Building character animation for intelligent storytelling with the H-Anim standard

Building character animation for intelligent storytelling with the H-Anim standard

Minhua Ma and Paul Mc Kevitt

School of Computing & Intelligent Systems, Faculty of Informatics, University of Ulster, Magee
Derry/Londonderry, Northern Ireland, BT48 7JL
Email: {m.ma, p.mckevitt}@ulster.ac.uk

Abstract

Here we describe character animation production in the intelligent multimedia storytelling system, CONFUCIUS, which presents natural language stories as 3D animation, speech and non-speech audio. The modelling of animated characters in CONFUCIUS follows the H-Anim standard LOA1. It uses keyframing technology to perform the actions specified in story scripts, and has low-level autonomy such as path finding and collision reaction. Compared with other related virtual human modelling systems, CONFUCIUS' character animation focuses on the language-to-humanoid animation process rather than considering human motion solely. We present the architecture of CONFUCIUS, and explain in detail design and implementation issues with illustratrative examples and the processes of how to get from semantics down to actual human motions.

Keywords: virtual human animation, autonomous agents, virtual environments, visualisation, natural language understanding, language animation, multimodal systems.

1. Introduction

Due to dramatic advances in computer graphics in the last few decades, realizing model-based 3D virtual humanoid agents has become manageable. With the advent of professional 3D authoring software such as Maya, 3D Studio Max and Poser, computer graphic artists can produce high quality character animations. However, animation production is still somehow a time-consuming and tedious task. On the other hand, the last decade has seen the development of human simulation [1], animated interface agents [2], and autonomous characters [13]. The next major step in the evolution of 3D human animation is very likely to be a shift towards highly automatic off-line generation, even real-time production. The purpose of this paper is to discuss essential aspects of off-line generation of virtual human character animation from an Artificial Intelligence perspective.

We first describe the state of the art of virtual human animation, then introduce the background of our work, an intelligent storytelling system—CONFUCIUS in section 2. Next we focus on character animation with the H-Anim standard in CONFUCIUS. Various aspects of off-line generation of human character animation for a virtual story world and language visualisation are discussed in section 3, and relation to other work is discussed in section 4. Finally, we conclude with potential applications and further work in section 5.

2. Background and previous work

There have been a number of virtual human simulation systems [1,2,11,13,16] and approaches varying in function, autonomy, and levels of detail according to different application domains. *Jack* [1] is an interactive system for definition, manipulation, animation, and performance analysis of virtual human figures. Jack is applied in industries to improve the ergonomics of product designs and workplace tasks. It enables users to position biomechanically accurate digital humans of various sizes in virtual environments, assign them tasks and

analyze their performance. The digital human *Jack* can tell what it can see and reach, and can simulate how a human performs specific tasks. Jack focuses on proper task planning and biomechanical simulation and its general goal is to produce accurate simulations of biomechanical robots. Applications in industrial design determined that Jack has too much accuracy and includes far more detail on human body modelling than what is necessary for general-purpose virtual agent applications.

Improv [13] uses script language to produce virtual actors' activities and applies them to drama performance. It consists of an Animation Engine which uses procedural techniques to generate layered, continuous motions and transitions between them, and a rule-based *Behaviour Engine* which governs how actors communicate with each other and make decisions reacting to the ever-changing environment. The *SimHuman* System of Vosinakis and Panayiotopoulos [16] also generates virtual agents as an autonomous entity in a virtual environment. *BEAT* [2] allows animators to input typed text that they wish to be spoken by an animated human figure. It can realize written text (dialogues) in embodied expressive verbal and nonverbal behaviors such as face expression, head nods, gaze, and hand gestures on 3D virtual agents.

WordsEye [3] is a text-to-scene conversion system which converts text into representative 3D scenes automatically. It relies on a large library of 3D models and poses to depict entities and actions. Every 3D model can have associated shape displacements, spatial tags, and functional properties to be used in the depiction process. WordsEye generates *static* 3D scenes rather than animation. Hence it focuses on the issues of semantics and graphical representation without addressing all the problems inherent in automatically generating animation.

Much research has been conducted in the field of virtual human modelling, motion and behavior. Some models such as Jack are very advanced in a narrow area (e.g. biomechanical robot simulation) but lack other desirable features such as real-time communication. Here we describe a general-purpose human character animation system included in an intelligent multimedia storytelling system called CONFUCIUS.

2.1. CONFUCIUS

CONFUCIUS is an intelligent multimedia storytelling interpretation and presentation system which automatically generates multimedia presentations from natural language input. The storytelling employs several temporal media such as animation, speech and non-speech audio for the presentation of short stories. Establishing correspondence between language and animation is the focus of this research. This requires adequate representation and reasoning about the dynamic aspects of the story world, especially about events. Modelling humanoid motion is one major task. Figure 1 shows the architecture of CONFUCIUS. The knowledge base on the left of figure 1 includes prefabricated models such as characters, props, and animations for basic actions, which will be used in the *Animation generation* module. When the input is a story, it will be transferred to drama-script like text by the *script writer*, then be parsed by the *script parser* and the *natural language processing* module respectively. The three modules of *Natural Language Processing* (NLP), *Text to Speech* (TTS) and *sound effects* operate in parallel. Their outputs will be synchronized at the *fusion module*, which generates a holistic 3D world representation including animation, speech and sound effects. NLP is performed using PC-PARSE [10] and WordNet [5], TTS is performed using Festival [15], VRML (Virtual Reality Modelling Language) is used to model the story 3D virtual world, and visual semantics is represented using a LCS-like (Lexical-Conceptual Semantics) formalism [8].

Figure 2 shows the knowledge base of CONFUCIUS, which is also a general knowledge base design suitable for any intelligent multimedia applications including both natural language processing and vision processing. This knowledge consists of language knowledge, visual knowledge, world knowledge and spatial & qualitative reasoning knowledge. Language knowledge is used in the natural language processing component to extract concept semantics from text. Visual knowledge consists of the information required to generate 3D animation. It consists of *object model, functional information, event model, internal coordinate axes*, and *associations between objects*. The *Object model* includes visual representation of the ontological category (or conceptual "parts of speech") – things (nouns), the *event model* includes visual

representation of events (verbs). It contains explicit knowledge about the decomposition of higher level complex actions into basic motions. The *internal coordinate axes* are indispensable in some primitive actions of *event models* such as rotating operations, which require spatial reasoning based on the object's internal axes. The *event model* in visual knowledge requires access to other parts of visual knowledge. For instance, in the event "he cut the cake", the verb "cut" concerns kinematical knowledge of the subject–a person, i.e. the movement of his hand, wrist, and forearm, hence it needs access to the *object model* of a man who performs the action "cut", and it also needs *function information* of "knife", the *internal coordinate axes* information of "knife" and "cake" to decide the direction of the "cut" movement.

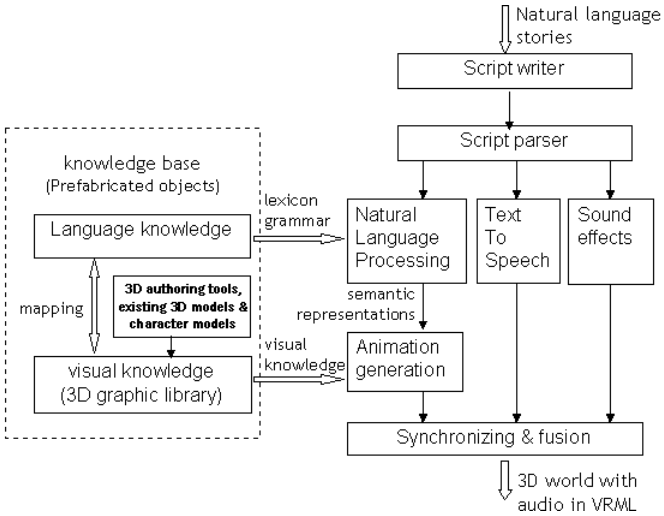


Figure 1: System architecture of CONFUCIUS

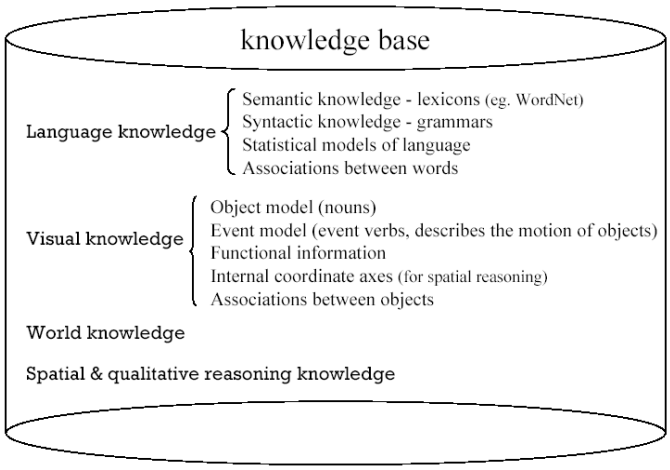


Figure 2: The knowledge base of CONFUCIUS

3. Character animation in a virtual story world

In order to implement a generic human character model, we use highly portable standard technologies in the implementation: we use Java for the control modules, VRML for modelling the geometric of objects, props and environment (e.g. buildings) [4], and H-Anim [7] for the definition of humanoid skeleton hierarchy and articulated features. Here we focus on humanoid modelling and motions.

Currently there are two standards in modelling virtual humans: H-Anim and MPEG 4 SNHC. Based on the VRML97 specification, H-Anim represents humanoids and allows humanoids, created using authoring tools from one vendor, to be animated using tools from another. H-Anim humanoids can be animated using keyframing, inverse kinematics (IK), performance animation systems and other animation techniques. H-anim defines standard human *Joints* articulation, *Segments* dimensions, and *Sites* for "end effector" in IK and attachment points for clothing. MPEG 4 SNHC standard offers an appropriate framework for 3D virtual human animation, gesture synthesis and efficient compression/transmission of these animations. The standard provides tools for the efficient compression of the animation parameters associated with the H-Anim articulated human model [14].

Since our task of virtual human animation in CONFUCIUS focuses on off-line generation which does not concern real-time interaction/ communication via the web, we adopt the H-anim standard to model the virtual characters in our stories. The H-Anim standard not only provides realistic geometry modelling of human size, movement capabilities, and joint limits, but more importantly it also allows us to separate action animations from action performers, and to apply the pre-defined animations in the library to any humanoid models with H-Anim, i.e. any humanoid that is built to the same joint hierarchy and dimensions will be able to share animations.

Most types of humanoid animation are independent of the body's actual dimensions. For example, walking, tilting the head to a specific angle or waving a hand have the same effect on any humanoid that has a skullbase *Joint*. However, some animations may be dependent on the lengths of individual segments or on the ratios of the segment lengths. For example, scratching one's head will

require knowledge of the arm's dimensions. Humanoids that are sized differently, but which use the same ratios of segment lengths, may also be able to share certain animations provided that the application adjusts the animation values accordingly.

H-anim 1.1 provides four LOA (Levels of Articulation) 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. So we use the Level 1 of Articulation (LOA1) of H-anim in character modelling for CONFUCIUS. Figure 3 illustrates the joints and segments of LOA1. The diamonds (with black text) denote joints, and lines (with blue italic text) denote segments.

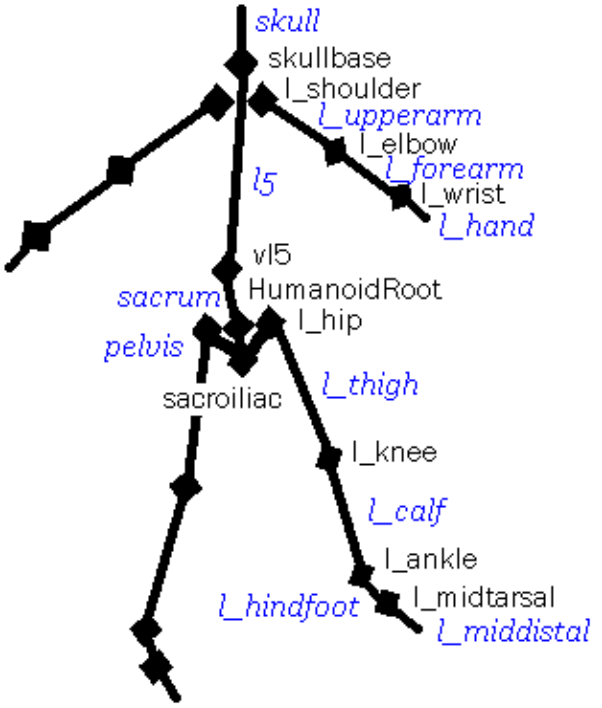


Figure 3: Joints and segments of LOA1

3.1. Agents and avatars – how much autonomy?

Virtual humans can be classified into two types based on their autonomy: autonomous agents and avatars. Autonomous agents have higher requirements for behavior control, sensing, memory, reasoning, and planning, whereas avatars require fewer autonomous actions since their behavior is "player-controlled". Therefore, an autonomous agent's behaviors are usually built on a sense-control-action structure that permits reactive behaviors to be locally adaptive to the environment context. With these agents virtual humans are beginning to exhibit the early stages of autonomy and intelligence as they react and make decisions in changing environments rather than carrying on fixed movements regardless of environment context.

A virtual character in a storytelling system is somewhere in between an autonomous agent and an avatar. Because most of its behaviors and responses to the changing environment are described in story input, it does not require sophisticated reasoning and behavior control; and the character neither receives world information from his own sensors nor calculates and saves perceptual status in his memory, since the world information of the virtual story world is already available in the system. However, simple behavior planning on path plans, e.g. to avoid obstacles, is still required, whereas it is not needed in avatars. This planning is performed by a controlling model rather than by the character itself. Therefore, the autonomy requirement of characters in a virtual story world is greater than that for avatars and less than that for agents.

3.2. Semantic representation

Using natural language description to generate animation is a tendency in Intelligent MultiMedia research [3]. To link language with dynamic visual information, a conceptual semantic representation is demanded. Here, we propose a semantic representation based on Jackendoff's [8] Lexical-Conceptual Semantics (LCS). We identify six ontological categories of concepts: EVENT, STATE, HUMAN, OBJ, PATH, and PLACE. HUMAN could be either human being or any other articulated animated characters (e.g. animals) as long as their skeleton hierarchy is defined in the graphic library. OBJ could be props or places (e.g. buildings). Table 1 shows the definition of PATH in conformity with Jackendoff's classification. In direction feature, 1 denotes the direction of *approaching*, and 0 denotes *leaving*.

The input of the animation generator takes a form which combines Jackendoff's LCS and Ma and Mc Kevitt's [9] decomposite predicate-argument representation. A complex action might be decomposed to a sequence of basic motions (the

predefined animations) and/or utterances of the character (lip movements with synthesized speech). Figure 4 gives three examples of this semantic representation.

PATH	*Direction feature*	*Termination feature*
to	1	1
from	0	1
toward	1	0
away_from	0	0
via	n/a	0

Table 1: The definition of PATH

Example 1: *Nancy ran across the field.*
LCS representation: [EVENT run [HUMAN nancy] [PATH via [PLACE on [OBJ field]]]]

Example 2: *John lifted his hat.*
LCS representation: [EVENT lift [HUMAN john] [OBJ hat]]

Example 3: *John called Nancy.* (make a phone call)
LCS representation: [EVENT call_phone [HUMAN john] [HUMAN nancy]]
LCS representation after first decomposition:
[EVENT pickup [HUMAN john] [EVENT move [OBJ tel.receiver] [PATH to [PLACE beside [OBJ john.skull[1]]]]]]
[EVENT dial [HUMAN john] [OBJ tel.keypad]]
[EVENT speak [HUMAN john] [OBJ tel.receiver]]
[EVENT putdown [HUMAN john] [OBJ tel.receiver] [EVENT move [OBJ tel.receiver] [PATH to [PLACE on [OBJ tel.set]]]]]

Figure 4: Examples of LCS representation and motion decomposition

3.3. Animation generation

The 3D graphic library in Figure 1 consists of the animation library, simple geometry files and H-Anim files. Figure 5 illustrates the composition of the 3D graphic library. *Animation library* defines a set of basic animations, such as "walk", "run", "jump", "crouch", by determining corresponding orientation and position interpolators for the rotations and movements of joints and body parts involved. Object/prop models can be found in *simple geometry files*, and character models are defined in *geometry & joint hierarchy files* following the H-Anim specification. In the flowchart of the *animation generator* (Figure 6), *motion decomposition* and *environment placement* are optional processes. Motion decomposition translates the EVENTs in LCS 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 OBJ) or agent (a HUMAN) and schedules the execution of the sequence of basic actions (i.e. timing).

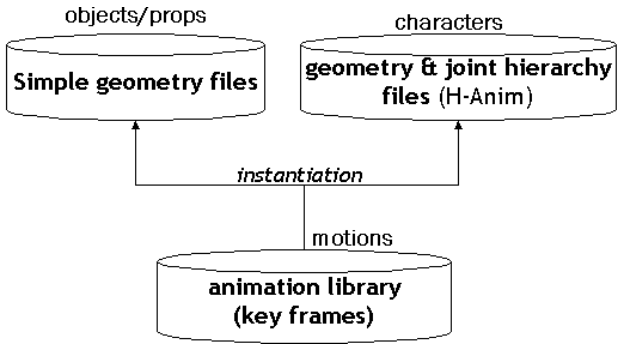


Figure 5: Composition of the graphic library

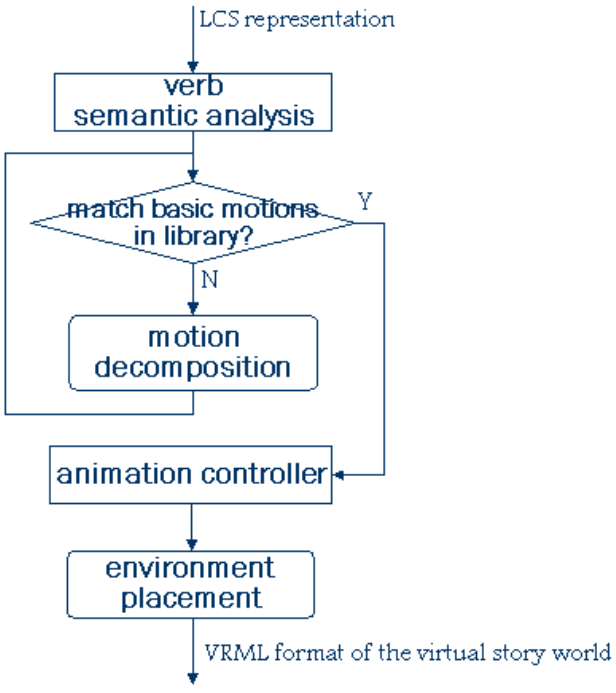
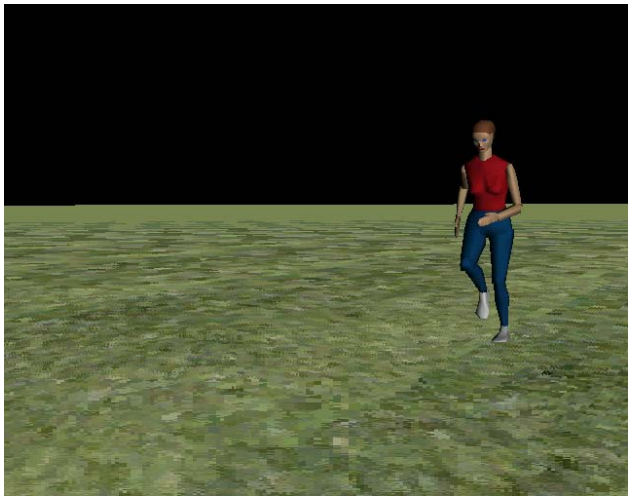


Figure 6: Flowchart of animation generator

[1] See the "skull" segment in H-Anim LOA1 structure in Figure 3.

Figure 7 shows visualisation of the sentence "Nancy ran across the field" (example 1 in Figure 4). Figure 7a is the 3D model of Nancy[2] in *geometry & joint hierarchy files* (see Figure 5), and Figure 7b is the output after instantiating Nancy to the `run` animation in *animation library* (see Figure 5) and placing this in a `field` environment. Because it is a basic action predefined in the animation library, the action `run` need not be decomposed further.

a. 3D model of Nancy



b. Visualisation of "Nancy ran across the field."

Figure 7: "Nancy ran across the field."

[2] Credit: Nancy V1.2b, by C. Ballreich, © 1997 3Name3D / Yglesias, Wallock, Divekar, Inc.

3.4. Collision detection and path planning

When we design the layout of the virtual story world, i.e. the placement of props, and the movement of actors, besides considering their size and position (spatial relationship) one inevitable problem we may encounter is *collision detection* -- a 'naïve' physical problem about how to examine collision/intersection of objects and actors. Although VRML provides a built-in collision detection mechanism for the avatar (user), the mechanism does not apply to other objects. However, a large proportion of collisions occur among objects and actors' motions, especially where actors are not the first person (non avatar).

Most collision avoidance algorithms use coarse approximations, such as bounding boxes or spheres, to detect possible future collisions. There are two possible implementations in VRML. The first is to write up scripts detecting intersection between the bounding boxes/spheres of objects/characters. This requires 3D translation calculation especially when the objects detected are moving (rotating). The other is to bind a viewpoint with object geometry temporarily in order to detect collisions, therefore we may utilize automatic collision detection for the avatar (the active viewpoint) by using Collision node and setting `avatarSize` in `NavigationInfo` Node meanwhile. However, when detecting collisions concerns the protagonists, those characters which are observed more closely and get more attention, more accurate collision handling is needed. Real people may bump up against each other in an enclosed area, hold hands, or grasp objects. At the extreme, there are computationally expensive approaches for highly detailed collision detection with virtual humans, performing polygon level checks between humans and objects in the scene, e.g. for cloth simulation and contact modelling.

In order to adapt to different requirements in both path planning and contact modelling, and give a computationally economical solution, CONFUCIUS' animation generator uses bounding cylinders around the human body segments for protagonists and a bounding cylinder around the whole human body for minor characters and characters beyond the scope of attention, and performs an approximate collision detection with them.

Collision detection is also a crucial issue in characters' maneuvering of objects and multiple

characters' activities. Grasping an object is an ordinary movement of a virtual human. The character should be able to reach for the object, check the object-specific information about its graspable site, position his/her hand on the proper approach direction, detect a collision/connection of his hand and the object, close his hand, and finally attach the object to his hand's *Site*[3].

3.5. Multiple characters synchronization and coordination

It is very likely that a story scene concerns not just one major character (the protagonist) but other minor characters who may communicate, react, or coordinate with the protagonist. Modelling multiple characters raises the requirements of synchronization and coordination, especially in a multimodal situation. A character can start a task when another signals that the situation (pre-conditions) is ready, characters can communicate with one another, or two or more characters can cooperate in a shared task. These multiple characters' activities concern many issues such as action synchronization and collision detection. An event-driven timing mechanism is used in CONFUCIUS' animation engine for action synchronization because VRML provides a utility for event routing (ROUTE node). For instance, in the modelling of the following event between two actors Nancy and John:

```
Nancy was walking along the street. John
called her. Nancy stopped and saw John.
John walked towards her. They exchanged
greetings.
```

the end of the animation `john_speech` (calling Nancy) triggers three events: (1) to stop the animation of `nancy_walk`; (2) to start the animation of `nancy_gazeWander` (searching for who's calling) and (3) to start the animation of `john_walk` (walking towards Nancy). Moreover, in a dialogue between two characters one's facial expressions or gestures may trigger the other's response such as addresser's raising eyebrow triggers addressee's nodding. It is obvious that event-driven synchronization is more natural and also convenient than an exact time-driven mechanism.

[3] A *Site* node in H-anim defines an attachment point for accessories such as jewelry, clothing, and other objects.

3.6. Efficiency issue -- Imposters of objects and humans

Billboards (sprites) have the advantage of being very cheap to render, but real 3D geometry has the advantage of looking realistic. The *impostoring* technique can somehow have both advantages. An impostor is essentially a sprite -- a 2D image that replaces a real 3D model. Impostors are rendered at run-time by the 3D engine itself. For example, if a 3D scene contains a tree that will be represented by an impostor, the 3D engine first renders the 3D tree model to a texture, then places that texture in the scene as a billboard. As long as the camera doesn't move too much, the impostor can be kept as it is. As the camera moves significantly, the engine simply re-renders the impostor.

This technique is computationally economical for rendering environments (e.g. buildings and objects in background) because the 3D engine renders an object once, and reuses it in several subsequent frames. This can obviously reduce polygon transformation and improve the performance. O'Sullivan et al. [12] also applied it to render minor characters, or characters out of the scope of attention in crowds and groups simulation. Although real-time performance is not essential for CONFUCIUS' offline animation generation, this issue is taken into consideration since the system may be expected to run on a low-end platform such as a PC with an Intel processor rather than a graphics workstation. Moreover, it also gives CONFUCIUS the potential to be extended to real-time animation production in the future.

4. Relation to other work

The field of virtual human modelling and animation is an area of continuous research and various different approaches have been proposed in the related work described in section 2. In CONFUCIUS' animation generation we model general-purpose human actors and their behaviours which follow the industry standard (H-Anim) and balance between computational efficiency and accuracy to produce believable human motions.

In addition, since our task is to produce neither agents nor avatars, but virtual actors whose behaviours are tightly coupled with verbal behaviours in story narratives, it requires more reasoning and inferences in natural language

understanding than agent autonomy. For example, to infer that the preconditions of the action "give" in the sentence "A gives x to B" are: (1) "A has x" (possession) and (2) "A is at B" (location). Halliday [6] distinguishes six types of processes involved in the sentences which describe human actors' performance: *doing* (actor and goal, implying a change of state), *sensing* (mental processes like perception, cognition and affection), *being* (attributes like "be happy", "be sad"), *behaving* (physiological behaviours), *saying* (addresser, addressee, and verbiage), and *existing* (existent and circumstance). Halliday's classification can be instrumental for generating humanoid animation from natural language in CONFUCIUS, because compared with those related systems in section 2, CONFUCIUS' character animation focuses on the language-to-humanoid animation process rather than considering human motion solely.

5. Conclusion and further work

We have described virtual human animation generation in an intelligent multimedia storytelling system called CONFUCIUS that creates human character animation from natural language, and the processes of how to get from semantics *down* to actual human motions. Implementation issues such as collision detection, autonomy, multiple characters synchronization and coordination are discussed. As development of CONFUCIUS is ongoing, some important issues like the facial expressions of virtual characters, underspecification problem, i.e. underspecified semantic representations for ambiguous or vague linguistic expressions, in language visualisation, and action compositing for simultaneous activities are not covered in this paper. 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 computer games, movie/drama production and direction, multimedia presentation, and shared virtual worlds.

References

1. Badler, N. Virtual humans for animation, ergonomics, and simulation. In *IEEE Workshop on Non-Rigid and Articulated Motion*, Puerto Rico, June 1997.

2. Cassell, J., H. Vilhjalmsson and T. Bickmore. BEAT: the Behavior Expression Animation Toolkit, Computer Graphics Annual Conference, *SIGGRAPH 2001 Conference Proceedings*, Los Angeles, 477-486. 12-17 August 2001.

3. Coyne, B and R. Sproat. WordsEye: An Automatic Text-to-Scene Conversion System. Computer Graphics Annual Conference, *SIGGRAPH 2001 Conference Proceedings*, Los Angeles, 487-496. 12-17 August 2001.

4. Dodd, A., M. Riding and N.W. John. Building realistic virtual environments using Java and VRML. In *Eurographics Irish Chapter Workshop Proceedings 2002*, Dublin, 53-62. 25-26 March 2002.

5. Fellbaum, C. (Ed.) *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press. 1998.

6. Halliday, M.A. *An Introduction to Functional Grammar*. London: Edward Arnold. 1994.

7. H-Anim. Humanoid animation working group. http://www.h-anim.org 2001.

8. Jackendoff, R. *Semantic Structures*. Current studies in linguistics series, Cambridge, MA: MIT Press. 1990.

9. Ma, Minhua and P. Mc Kevitt. Semantic representation of events in 3D animation. In *Proceedings of The Fifth International Workshop on Computational Semantics* (IWCS-5), H. Bunt, I. van der Sluis and R. Morante (Eds.), 253-281, Tilburg, The Netherlands, 15-17 January 2003.

10. McConnel, S. KTEXT and PC-PATR: Unification based tools for computer aided adaptation. In H. A. Black, A. Buseman, D. Payne and G. F. Simons (Eds.), *Proceedings of the 1996 general CARLA conference*, 39-95. Waxhaw, NC/Dallas: JAARS and Summer Institute of Linguistics. 14-15 November 1996.

11. Milde, J. Lokutor: towards a believable communicative agent. In *Proceedings of Agents 2000 Workshop*, Barcelona, 2000.

12. O'Sullivan, C., J. Cassell, H. Vilhjalmsson, J. Dingliana, S. Dobbyn, B. McNamee, C. Peter and T. Giang. Levels of detail for crowds and groups. In

Computer Graphics Forum 21(4), D. Duke and R. Scopigno (Eds.), 733-742. 2002.

13. Perlin, K. and A. Goldberg. Improv: a system for scripting interactive actors in virtual worlds. In *SIGRAPH'96 Conference Proceeding*, 205-216. 1996.

14. Preda, M., T. Zaharia and F. Preteux. 3D Body Animation and Coding within a MPEG-4 Compliant. In *International Workshop on Synthetic – Natural Hybrid Coding & Three Dimensional Imaging*, 15-17 September, Santorini, Greece, 1999.

15. Taylor, P., Black, A. and Caley, R. The architecture of the Festival Speech Synthesis system. In *Proceedings 3rd ESCA Workshop on Speech Synthesis*, 147-151, Jenolan Caves, Australia, 1998.

16. Vosinakis, S. and T. Panayiotopoulos. SimHuman: a platform for real-time virtual agents with planning capabilities. In *IVA 2001*, LNAI 2190, A. de Antonio, R. Aylett and D. Ballin (Eds.), 210-223. Berlin Heidelberg: Springer-Verlag. 2001.