2001-01-2103

# Standardizing Human Model Queries[1]

**John D. Ianni**
Air Force Research Laboratory
Crew System Interface Division
Wright-Patterson AFB, Ohio 45433



## ABSTRACT

One of the goals of the Society of Automotive Engineers Human Modeling Subcommittee has been to link models of human processes with anthropometric (human body) models. This requires conveying actions such as climb, lift or operate, and it requires a method to convey the state of the virtual world. This paper describes a possible standard protocol to query figure models for information such as the human's posture, clothing worn or an object's location. This protocol was derived from user requirements and specifications of several commercial figure models. This paper is intended to provide a starting point for further discussions about such a standard.

## INTRODUCTION

The Society of Automotive Engineers (SAE) Human Modeling Subcommittee (G-13) exists because users, developers and researchers see the benefits of standardization. Since human modeling is still a relatively new technology, there are opportunities to converge on best practices. The G-13 Software and Virtual Reality Subcommittee is looking for ways to promote open systems architectures. The goal is to allow software segments from various vendors to be combined to create the best human model for the application. This paper is intended to describe one segment of this standardization effort. The specification contained herein is not an official SAE protocol.

One of the goals of SAE G-13 has been to link models of mind and body (Ianni, 1999). More specifically stated, models of the mind can include detailed models of cognition or, at a more abstract level, task performance. This paper will collectively refer to these models as process models. Body models are sometimes referred to as human figure models, avatars or manikins. In this paper they will be referred to as figure models. If a linkage is created between process and figure models, physical and non-physical aspects of tasks may be evaluated concurrently rather than running them independently. This can be a powerful merger as described later.

The type of information to be exchanged by these models is depicted in Figure 1. The process models send task commands to the human figure model as well as a unique identification number for that instantiation. This identification is needed to allow the process model to request and receive task status information including task times or a failure code if the figure model is unable to complete the task. The figure model can return information such as a task's status, task times, measurements of a human, distances between objects, or the state of a door (open or closed).

Interconnecting these types of models can reap benefits in addition to concurrent analyses. Process models can provide a valuable task-network interface to figure

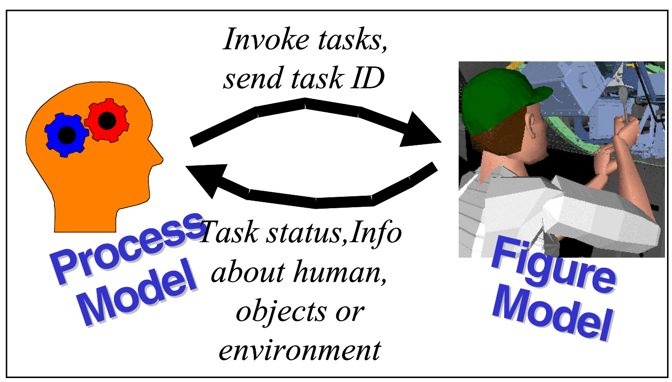


*Figure 1: Communication between models of human figures and processes*

1 The views expressed in this paper are those of the author, and do not necessarily represent the views of the U.S. Department of Defense or its components. Cleared for Public Release 28 December 2000, ASC:00-2500.

models that often have crude methods to simulate tasks. Conversely, process models often lack a realistic representation of the body, which can detract from the analysis process.

A less obvious benefit involves the implementation of natural language interfaces. A neutral protocol is critical in developing natural language interfaces to kinematic models (Badler, 1999). The goal is to direct a figure model using phrases such as "Walk over to the car" and "Remove all of the spark plugs." Such capabilities will greatly simplify the creation of task simulations with figure models, thus reducing the time and training required.

Likewise imagine being able to ask a modeling system "How tall is Human1?" or "What is the temperature of the spark plug?" or "How far is Human1's hand from the fastener?" This capability is more than a convenience for task simulation; without the ability to pass certain information a task cannot be performed. For example before adding gasoline to a car, you must first ensure the cap has been removed from the tank. For more complex tasks (e.g., engine removal) with possibly redundant steps, it is important to check if the step has already been completed. Several subtasks may call for eye protection but the figure model should only put on the glasses once.

As we become increasingly able to import product and engineering data in addition to geometry from computer-aided design (CAD) systems, this type of capability will be invaluable for simulation of human activities. For example, we may be able to query about the degrees of freedom of a steering wheel or an access panel. We should also be able to determine what objects are in the environment, their dimensions, and what they weigh. Likewise these values should be able to be set or changed by the process model using similar commands.

Other specifications have been developed to control figure models including those from Moving Picture Experts Group (MPEG) and Virtual Reality Modeling Language (VRML; H-Anim, 1999). However these specifications are primarily intended to define the body size and shape, and control via joint motions or key frame animations. This type of control is good for non-engineering types of applications but is not practical to interface process and figure models.

This was a conclusion in the Air Force Depth program (Ianni and Lane, 1998). On this program, the Army's Human Operator Simulator (HOS) was interfaced to Jack. HOS was an early incarnation of Integrated Performance Modeling Environment (IPME) depicted in Figure . However it was found that using HOS to control every minute motion was impractical. The manikin needed to "understand" certain concepts such as lifting, pushing, carrying, and climbing. The figure model also needed to be able to convey the state of the virtual world. It was impractical and inefficient to have the process model maintain the state of every object in the scene.

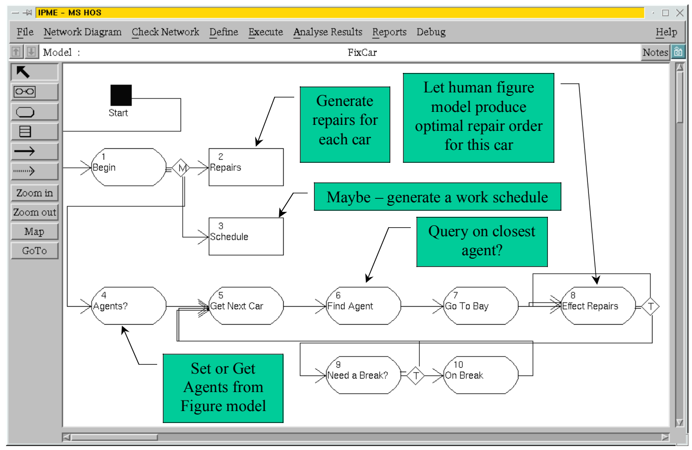


*Figure 2: IPME task network with annotations to clarify certain steps*

## SPECIFICATION

The specification described in this paper was developed based on experiences developing two Air Force human modeling systems (namely Crew Chief and Depth), my involvement with the human modeling community through SAE G-13, reviews of application programmer interfaces (APIs) for three human modeling systems, and reviews of other literature. Although some syntactical examples are provided, this specification is not intended to provide the final syntax for such a standard but rather provide an initial set of queries to be incorporated into such a standard. Whenever possible international standard units were used. In most cases the output values would be floating point numbers.

Although this specification is intended primarily to query human models, it can also be used to set values. For example, the stature of a specific human can be changed or a program external to the figure modeling system can specify the location of a certain object.

The tables in the appendix describe the type of information that should be supported in a standard query language. These are categorized into queries about the human (e.g., stature), about objects (e.g., weight) and about the environment (e.g., temperature). The following query would return the gender (sex) of the specified human model:

```
Human1.GetProperty(“gender”)
```

Table 1 includes queries about anthropometry such as stature, age, gender, mass and details about body segments. Most of the queries are accomplished using a GetProperty or SetProperty method:

Clothing queries are included within anthropometry table and are further elaborated in Table 2. The syntax for these is shown in the following examples:

Human1.GetProperty("clothing","foot")

This example returns values meaningful to the user such as "combat boot," "high heels," or "tennis shoes." A more rigorous definition of clothing may be necessary in the future but is beyond the scope of this paper. The status of the human, including location, posture and task performance, is included in Table 3.

Since humans interact with their surroundings, it is important to be able to get information about objects and the environment. Table 4 describes queries that can be made about objects. Environmental factors, described in Table 5, were derived from Bridgman, et al, 1996. These include climatic conditions as well as noise and radiation.

## APPLICATIONS AND EXAMPLES

There are different applications for this specification. As an application programmer's interface (API) it can be used to link figure models with process models that lack a realistic body representation or provide an API for web-based applications. It may also provide a foundation for a common user interface for figure models. This may facilitate embedding figure models within analysis and engineering software packages.

To illustrate how this specification may be used, let's consider a simulation of a spark plug removal. Since the process model has no information about the environment, it must first position the human at the automobile to perform engine maintenance using the protocol described in Ianni, 1999:

Human1.MoveTo(car1.engine,maintenance)

Once this is has been completed the process simulation needs to know whether the hood is up (open) or down (closed). So using the query specification a message is sent from the process simulation to the figure modeling system such as:

car1.hood.GetState("Open")

To which "True" would be returned since the hood on car1 is already open. Next it may be necessary to determine if the engine is running:

car1.engine.GetState("Running")

To which "False" is returned because the engine is not running. However the spark plug may be too hot to work on with bare hands if the engine was just shut off. Thus to determine the temperature of the second spark plug the following query is sent:

car1.plug2.GetTemperature()

To which $300^\circ$ Kelvin is returned which is safe for barehanded work.

Likewise using a similar protocol, state information can be set by the process model. The following example sets the temperature of the spark plug to $350^\circ$ Kelvin:

car1.plug2.SetTemperature(350)

Other parts of this specification can be used to set or change, rather than query, values. Having an interface to change conditions in the virtual world could be quite powerful. For example, altering a virtual human, such as stature, gender or weight, can allow tasks to be analyzed with different body types.

## CONCLUSION

The proposed standard for querying figure models, along with the action invocation standard (Ianni, 1999), has considerable potential. With a standard interface to figure models, it will be possible to analyze task performance in conjunction with traditionally CAD-based analyses. Through the implementation of Distributed Interactive Simulation and High Level Architecture (HLA), we have discovered great potential in linking simulations. The somewhat narrow field of human factors simulations can probably benefit from interconnectivity. However it is important to note that this specification is intended for detailed engineering analyses and not less detailed simulations normally included in an HLA Federation Object Model.

## ACKNOWLEDGEMENTS

The author would like to thank Mr. Ceri Pritchard of BAE SYSTEMS, Inc.; Ms. Shelley Scott-Nash of Micro Analysis and Design, Inc.; Mr. Chris Ege of UGS, Inc. (Engineering Animation, 2000; Carnegie Mellow University, 1999); Mr. Casey J. Pruett of TecMath, Inc. (TecMath, 2000); and Mr. Sylvain Marie of Genicon, Inc. for their very useful input.

## REFERENCES

1. Badler, Bindiganavale, Allbeck, Schuler, Zhao, Palmer. "Parameterized Action Representation for virtual human

agents," In J. Cassell (ed.), Embodied Conversational Characters, MIT Press, 2000.
2. Badler. "Animation 2000++," IEEE Computer Graphics and Applications 20(1), Jan-Feb 2000, pp. 28-29.
3. Bridgman, Hale, Ianni, Ziolek. Maintenance Hazard Simulation: A Study of Contributing Factors, AL/HR-TP-1996-0044 | DTIC/AD-A329627, 1996.
4. Carnegie Mellon University. Alice Web Site, http://www.alice.org. 1999.
5. Defense Modeling and Simulation Office, High Level Architecture Web Page, http://hla.dmso.mil/index.php?page=64.
6. Engineering Animation Inc., Jack Applications Programmer's Interface Specification (proprietary), 2000.
7. H-Anim Working Group, Specification for a Standard Humanoid Version 1.1, Aug 3, 1999.
8. Ianni and Lane. DEPTH Final Report. AFRL/HE-WP-TR-1998-0007, DTIC/AD-A343646, 1998.
9. Ianni. "A Specification for Human Action Representation." Society of Automotive Engineers (SAE) Digital Human Modeling International Exposition Proceedings, 1999.
10. TecMath, RAMSIS Applications Programmer's Interface Specification (proprietary), 2000.

*Table 1: Information About Human*

| **Description** | **Inputs** (in addition to human identifier) | **Output** (NULL upon error) |
|---|---|---|
| Stature | standing \| sitting | meters |
| Segment length | name of segment | meters |
| Segment circumference | name of segment, distance from middle of segment in mm | meters |
| Landmark measurement | name of measurement | meters |
| Mass | | kilograms |
| Gender | | male \| female |
| Age | | years |
| Joint | Location \| limits on x-y, y-z or x-z plane | |
| Degrees of freedom | Joint designation | 3-D degrees of freedom |
| Clothing | Category from table 2 (i.e., purpose, body coverage, gender, etc.) | If category is not specified, all information on the clothing for that human is returned |
| Strength | Male \| Female | Percentile |
| Nationality | Not yet defined | |
| Race (descent) | Not yet defined | |

*Table 2: Clothing Descriptions*

| |
|---|
| This table lists some aspects of clothing that should be considered. Although it may not be a complete list, it should give an indication of what types of information should be conveyed. There can be multiple entries for most of the following items. For example, clothing can have multiple colors, materials, or purposes. |
| <u>Purpose</u> – casual, business, construction, manufacturing, medical personnel (e.g., surgeon, nurse), medical correction (e.g., eye glasses, braces), sports (e.g., running, tennis, baseball) |
| <u>Body coverage</u> – foot \| ankle \| leg \| pelvis \| torso \| arm \| hand \| finger \| neck \| head \| face |
| <u>Climate</u> – arctic \| cold \| mild \| hot \| tropical; rain \| snow \| dry |
| <u>Material</u> – default is the outer material but other layers can be specified as well |
| <u>Color</u> – red-green-blue (RGB) designation, default is the most significant color |
| <u>Thickness</u> – maximum, minimum, average; for footwear – sole thickness at ball and heel |
| <u>Stiffness</u> – a range from 0 (no resistance, essentially bare) to 100 (full resistance like medical casts used to set broken bones) |

*Table 3: Human's State*

| **Description** | **Input** (in addition to human identifier) | **Output** (NULL upon error) |
|---|---|---|
| Location | | 3-D coordinate of center of gravity |
| Orientation (perpendicular to input value) | head \| shoulders \| hips \| feet \| left foot \| right foot | Normal vector (3 ordinates) |
| Direction of motion | | Normal vector (3 ordinates) |
| Speed | | meters per second |
| Looking at | | 3-D coordinate of where the human is focusing |
| Holding | | Designation (name) of object being held, if any |
| Posture | | Description of estimated posture (e.g., standing, sitting, supine, kneeing, kneeing on right knee, etc.) |
| Joint angle | Name of joint (from model definition group) | Up to three rotations in radians |
| Task State | TaskID – This is a unique identification number defined when each task is initiated. | If TaskID = NULL then return human designation and TaskID for each human, otherwise return True if task was completed (False otherwise) and simulation time elapsed performing the task so far |

| | | |
|---|---|---|
| Fatigue | TaskID | Percentage |
| Role | | Name of job being performed (e.g., operator, navigator, or foreman) |

*Table 4: Information About Objects*

| **Description** | **Input** (in addition to object identifier) | **Output** (NULL upon error) |
|---|---|---|
| Names | Object or criteria | Names of all non-human objects in the environment meeting the input criteria |
| Location | | 3-D coordinate of center of gravity |
| Relative distance | Object #2 (in this case, objects can be humans, sites or 3-D coordinates) | Straight-line distances and x,y,z offsets |
| Orientation (perpendicular to input value) | [Designation of front] | Normal vector (3 ordinates) |
| Vector of motion | | Vector, meters per second |
| Degrees of freedom | Joint designation | 3-D degrees of freedom |
| Speed | | meters per second |
| Mass | | kilograms |
| Color | | Hexadecimal red/green/blue ranging from 000000 (black) to FFFFFF (white) |
| Shape | | box \| cone \| cylinder \| parallelpiped \| prism \| pyramid \| regular pyramid \| right cone \| right cylinder \| right prism \| sphere \| common real world object such as a hand tool, gear shift or steering wheel that are enumerated in Ianni, 1999. |
| Dimensions | | Can be one dimension (length), two dimensions (length, width), or three dimensions (length, width, height) |
| State | | Open \| Closed \| Locked \| Moving \| At rest \| Right-side-up \| Up-side-down \| Normal \| Abnormal \| Mounted \| Dismounted \| Fastened \| Unfastened \| Running \| Not running |
| Temperature | | Kelvin |

*Table 5: Information About The Environment or Scene*

| Description | Input | Output |
|---|---|---|
| Coordinate system | Which axis is up, left vs. right handed | |
| Number | Humans \| Objects \| Any other non-ambiguous entity | Number of a certain type of entity in the environment (integer) |
| Temperature | Required: "Ambient" to specify this is not a query for the temperature of an object; Optional: wind chill | Ambient temperature in dC |
| Noise level | | Ambient noise level in dB |
| Precipitation | | Light \| Medium \| Heavy Rain \| Snow \| Hail |
| Wind | | Vector, meters per second |
| Radiation | Type (Radio \| X \| Gamma \| UV \| Microwave) or frequency | E = electric field strength<br>H = magnetic field strength<br>S = power densities |
| Lighting | Coordinate | L = illumination in lux (lx) |
| Voltage | | Amps, Volts |