2007-04

Composing behaviors and swapping bodies with motion capture data in X3D

Weekley, Jeffrey D.

ACM

Web3D 2007, Perugia, Italy, April 15-18, 2007

Composing Behaviors and Swapping Bodies with Motion Capture Data in X3D

Jeffrey D. Weekley
The MOVES Institute
700 Dyer Road, Bldg 246, Rm.266,
Naval Postgraduate School,
Monterey CA 93943-5001
1.831.656.2307
jdweekle@nps.edu

Curtis L. Blais
The MOVES Institute
700 Dyer Road, Bldg 246, Rm.266,
Naval Postgraduate School,
Monterey CA 93943-5001
1.831.656.3215
clsblais@nps.edu

Don Brutzman
The MOVES Institute
Code USW/Br
Naval Postgraduate School,
Monterey CA 93943-5001
1.831.656.2149
brutzman@nps.edu

ABSTRACT
This paper describes current work in the evolution of open standards for 3D graphics for Humanoid Animation (H-Anim). It builds on previous work to encompass plausible humanoids, humanoid behaviors and methodologies for composition with interchangeable and blended behaviors. We present an overview of the standardization activities for H-Anim, including a proposed extension for the H-Anim Specification which allows for interchangeable actors and dynamic behaviors. We demonstrate a standards-based approach to the complex work flow and data extraction for 3D optical motion tracking systems. We describe how to archive, annotate and transform the whole body and segmented performance data so that they can be used more widely and with less effort. The approach is compressible, streamable, scaleable, repeatable and suitable for large-scale training and analysis, entertainment and games.

Often, X3D and VRML simulations lack the realistic representation of humans. They lack the direct flexibility of control required to build small, but meaningful, task-oriented training scenarios like deploying force protection assets in a busy commercial port. While high-value assets, defensive and offensive agents can be easily and realistically modeled using discreet event simulations, that realism is diminished by the lack of humanoid representations. The visualization is not as engaging and the training not as immersive. Including a rich set of characters with composable and swappable behaviors demonstrating intent of the agent entity heightens both the sense of realism and immersion. Deriving these behaviors is difficult and translating the data into custom applications is craftwork. We propose a standard, archival data format so that captured behaviors can be repurposed and reused, following the SAVAGE approach. This data format will include behavior information and skeletal information such that they can be retargeted and repurposed with a minimum of effort.

Web3D 2007, Perugia, Italy, April 15–18, 2007.
© 2007 ACM 978-1-59593-652-3/07/0004 $5.00

Categories and Subject Descriptors
E.2. [**Data Storage Representations**]: Composite Structures and Object representation – *data types, polymorphism, control structures.*

General Terms
Algorithms, Design, Human Factors, Standardization.

Keywords
Humanoid Animation (H-Anim), motion capture data transformation, blended behaviors and composable bodies prototypes, general pipeline description for motion capture to H-Anim data conversion, XML data archive for motion capture-derived behaviors.

1. INTRODUCTION
There has been an explosion in the use of human avatars in the entertainment industry and in simulations for training and education, but these efforts have been hindered by the highly technical nature of composing meaningful humanoid behaviors and by the wide variance of human phenotypes, as well as by the difficulty in generating realistic human models which are reusable across applications. Hand animation often results in models that are quite satisfying, but these techniques are highly labor intensive. Using human motion capture systems yields many millions of bytes of data which must be laboriously converted for use and still often requires traditional animation techniques for blending of behaviors or retargeting for variances in the kinematics of individual avatars. Further complicating these efforts is our innate ability to sense inconsistencies in humanoid animations. Often, designers of training systems opt for limited sets of behaviors, with few characters in highly constrained conditions. This illusion is mostly satisfactory for game developers and for non-real time computer animation, but it falls far short of the rich variety required to simulate realistic humans in complex Virtual Environments for training. Certainly, none of these humanoids would pass the classical "Turing Test" to be indistinguishable from the real thing.

Since the human motion capture work flow (called a “pipeline”) is complex and fraught with technical pitfalls, analytical simulation tools often choose not to include avatars; rather they use abstractions and aggregations of humanoids. Generating humanoid motion simply to enhance the visualization of a simulation is generally not worth the effort, as the barriers to entry into MoCap are high and the pipeline is complex. But increasingly, simulations do include human motion, which has to be generated and fit to the application in a painstaking process. A discussion of the use of motion capture data in a more persistent way follows.

2. EXTENSIBLE 3D GRAPHICS (X3D)

2.1 Standardization

X3D was approved in 2004 as a successor to ISO/IEC 14772:1997—Virtual Reality Modeling Language (VRML) with new features, advanced application programmer interfaces, additional data encoding formats, stricter conformance, and a componentized architecture that allows for a modular approach to supporting the specification.

2.1 Humanoid Animation Specification

There has been a steady emergence of character modeling software to create and animate 3D human figures, such as Poser from Curious Labs. During the same period a number of systems have also been developed for tracking the motions of a "real world" human being, also known as Motion Capture. The prevalent obstacle encountered when using multiple of these software packages and systems is in the area of information exchange. The lack of a standardized skeletal system that includes segments with lengths and joints with constraints within this community often forces animation houses and motion capture studios to develop their own proprietary solutions to help smooth the transitions between the systems and software they want to use. While systems are improving, demonstrating that

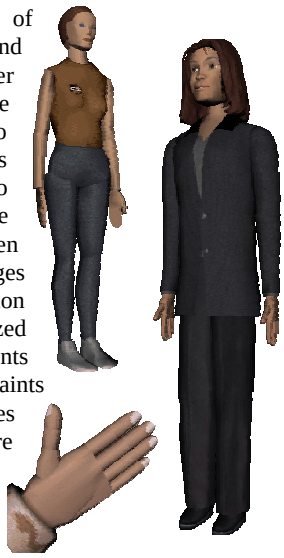


Figure 1 H-Anim 1.0 from Cindy Ballreich and VCom3D

The International Standard known as H-Anim is an abstract representation for modeling three dimensional (3D) human figures is an initial step in unifying these approaches, but it is incomplete without data to populate it. H-Anim describes a standard way of representing humanoids that, when followed, will allow human figures created with modeling tools from one vendor to be animated using motion capture data and animation tools from another vendor. Even with this abstract description and with the data structure mostly realized, there are still significant obstacles to efficient use of humanoids in traditional simulation: data transfer, retargeting and interpolation for missing, incomplete, or combined data sets.

Humanoid Animation Specification 1.0 (H-Anim 1.0) was the first iteration of the standard, but it had its short-comings. For instance, it did not allow for smooth skinning of skeletal representations. The geometry that comprised the body of the humanoid was composed of discreet geometric meshes. These meshes intersected and created seams. They did not deform in realistic ways. They look, at times, cartoon-like and unsophisticated, even though they are quite the opposite. But, as with many visualizations, the sophistication of the application is judged primarily by its visualization component. H-Anim 2.0 aims to address many of the real and perceived short comings of H-Anim 1.0. Most notably, it supports smooth skinning of humanoid skeletons. Since we can now create seamless meshes for skinning characters in H-Anim 2.0, and we can generate complex and realistic motions through motion capture, we should be able to create lifelike characters and animations.

2.2 MPEG-4 AFX

The Motion Picture Experts Group working group has proposed the MPEG-4 standard, which includes the Animation Framework eXtension (AFX) and Bone-based Animation (BBA) for streaming of 3D human motion through MPEG-4. [3]

A review of the available documentation reveals that there are similarities in goals, but this work focuses on practical implementation of relative standards in X3D and implementation of repeatable results.

Because the MPEG-4 AFX work is highly encumbered by patents, even citing it could taint any open source effort. Therefore, it is not considered here.

2.3 X3D Prototypes for Composing Behaviors and Swapping Bodies

X3D Prototypes provide a way for X3D authors to create new nodes that can be used and reused in the X3D scene graph like any other node. Basically, it’s a custom node that carries all the functionality to each instance. Each prototype instance is traversed and rendered when the X3D browser repeatedly loops over the scene graph at run time. It allows for repeatability of branches of the scene graph and avoids the hardwired functionality of repeated nodes that the DEF and USE construction (repetition by reference) require. Each instance carries its own functionality, even though they have the exact same scene graph chunk.

Prototype declarations can also be kept in separate X3D files or library archives. The author then retrieves the prototype of interest by using the ExternProtoDecrlare statement. Thus, a single master copy of the prototype can be held centrally and maintained in a controlled fashion, much like web pages written in hypertext markup language. Over time, this deliberate management technique makes it feasible to use Prototypes for large and complex projects such as a large library of humanoid behaviors and bodies.

Because Prototypes allow authors to define fields and even embed Script code, they are quite helpful when customized X3D objects are needed. Figure 2 illustrates a chart view of the proposed architecture of this paper, with the required Prototypes.

The central function of this interlocking design of prototypes and data is the Human Body Behavior Chooser (HBBC). This Prototype node functions simply to choose which behavior should

be assigned to which body. This is programmatic way of doing

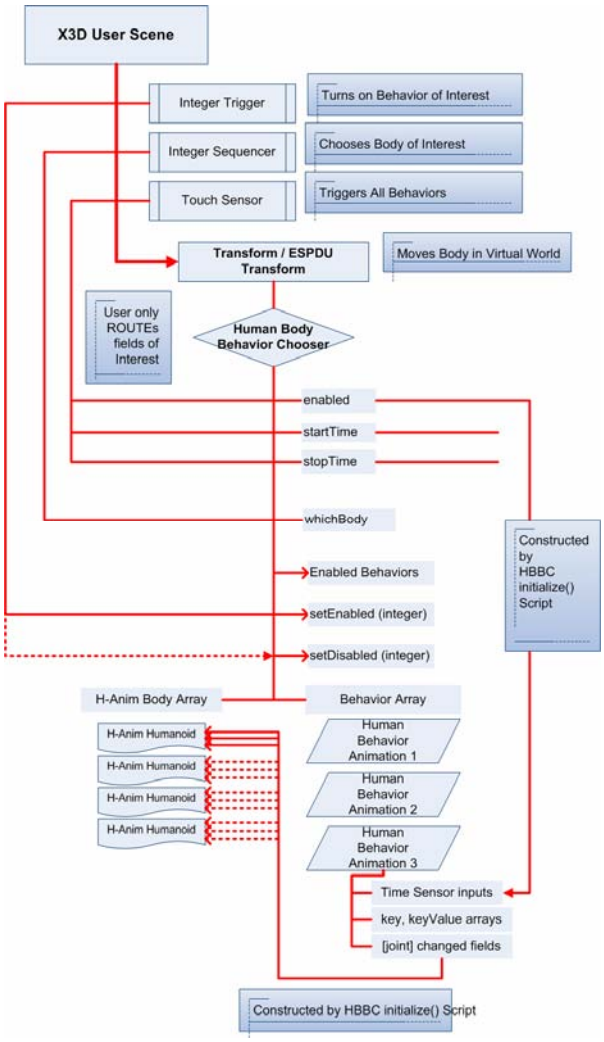


Figure 2 Abstract Data Structure for Composing Behaviors and Swapping Bodies

what a scene author would otherwise do manually (and not at runtime) by choosing an ExternProtoDeclare over another.

The HBBC resides beneath an Entity State Protocol Data Unit Transform (ESPDU Transform). The ESPDU Transform is a way for an X3D scene to interface with the network using the

Distributed Interactive Simulation (DIS) defines an infrastructure for linking simulation of various types at multiple locations to create realistic complex, virtual "worlds" for the simulation of highly interactive activities. It defines the binary layout of a series of messages used to transmit simulation information with support for multicast, unicast or broadcast transport mechanisms for network communications. [3]

The prototype chooser receives entity states from an external packet generator, such as a multiplayer game or discrete event simulation and passes it into the scene via the ESPDU Transform node. This entity state information can be customized to articulate the prototype parameters, choose the behaviors and assign them to a body. It also receives information such as location in the 3D space, velocity, acceleration, and a host of other data required to simulate reality. Though this has not been implemented yet in this research, it is theoretically simple.

The next significant prototype in this schematic is the Human Behavior Prototypes. This is a collection of behaviors at various levels of decomposition. For instance, one behavior might be a walking behavior. This behavior would involved the H-Anim joints and segments from the root, through the sacrum, hip and down each leg. It would not affect any joints or segments in the direction of the head or arms. But another animation, such as a hand wave, which involves other segments and joints would be free to be blended with the walk sequence, making a new behavior not in the library.

3. MOTION CAPTURE

3.1 Laboratory Infrastructure

The use of Motion Capture for generating computer animations is not new relative to the ever-changing landscape of computer science and graphics. While it is still highly specialized and still requires a large investment ($50K - $250K for a small studio), the technology is becoming ubiquitous. The idea of using actual human motion to generate animations is even older than motion capture. Disney filmed the performances of live actors to inform the animation of Snow White (1939), using a process called rotoscoping. Motion capture systems used in the late 1970's and early 1980's required that subjects be fitted with exoskeletons and sensors which fed analog signals to digital converters to generate useable data. Soon, optical tracking systems began to emerge. It is these optical tracking systems, which use markers placed strategically on a subject, which are most widely used today. Optical tracking systems are limited by the speed and resolution of the cameras, and by occlusion of the markers. The occlusion problem can be easily (if not cheaply) solved by adding more cameras. The speed and the resolution of the cameras have steadily improved, analogous to the improvements made in other digital cameras, so that these other limits are effectively removed for single or two person captures. The MOVES Institute currently has a 10-camera MX40 System from Vicon Peak.

This system is capable of generating 2-Dimensional (2D) data rates of approximately 2.683 x 109 bits/second at full resolution and speed for all ten cameras. Practically, however, there is no way to write this amount of data directly to the hard drive of the average high-end PC. So, the cameras pre-process much of this data, screening each full frame (4000 pixels) to return just the pixels which have marker data. This results in practical data rates of between 1.152 X 106 and 1.006 X 107 bits per second. Still, this is a large amount of 2D marker data. It isn't until post processing that a third dimension is calculated from the relative positions of the cameras and markers. Once the third dimension is added to the data set, it results in a predictable increase of 33% in size. Research in meaningful compression of Motion Capture data is just now emerging and is not addressed in the scope of this paper, but its applicability to libraries of motion capture-derived behaviors is obvious.

3.2 Data Representation in Motion Capture Systems

Data export from applications such as Vicon IQ is dependent on pre-processing the data to a meaningful form and a manageable

size by using tools included in the software. Nearly always, the data must be managed to remove gaps in the data caused by occlusion of the optical tracking markers, noise, false returns and a myriad of other faults. Adding additional tracking markers, tuning the cameras, training the performers, defining kinematics preconditions (rigid bodies, properly sized skeletons, constrained joints) all contribute to the quality of the data.

In the Vicon system, the information about the structure of the skeleton is contained in an XML file. There is a high correspondence between this data representation and the H-Anim representation. We applied a style sheet transformation (XSTL) to the Vicon representation, mapping the output to the appropriate H-Anim Level of Articulation (LOA). In the sample data with which we worked, this corresponded to H-Anim LOA 1, but incompletely. A more complete correlation could be achieved using non-sample data and constructing the Vicon skeleton from the H-Anim template. In either case, during the pipeline of data processing in the Vicon IQ software, the skeleton is fit to the marker data following their proprietary algorithms.

This is called "Kinematic Fit" and is dependant on matching markers with marker nomenclature within the Vicon IQ software. Through an iterative process, joints are localized in the 3D space based on marker movement; for example, an elbow joint is located at the rotational axis between. Standard marker placement and naming assures that this process proceeds with a minimum of human intervention, but highly customized skeleton/marker combinations are also possible, since the underlying mathematics is essentially the same. [5]

Other systems each deal with this complexity in different, but similar ways. We believe that there is enough commonality so that H-Anim could be used as an interchange format and for archiving. Keyframe animation is often employed in non-motion capture data sets, and conceptually it is similar to the piece-wise linear interpolation of orientations we propose to construct from motion capture data, even though the sample rates for such data are much higher than typically found in key framing. Another critical difference is that skeletal kinematics models are not often retained when exporting from tools such as Poser® to VRML. The tool simply exports a single VRML scene for each key frame of animation. The scene consists of a textured polygonal mesh with no animation or interpolation.

3.2 XSLT Data Transformation

Mapping one structured data to another via XSLT is a basic function in XML. A transformation expressed in XSLT is called a stylesheet. This is because, in the case when XSLT is transforming into the XSL formatting vocabulary, the transformation functions as a stylesheet. [6] A transformation expressed in XSLT describes rules for transforming a source tree into a result tree. The structure of the result tree can be completely different from the structure of the source tree. In constructing the result tree, elements from the source tree can be filtered and reordered, and arbitrary structure can be added. In our case, we will also apply an algorithm to parse the comma-delimited rotational data, calculate rotations in radians (vice axis-angle rotations) and populate arrays which are matched to joints in the resultant tree, based on their associations to the source. A second pass through the original file is necessary to perform this transformation, as XSLT is not intended for higher mathematical exercises.

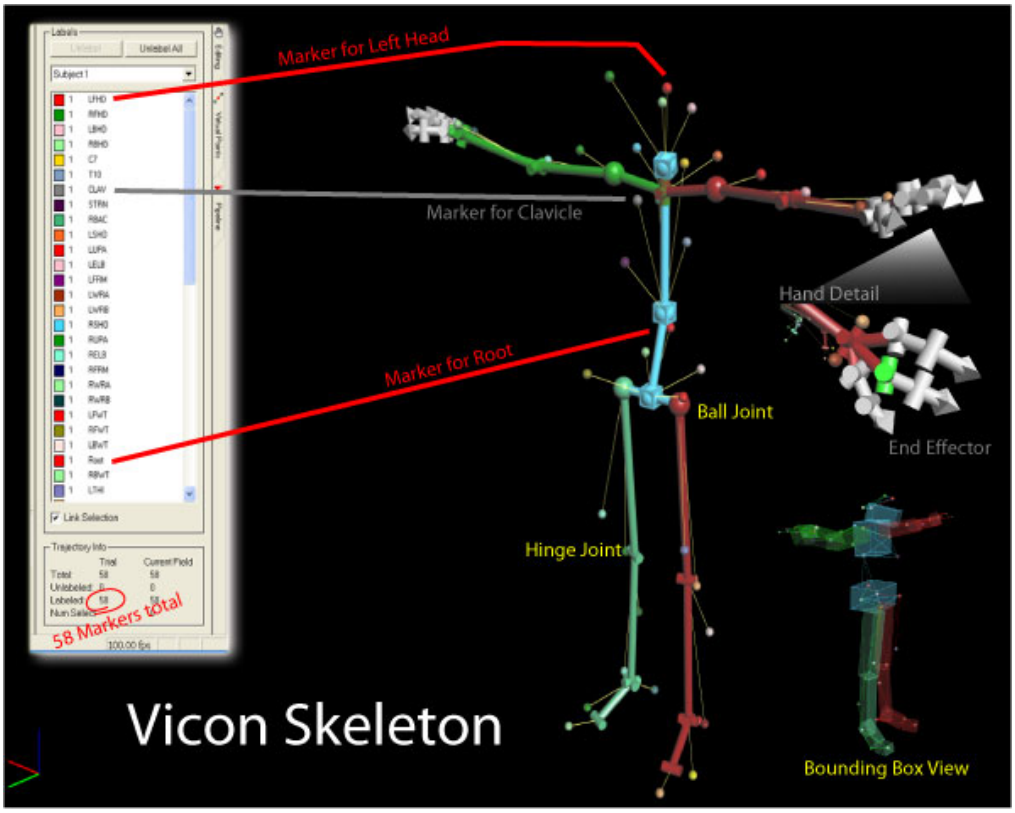


Figure 3 Vicon Skeleton

```
<xsl:template match="KinematicModel/Skeleton">
                <!-- read/output joint and segment data into an HAnimHumanoid element -->
        <xsl:element name="HAnimHumanoid">
                <xsl:attribute name="DEF">HUMANOID</xsl:attribute>
                <xsl:attribute name="name">Humanoid</xsl:attribute>
                <xsl:element name="HAnimJoint">
                        <xsl:attribute name="DEF">hanim_HumanoidRoot</xsl:attribute>
                        <xsl:attribute name="center"></xsl:attribute>
                        <xsl:attribute name="containerField">skeleton</xsl:attribute>
                        <xsl:attribute name="name">HumanoidRoot</xsl:attribute>
                        <xsl:attribute name="skinCoordIndex"></xsl:attribute>
                        <xsl:attribute name="skinCoordWeight"></xsl:attribute>
                        <xsl:element name="HAnimSegment">
                                <xsl:attribute name="DEF">hanim_sacrum</xsl:attribute>
                                <xsl:attribute name="name">sacrum</xsl:attribute>
                        </xsl:element>
                        <xsl:apply-templates select="Segment"/>
                </xsl:element>
        </xsl:element>
</xsl:template>
```

Figure 4 Vicon to H-Anim XSLT

3.2.1 Validating the Transformation

With X3D, validating an XSLT-generated document is as simple as opening it the latest version of X3D-Edit. X3D-Edit is based on Xeena which is a generic Java application from the IBM Haifa Research Laboratory for editing valid XML documents derived from any valid DTD. The DTD-level checks have been carefully authored to assure strict adherence to X3D specifications for data type and structure. This way, the general XML editor becomes a graphics file editor that enables simple error-free editing, authoring and validation of X3D or VRML scene-graph files. [7]

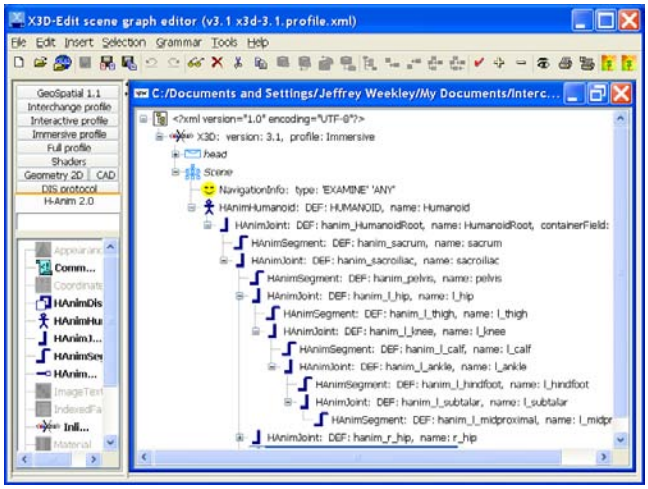


Figure 5 X3D-Edit View of H-Anim Skeleton Derived from Vicon Skeleton File

Figure 5 illustrates the valid X3D file generated by applying the XSLT to map from Vicon to H-Anim skeletons. This is significant because the output file from the motion capture performance is free of any reference to the length of the segments or constraints to the joints required to reconstruct the performance outside of the Vicon IQ software (which associates the skeleton with the fitted motion capture data within a custom Eclipse data base). One must have both the segment lengths and rotational values for the joints in order to propagate rotations through the kinematics chain to the end effectors (sites).

This ensures that the skeleton matches the performer after it is fit and smoothed and that the exported skeleton can be easily populated with rotational data. Historically, this has been the most difficult part of H-Anim behavior authoring – generating all but the grossest gestures by hand was painstaking. Admittedly, generating MoCap-derived behaviors is just as difficult, but providing large libraries of valid H-Anim behaviors as prototypes and building these archives as we build other model repositories will make authoring easier, as it has made generating meaningful visualizations in other domains possible. [8] Furthermore, if the behaviors are described as proposed in this paper, they can be composed by humans and by simulation.

3.3 Motion Synthesis

The data which is exported from the Vicon system is somewhat vague. Precisely translating this from a comma separated value file to rotations in radians is work to be done in the near future, pending confirmation of our presuppositions about the data itself.

We looked at a Range of Motion (ROM) study often used in calibrating skeleton models, as described above. But the data also contained clearly visible gaps. There were entire segment joint sequences with no data at all. You could also segregate the data based on the hierarchy of the skeleton itself. Removing the null values (and their associated segments and joints) from the data representation in the body representation or culling the behaviors at logical points, e.g. from the hips downward for walk animations, or from the shoulder outward for wave animations, would allow for behaviors to be blended or composed. The Behavior Chooser in the HBBC would not be constrained to choose only whole body animations, but could construct a new behavior from compatible less-than-whole-body behaviors.

By including meaningful meta data about the extent of the behaviors in the behavior array, and by including data for

retargeting these behaviors for polymorphic bodies types, we hope to be able to repurpose and compose these behaviors. It may be computationally intense, but following the prototype architecture and maintaining control on the server side, would allow for complex transformations for retargeting to be provided by a web service.

Behavioral Animation is a robust area of research, but it is not within the scope of this work. This work might simply allow programs within this research area to visualize their results more easily by providing bodies and behaviors as a web service.

4. SWAPPING BODIES

An important consideration in the overall design of this functionality was whether or not to include static or dynamic routing. The highly structured nature of humanoid skeletal representations would argue for static routing. A non-behavior would send null values to joint rotations. But overloading the structure would assure that when a behavior is triggered, the route to carry that behavior from interpolator to joint would be present, always. But since the node names are static and universal in H-Anim, static routing isn't necessary. See Figure 6.

Routes can be created at initialization, perpetuated as long as there are behaviors which require them and either left intact, or collected as garbage by the 3D browser or viewer. If a behavior requires a new route, it is constructed with a priori knowledge of the affected nodes, and disallowed if routes already exist and have populated arrays of rotational values.

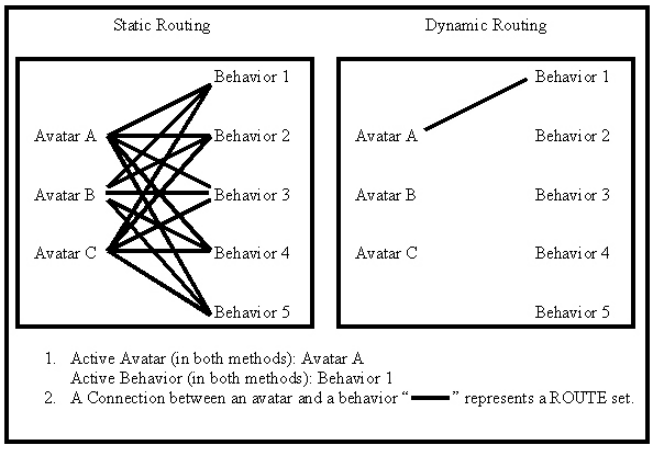
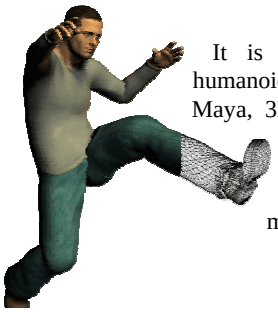


Figure 6 Static Vs. Dynamic Routing

4.1 Complex Human Meshes



It is now possible to create complex humanoid meshes in applications such as Maya, 3DS Max or Poser. Both Maya and 3DS Max allow for parametric description of humanoid meshes and skinning of skeleton rigs (which may or may not conform to H-Anim standards for naming and hierarchy). Poser is best suited for mesh generation with highly realistic textures and clothing. While these meshes are compelling and interesting in static poses, often because the animations generated for them are hand-authored, the suspension of disbelief is destroyed as soon as they begin moving. It's our intention to marry these complex meshes with H-Anim behaviors using the HBBC prototype. This approach is scaleable, repeatable and follows the SAVAGE approach. [9]

Figure 7 Complex Humanoid Mesh with Textures from Poser®

5. FUTURE WORK

Admittedly, this work is in its preliminary stages. We hope to demonstrate more fully repeatable results when the paper is presented. We have yet to complete the scene parser which will build the prototypes, parse the MoCap data and generate a valid X3D H-Anim humanoid, but the tough problems of prototyping and translation of skeletal information are done. Ideally, we will discover that the H-Anim extension is fully capable of describing data generated from most common tools and that it can be used as a general repository for these data types. Furthermore, reducing the complexity of the required data pipeline from tools such as ViconPeak's IQ system [10] We are confident that this work will progress.

6. ACKNOWLEDGMENTS

Thanks to the people a ViconPeak for their generous donation of some of the motion capture equipment through the NPS Foundation. Also, thanks to the NPS students, past and present, who have contributed so much to the success of all the X3D-related projects.

7. REFERENCES

[1] https://savage.nps.edu/Savage/

[2] "An Introduction to the MPEG-4 Animation Framework eXtension," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 7, July 2004

[3] www.web3d.org/x3d/workgroups/dis/

[4] www.cs.utah.edu/~halzahaw/MotionCapture_main.html

[5] www.vicon.com/applications/animal.html

[6] www.w3.org/TR/xslt#section-Introduction

[7] www.web3d.org/x3d/content/README.X3D-Edit.html

[8] Apaydın, Ozan Networked Humanoid Animation Driven by Human Voice Using Extensible 3D (X3D), H-Anim and Java Speech Open Standard, Master's thesis, Naval Postgraduate School, Monterey, CA 2002

[9] "Emerging Web-based 3D Graphics for Education and Experimentation," *Proceedings of Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)* 2001.

[10] www.vicon.com/products/viconiq.html

Figure 1 H-Anim 1.0
from Cindy Ballreich and VCom3D

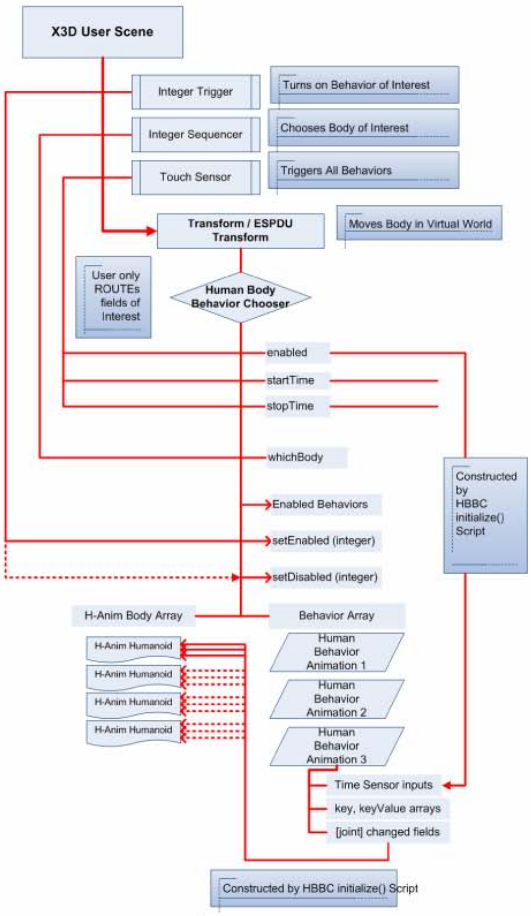


Figure 2 Abstract Data Structure
for Composing Behaviors and Swapping Bodies

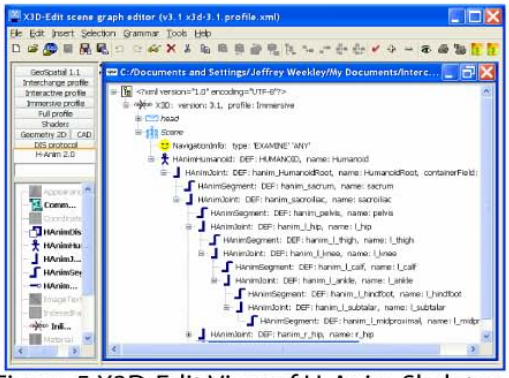


Figure 5 X3D-Edit View of H-Anim Skeleton
Derived from Vicon Skeleton File

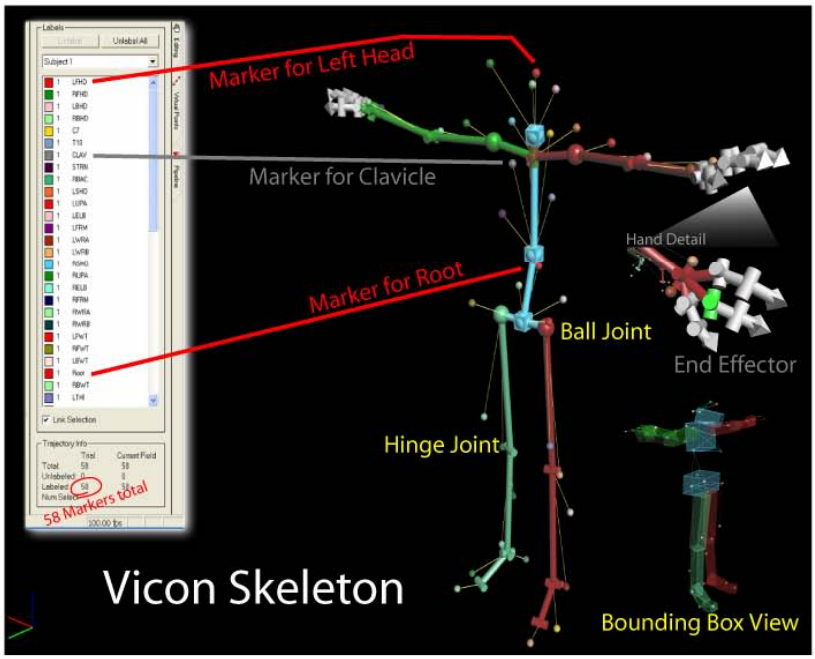


Figure 3 Vicon Skeleton

Figure 7 Complex Humanoid Mesh
with Textures from Poser®