwek, B. Krenn, M. Schröder, M. Grice, S. Baumann, H. Pirker, "RRL: A Rich Representation Language for the Description of Agent Behaviour in NECA", Proc. of the AAMAS workshop on "Embodied conversational agents - let's specify and evaluate them!", July 2002, Bologna, Italy.
- [8] S. Kshirsagar, N. M. Thalmann, A. Guye-Vuilleme, D. Thalmann, K. Kamyab, and E. Mamdani, "Avatar Markup Language", *Proceedings of Eurographics Workshop on Virtual Environments*, 2002, 169-177.
- [9] B. Webber, N. Badler, B. Di Eugenio, C. Geib, L. Levison, and M. Moore, "Instructions, intentions and expectations", *Artificial Intelligence Journal*. 73, 253-269, 1995.
- [10] M. Ma and P. Mc Kevitt, "Building character animation for intelligent storytelling with the H-Anim standard". *Eurographics Ireland Chapter Workshop Proceedings 2003*, M. McNeill (Ed.), 9-15, Coleraine, 2003.
- [11] M. Kallmann and D. Thalmann, "Modeling Behaviors of Interactive Objects for Real Time Virtual Environments". *Journal of Visual Languages and Computing*, 13(2):177-195, 2002.
- [12] M. Gutierrez, F. Vexo, D. Thalmann, "Reflex Movements for a Virtual Human: a Biology Inspired Approach", *Proceedings of the 3rd Hellenic Conference on Artificial Intelligence, Special Session on Intelligent Virtual Environments*, May 2004, Samos, Greece, *Lecture Notes in Artificial Intelligence*, Springer Verlag, 525-534.
- [13] J. Esmerado, F. Vexo, D. Thalmann, "Interaction in the Virtual Worlds: Application to Music Performers", *Computer Graphics International*, 2002.
- [14] N. Badler, "Virtual humans for animation, ergonomics, and simulation". In *IEEE Workshop on Non-Rigid and Articulated Motion*, Puerto Rico, June 1997, 28-37.
- [15] J. Cassell, H. Vilhjalmsson and T. Bickmore, "BEAT: the Behavior Expression Animation Toolkit", *Proceedings of Computer Graphics Annual Conference, SIGGRAPH 2001*, Los Angeles, 477-486, August 2001.
- [16] P. Lemoine, F. Vexo and D. Thalmann, "Interaction Techniques: 3D Menu-based Paradigm", *AVIR2003*, Geneva, 2003.