A FRAMEWORK FOR H-ANIM SUPPORT IN NVES

Ch. Bouras[1,2]
[1] *Research Academic Computer Technology Institute (CTI), Greece*
[2] *Computer Engineering and Informatics Department (CEID), University of Patras, Greece*

K. Chatziprimou
Computer Engineering and Informatics Departmet (CEID), University of Patras, Greece

V. Triglianos[3,4]
[3] *Research Academic Computer Technology Institute (CTI), Greece and*
[4] *Computer Engineering and Informatics Departmet (CEID) University of Patras, Greece*

Th. Tsiatsos[5,6]
[5] *Department of Informatics, Aristotle University of Thessaloniki, Greece*
[6] *Research Academic Computer Technology Institute (CTI), Greece*

Keywords: Virtual reality, H-Anim, Humanoid Animation, Animation, Networked Virtual Environments, Multimedia systems, Architecture, and Applications.

Abstract: Many applications of Networked Virtual Environment presuppose the users' representation by humanoid avatars that are able to perform animations such as gestures and mimics. Example applications are computer supported collaborative work or e-learning applications. Furthermore, there is a need for a flexible and easy way to integrate humanoid standardized avatars in many different NVE platforms. Thus, the main aim of this paper is to introduce a procedure of adding animations to an avatar that is complying with H-Anim standard and to present a standardized way to integrate avatars among various NVEs. More specifically this paper presents a framework for adding/loading custom avatars to an NVE and applying to them a set of predefined animations, using H-Anim standard.

1 INTRODUCTION

Currently, a large number of Networked Virtual Environments (NVEs) have made their way to the public. Users are represented in NVEs by entities called avatars. Using virtual humans to represent participants promotes realism in NVEs (Capin et. al, 1997). Moreover, users tend to develop a psychological bond with their avatar/s. One of the main tasks in a NVE is the integration of humanoid avatars and humanoid animations on top of them. This could be useful in order for the users to transfer their avatars from an NVE to another. In many Networked Virtual Environments, such as Massively Multi-User Online Role-Playing Games, users are given a large degree of control over the appearance of their avatars (Yee, 2006). Usually the users exploit this functionality to customize their avatars in order to reflect their real or virtual personality.

The creation and integration of virtual humans that are compatible in many different NVE platforms is a challenging task for the following reasons: (a) It requires the creation of a humanoid avatar based on a standardized way; (b) The creation of humanoid animation is a complex task which usually requires particular skills and training (Buttussi et al., 2006); and (c) there is no

Bouras C., Chatziprimou K., Triglianos V. and Tsiatsos T. (2009).
A FRAMEWORK FOR H-ANIM SUPPORT IN NVES .
In *Proceedings of the Fourth International Conference on Computer Graphics Theory and Applications*, pages 286-291

standardised way to integrate and share humanoid avatars in a networked virtual environment.

The first challenge has been resolved with the H-Anim standard (Humanoid Animation Working Group, 2004), now included in X3D, which describes humanoids as an hierarchically organized set of nodes. Furthermore, Ieronutti and Chittaro (2005) proposed Virtual Human Architecture (VHA), which is an architecture that integrates the kinematic, physical and behavioral aspects to control H-Anim virtual humans. This solution is fully compatible with Web standards and it allows the developer to easily augment X3D/VRML worlds with interactive H-Anim virtual humans whose behavior is based on the Sense-Decide-Act paradigm (represented through HSMs).

Concerning the animation (which is the second challenge) there are many tools (free of charge or commercial) which simplify this process by visual authoring tools. Buttussi et al., (2006), present such a tool called H-Animator as well as an overview of other visual authoring tools. However, the main problem is the third challenge because there is no standardized way to integrate and share humanoid avatars in a networked virtual environment. Miller (2000) achieved to provide an interface to aggregate and control articulated humans in a networked virtual environment by addressing the following areas:

- The creation of an articulated joint structure of virtual human avatars and a limited motion library in order to model realistic movement. More specifically Miller (2000) achieved rapid content creation of human entities through the development of a native tag set for the Humanoid Animation (H-Anim) 1.1 Specification in Extensible 3D (X3D).
- The development and implementation of a set of rule-based physical and logical behaviors for groups of humans in order to execute basic tactical formations and activities.
- The aggregation of human entities into a group or mounting of other human entities (such as vehicles) and then separation back to individual entity control. Otherwise, the high-precision relative motion needed for group activities is not possible across network delays or in geo-referenced locations.

Even though Miller's work is based on standards it does not solve the problem of dynamic avatar creation and change. Based on the above, it is obvious that there is an need for a flexible and easy way to integrate humanoid standardized avatars in many different NVE platforms. Rapid application development process could help on this direction by utilizing reusable frameworks and API's. This work deals with two important challenges in the field of NVEs. The first is the procedure of adding animations to an avatar that is complying with a standard, in our case H-Anim. The second is to introduce a standardized way to integrate avatars among various NVEs. Our contribution focuses on providing a framework that allows a user to upload an avatar to an NVE, to automatically add predefined animations to it and integrate it to the environment. All these steps could happen at execution time. More specifically, in this paper we present a framework that allows any X3D compliant platform to import an H-Anim compliant avatar, to add a set of custom animations to it and finally, to add the avatar to the virtual environment.

This work is structured as follows. The next section is an overview of an NVE Platform called EVE that is the platform we used to integrate and test our framework. Afterwards, the architecture of the proposed solution is presented. The fourth section describes the process of integrating the framework to an NVE. The fifth chapter illustrates practical examples concerning the practical exploitation the proposed framework. The final section presents the concluding remarks and our vision for the next steps.

2 EVE OVERVIEW

Even though the proposed H-Anim integration framework aimed to support any NVE platform based on X3D standard, we are presenting it through its integration in EVE (http://ouranos.ceid.upatras.gr/vr) networked virtual environments platform (Bouras et al, 2006).

Thus, it is essential to present the main characteristics of this platform. EVE is based on open technologies (i.e. Java and X3D).

It features a client – multi-server architecture with a modular structure that allows new functionality to be added with minimal effort. Initially, it provides a full set of functionalities for e-learning applications and services, such as avatar representation, avatar gestures, content sharing, brainstorming, chatting etc.

Furthermore, the current version of the platform supports collaborative design applications. Currently, the architecture of the platform consists of five servers as shown in Figure 1.

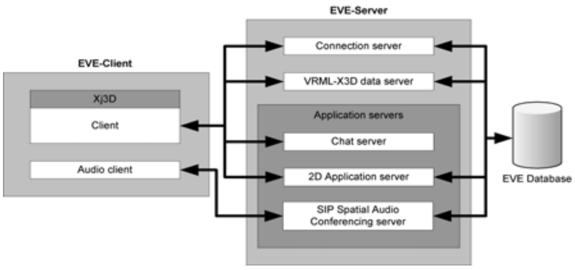

Figure 1: EVE Architecture.

The "Connection Server" coordinates the operation of the other servers. The "VRML-X3D Server" is responsible for sending the 3D content to the clients as well as for managing the virtual worlds and the events that occur in them. The "Chat Server" supports the text chat communication among the participants of the virtual environments.

The "SIP Spatial Audio Conferencing Server" is used in order to manage audio streams from the clients and to support spatial audio conferencing. Finally the "2D Application Server" is handling the generic (i.e. non - X3D) events.

On the client side, the interface of the platform is rendered by a Java applet that incorporates an X3D browser (based on Xj3D API). The following section presents the architecture of the new H-Anim component integrated in EVE platform and describes its operation.

3 H-ANIM COMPONENT ARCHITECTURE

In this section the architecture behind the framework for H-Anim support in NVEs is presented.

The "H-Anim 200X" standard has been chosen for the representation of humanoid models in EVE platform. Originally created to enable the design and exchange of humanoids in virtual online environments, the H-anim standard offers a suitable data structure for the real time animation. The main features of H-anim standard are the following:

- Support of VRML, XML and X3D data formats.
- Support of five (5) types of nodes: Humanoid, Joint, Segment, Site and Displacer nodes.
- Different levels of complexity are suggested. These suggestions are called "Levels Of Articulation" of the skeleton (LOA). There are three of them (excluding the trivial, zero LOA), varying in number of joints and sites needed.
- Objects that are not part of the anatomy can still be specified within the H-Anim humanoid data structure.
- "Skinning" the avatar is made possible by a dual specification of the body geometry, which is divided into the skeletal and the skinned part.

Moreover, the anatomy is specified through a tree-like hierarchical structure of H-Anim specific nodes, which is referred to as the skeletal body geometry specification.

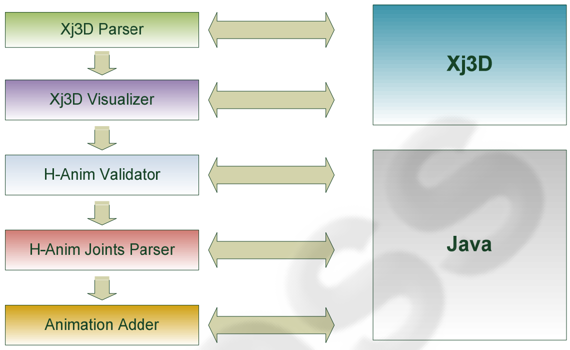

Figure 2: Architecture Components.

As long as a client (i.e. a user) wishes to connect to the NVE, using her/his own avatar and utilize the supplied animations, her/his humanoid has to be H-Anim compliant. The overview of the architecture of the proposed H-Anim component is presented in Figure 2.

The following paragraphs describe:

- the mechanism for inserting an H-Anim avatar in an NVE platform along with its animations.
- the main parts of the H-Anim component (i.e. the parser and the user interface).

3.1 Mechanism

The operation of H-Anim component presupposes the availability of a VRML/X3D file that contains the avatar as well as a set of VRML /X3D animations.

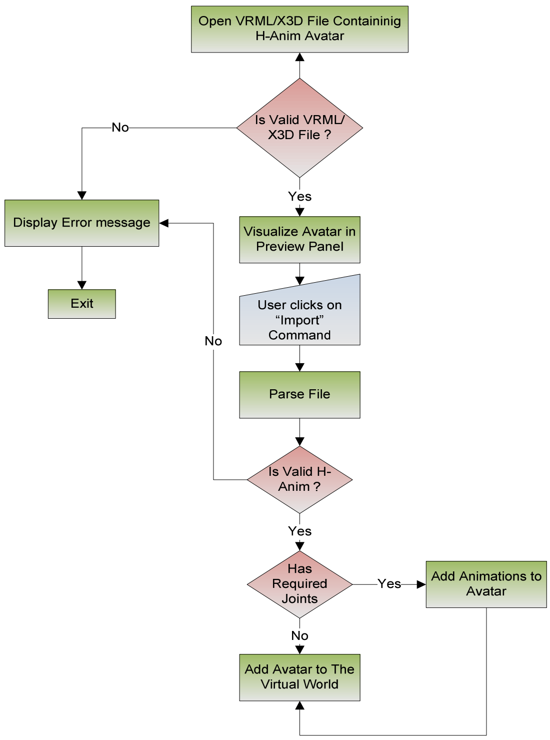

Figure 3: Flowchart of the import process.

Based on these necessary inputs, the operation of the H-Anim component is based on the following steps (Figure 3):

- File submission and opening: Reading of the VRML/X3D file that contains the avatar.
- Validation of X3D/VRML compliance: Validation of the above file. If it's a valid X3D/VRML file, visualize it on a *Xj3D Visualization Panel*. If not a response describing the problem is displayed and the user can try again by restarting this process.
- File insertion: Using the *Xj3D Visualization Panel* the user can import the already validated (against VRML/X3D) avatar file in order to perceive its Level Of Articulation (LOA). If LOA is equal or greater than 0 then we move to the next step. If this is not the case then a message describing the problem is displayed. The user can try to upload another file.
- Animation readiness validation: Check the file to find out if the joints that are required by the list of animations that we want to append, are present. If they are, then the animations are added to the avatar. If not then we simply import dynamically the avatar to the NVE. In the latter case a message informs the user that although his/her avatar was imported to the platform, the animations he chose were not added to the avatar due to joints incompatibility.
- Integration of animations: Finally, when all of the above controls are completed, the Animation Adder can be utilized. Choosing the option "Add Animation" from the main menu, a dialog frame pops up, and the user can select, among a list of animations, the animation he wishes to append to his avatar.

3.2 The Main Parts of the H-Anim Component

The framework's architecture mechanism consists of five main parts: the *X3D/VRML Parser*, the *Xj3D Visualizer*, the *H-Anim Validator*, *H-Anim Joints Parser* and finally the *Animation Adder*. The following paragraphs describe each one of the above respectively.

The *X3D/VRML Parser*, based on Xj3D API, which provides both syntax and semantic level checking, is given a URL that represents the location of a X3D/VRML file, making it transparently useful as either part of a web browser or standalone client. As the parser carries out the VRML/X3D grammar, a set of VRML/X3D nodes is generated.

Once the parsing of the client's file is correctly accomplished, the content is displayed in the *Xj3D Visualizer*. The interface that hosts the Xj3D Visualizer, provides an option menu wherein the user can browse a file in his computer system, and thereafter submit it to *H-Anim Validator*. As long as the selected file is validated as H-Anim compliant, the client is enabled to interact with a dialog menu, choose the desired animation and therefore apply it to his own avatar.

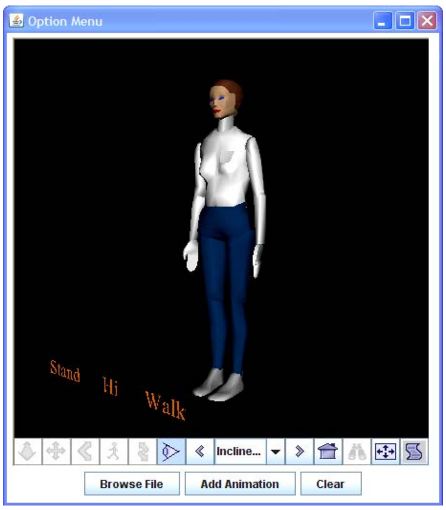

Figure 4: User Interface main menu.

In particular, the *H-Anim Joints Parser* takes as input the location of the user's avatar file, and examines if there exists an hierarchical structure of H-Anim specific joints. The *H-Anim Joints Parser* searches for the joints required by the selected animation. Such a check is crucial in order for the desired animation to be compatible with the offered 3D avatar. As has been already mentioned before, the skeleton of the avatar consists of a sequence of mutually alternating joints and segments (bones). By that way, a graph is formed, where joints represent vertices and segment stand for edges. The root of the graph is referred to as the basis. The leaf of the graph is called the end effector. The illusion of the movement of the humanoid is reached by rotating the segments around the joints. Consequently if specific joint nodes are missing, no animation can occur. As a result only if the user's humanoid is H-Anim compliant, containing the necessary joint's sequence, the existing animation can be attached to her /his file. Responsible for the latter is the *Animation Adder,* which appends the animation code to the avatar's file, under the correct context. Finally, the user may clear the history of his actions.

4 FRAMEWORK INTEGRATION

There are three major steps in order to integrate this work to an NVE, as were done in EVE's case. The first deals with the Graphic User Interface (GUI) of the Framework. The second deals with the input and output of the framework. In the next two paragraphs we will address this two issues respectively.

The entire user interface of the framework is embedded in a JFrame class instance. This makes it very easy to integrate it to both desktop and WebTop applications that are developed in Java. The process of adding the framework GUI to EVE was very simple: We just added the JPanel object to the specific portion of the JApplet object that forms EVE's GUI.

The framework needs to have access to the X3D/VRML code of the set of animations that an NVE supports. The path to these animations can be appended to a configuration file that the framework reads during initialization. Finally the output of this framework, that is X3D/VRML files that describe an avatar along with the set of animations that were added to it, can be stored to a specific location (described in the configuration file). Once the output is produced, the framework can be configured to call one or more NVE specific functions that adds/add the X3D/VRML code of the avatar to the virtual world. In EVE a function that adds X3D/VRML content to the world was invoked as soon as the framework produce the avatars file.

5 EXAMPLE APPLICATIONS

Example applications for this framework could be distance learning or collaborative work applications. Bailenson and Beall (2006) said that given the advent of collaborative virtual reality technology, as well as the surging popularity of interacting with digital representations via collaborative desktop technology, researchers have begun to systematically explore this phenomenon of Transformed Social Interaction (TSI). TSI (Bailenson et. al 2004) involves novel techniques that permit changing the nature of social interaction by providing interactants with methods to enhance or degrade interpersonal communication. Our proposed framework could be used in order to support the implementation of TSI by attaching in real time humanoid animations in users' avatars.

Furthermore, we can exploit the proposed framework for supporting role changing in collaborative e-learning scenarios that are conducted in a NVE. Our framework can offer the dynamic change of avatars according to the users' roles. Figure 5A, shows an avatar in which we have added various animations hat have been created by the proposed process and is able to walk (Figure 5B, C and D), salute (Figure 5E) and raise a hand (Figure 5F).

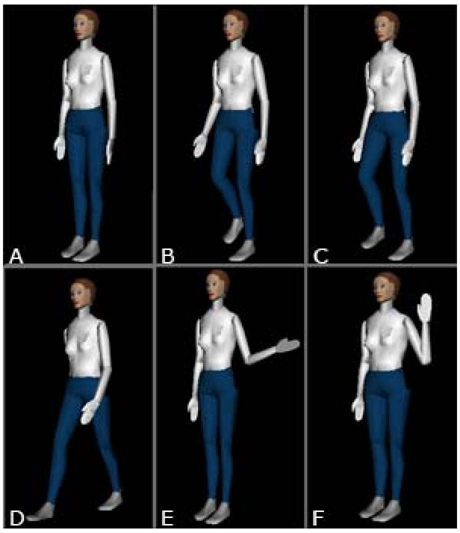

Figure 5: Example avatar.

6 CONCLUSIONS FUTURE WORK

This paper presented a framework that (a) allows a user to import custom H-Anim avatars to an NVE and (b) adds the animations that are supported by the NVE to an H-Anim Avatar. This framework enables NVE's developers to quickly add H-Anim support to their applications. In addition, users can benefit from this framework, in the sense that they are able to import their favourite avatars in every NVE they like.

Our next steps will focus on designing and implementing a VRML/X3D visual editor that will allow the user to modify his/her avatar at execution time and before importing it to an NVE. We also focus on releasing a Public API for this work.

REFERENCES

Bailenson, J., Beall, A. 2006. Transformed Social Interaction: Exploring the Digital Plasticity of Avatars. *Avatars at Work and Play*, Book Series Computer Supported Cooperative Work, Volume 34, ISSN 1431-1496, Springer Netherlands, DOI 10.1007/1-4020-3898-4, pp. 1-16.

Bailenson, J.N., Beall, A.C., Loomis, J., Blascovich, J., & Turk, M. (2004). Transformed social interaction: Decoupling representation from behavior and formin collaborative virtual environments. Presence: Teleoperators and Virtual Environments, 13(4): 428–444.

Bouras, C., Giannaka, E., Panagopoulos, A., Tsiatsos, T., (2006) A Platform for Virtual Collaboration Spaces and Educational Communities: The case of EVE, *Multimedia Systems Journal, Special Issue on Multimedia System Technologies for Educational Tools*, Springer Verlang, Vol. 11, No. 3, pp. 290 – 303.

Buttussi, F., Chittaro, L., and Nadalutti, D. 2006. H-Animator: a visual tool for modeling, reuse and sharing of X3D humanoid animations. In *Proceedings of the Eleventh international Conference on 3D Web Technology (Columbia, Maryland, April 18 - 21, 2006). Web3D '06*. ACM, New York, NY, 109-117. DOI= http://doi.acm.org/10.1145/1122591.1122606.

Humanoid Animation Working Group, 2004. H-Anim.http://h-anim.org.

Miller, T., E. (2000), "Integrating Realistic Human Group Behaviors Into a Networked 3D Virtual Environment", Master's thesis, NAVAL POSTGRADUATE SCHOOL MONTEREY CA.

Ieronutti, L. and Chittaro, L. 2005. A virtual human architecture that integrates kinematic, physical and behavioral aspects to control H-Anim characters. In *Proceedings of the Tenth international Conference on 3D Web Technology (Bangor, United Kingdom, March 29 - April 01, 2005)*. Web3D '05. ACM, New York, NY, 75-83. DOI= http://doi.acm.org/10.1145/1050491.1050502.

Yee, N. 2006. The Psychology of Massively Multi-User Online Role-Playing Games: Motivations, Emotional Investment, Relationships and Problematic Usage. *Avatars at Work and Play*, Book Series Computer Supported Cooperative Work, Volume 34, ISSN 1431-1496, Springer Netherlands, DOI 10.1007/1-4020-3898-4, pp. 187-207.

Capin, T.K., Noser, H. Thalmann, D., Sunday Pandzic, I., Thalmann, N.M. 1997. Virtual human representation and communication in VLNet, Computer Graphics and Applications, IEEE Volume: 17, Issue: 2, ISSN: 0272-1716, DOI: 10.1109/38.574680, pp. 42-53.