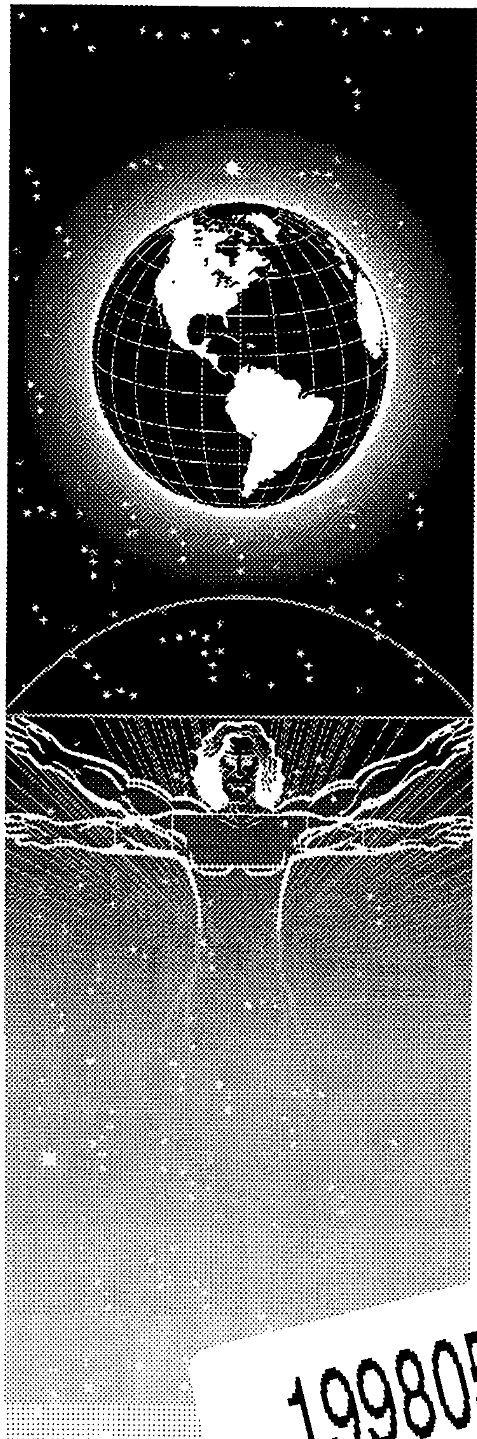
AFRL/HE-WP-TR-1998-0007



UNITED STATES AIR FORCE ARMSTRONG LABORATORY

Design Evaluation for Personnel, Training and Human Factors (DEPTH) Final Report

John D. Ianni

HUMAN EFFECTIVENESS DIRECTORATE
Crew Survivability & Logistics Division
Sustainment Logistics Branch
Wright-Patterson AFB OH 45433 7604

Ken Lane

HUGHES MISSILE SYSTEMS COMPANY
Post Office Box 11337
Tucson, Arizona 85734

January 1998

Final Report for the Period Oct 1991 to Sep 1997

Approved for public release; distribution is unlimited.

AIR FORCE MATERIEL COMMAND
Human Effectiveness Directorate
Crew Survivability & Logistics Division
Wright-Patterson AFB OH 45433 7604

NOTICES

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-1998-0007

The voluntary informed consent of the subjects in this research was obtained as required by Air Force Instruction 40-402.

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE DIRECTOR

For [signature] Lt Col
THOMAS J. MOORE, Chief
Crew Survivability & Logistics Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
13 January 1998

3. REPORT TYPE AND DATES COVERED
Final Technical Report, 1 Oct 91 - 12 Sep 97

4. TITLE AND SUBTITLE
Design Evaluation for Personnel, Training and Human Factors (DEPTH)

5. FUNDING NUMBERS
Contract: F33615-91-C-0001
PE: 63016F
PR: 2940
TA: 03
WU: 05

6. AUTHOR(S)
John D. Ianni - Air Force Research Laboratory (AFRL/HESS)
Ken Lane - Hughes Missile Systems Company

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Hughes Missile Systems Company
Post Office Box 11337
Tucson, AZ 85734

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/HESS
2698 G Street
Wright-Patterson AFB, OH
45433-7604

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-HE-WP-TR-1998-0007

11. SUPPLEMENTARY NOTES
AFRL/HES Technical Monitor: John Ianni, AFRL/HESS

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
Design Evaluation for Personnel, Training and Human Factors (DEPTH) was a five phase program to develop software to facilitate maintainability assessment of weapon systems. This three dimensional (3-D) graphical simulation of human activity was primarily intended to facilitate man-machine design analyses of complex systems. By importing computer aided design (CAD) data, the human figure models and analysis algorithms can help to ensure components can be seen, reached, lifted and removed by most maintainers. These simulations are also useful for logistics data capture, training, and task analysis. DEPTH was also found to be useful in obtaining task descriptions for technical manuals.

14. SUBJECT TERMS
maintenance simulation, human model, virtual mockup, design evaluation, computer-aided design

15. NUMBER OF PAGES
34

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UNL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

Preface

This document summarizes the research and development performed on the DEPTH program (F33615-91-C-0001). DEPTH was funded by Armstrong Laboratory, Logistics Research Division (which became the Air Force Research Laboratory, Sustainment Logistics Branch – AFRL/HESS), Wright-Patterson Air Force Base, Ohio 45433-7604. AFRL and the prime contractor, Hughes Missile Systems Company (HMSC), Tucson, Arizona, gratefully acknowledges the efforts of the following organizations:

Battelle Memorial Institute, Columbus, OH

Center for Computer Aided Design, University of Iowa, Iowa City, IA

Center for Human Modeling and Simulation, University of Pennsylvania, Philadelphia, PA

Computer Sciences Corporation, Dayton, OH

DRaW Computing Associates, Inc., Philadelphia, PA

Hughes Training, Incorporated, Minneapolis, MN

Hughes Information Systems, Fullerton, CA

Integrated Engineering Solutions, Inc., Iowa City, IA

Transom Technologies, Inc., Ann Arbor, MI

University of Dayton Research Institute (CSERIAC), Dayton, OH

Summary

Design Evaluation for Personnel, Training and Human Factors (DEPTH) was a five phase program to develop software to facilitate maintainability assessment of weapon systems. This three dimensional (3-D) graphical simulation of human activity was primarily intended to facilitate man-machine design analyses of complex systems. By importing computer aided design (CAD) data, the human figure models and analysis algorithms can help to ensure components can be seen, reached, lifted and removed by most maintainers. These simulations are also useful for logistics data capture, training, and task analysis. DEPTH was also found to be useful in obtaining task descriptions for technical manuals.

Table of Contents

1. Introduction

The DEPTH program addressed the feasibility of analyzing maintainability in early design by simulating interaction between technicians and equipment [1]. These analyses were traditionally performed only after costly physical mockups were built – too late in the acquisition process to make many changes. These simulations also provided a method to establish manpower requirements, determine maintenance procedures and create training aids.

This report addresses the key objectives as established by Armstrong Laboratory, Acquisition Logistics Branch[1] and the contractor, Hughes Missile Systems Company (HMSC). A description is given for each objective, approach used and lessons learned. Articulated models of the human body – known as human figure models, virtual humans, mannequins, and avatars – will be referred to as "human models" in this report. The person operating the DEPTH software will be referred to as the "user."

Virtual mockups can be created in DEPTH or imported from external CAD software. Since most new weapon systems are designed with 3-D CAD, they are ideal candidates for DEPTH analyses. DEPTH demonstrated on the F-22, F-15 and B-1B aircraft the feasibility to simulate maintenance activity on portions of weapon systems. Simulations created by Lockheed-Martin Tactical Aircraft Systems Company provided a useful analysis of the left-forward power supply removal for the F-22 System Program Office.

DEPTH provides general purpose task simulations called motion models that ease the construction of specific simulations. Creating these motion models was not a simple task. Even basic movements – actions that are taken for granted in the real world – had to be constructed using numerous low-level motion primitives. Walking, grasping, turning, and positioning are not simple tasks for virtual humans, especially since they must perform them in various situations [2,3]. Simulating maintenance is even more complex since it must do these "simple" tasks and more. For example, removing a rack-mounted unit requires determining:

- What fasteners and cables need to be removed
- The order in which to remove them
- How to deal with objects that limit access

If the simulation gets "confused" the results can be comical, but there is usually a way to eliminate the ambiguity. In the worst case, each joint motion can be specified for total control of the simulation, however, some analyses will not be performed if joint motions are used in place of task simulations.

As a DEPTH subcontractor, the University of Pennsylvania (Penn) improved the Jack articulated figure modeling software. Realistic human bodies can be created from established anthropometric data bases or from 19 body measurements. Strength limitations can be assessed using data from validated studies on the Crew Chief program [7]. Tasks can be simulated autonomously as mentioned above or virtual reality devices can be used. Specifically, the human model can mimic movements of body-mounted sensors and tracking gloves for finger motion.

The structure of the DEPTH software is shown in Figure 1. The graphical user interface (GUI) module controls most of the user input and output. It interacts with the other modules that handle the details of figure generation, environment definition, and simulation. The virtual reality module, however, provides a direct interface between the user and Transom Jack. Other modules handle analysis and evaluation functionality. Each module shows a different relationship between the components.

[1] In November 1997, Air Force laboratories were consolidated. Now the Acquisition Logistics Branch is the Air Force Research Laboratory, Crew Survivability and Logistics Division, Sustainment Logistics Branch.

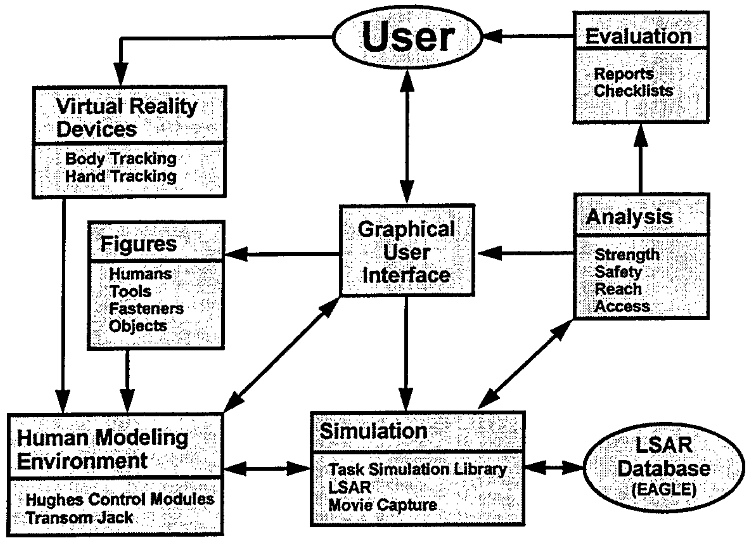


Figure 1: DEPTH Software Structure

1.1 Phases I and II

The first two phases involved baseline research and development (R&D) in human modeling, virtual hand tools, CAD translation, and motion simulation. The hand tools were obtained from the Crew Chief program and an example dialog is shown in Figure 2. These two phases were conducted at the Kearny Mesa, CA facility of General Dynamics, Convair Division – later purchased by the HMSC – and are not addressed in detail in this report. The remaining phases were performed by HMSC in Tucson, AZ.

Although most of the software was eventually rewritten from these phases, several important lessons were learned. For example, it was learned that creating low-level motion simulations using an external process modeling software was impractical. Specifically, message passing was insufficient to handle the timely interaction needed between the physical human and process models.

1.2 Phase III

The Phase III objective was to develop the human factors analysis capability. This was accomplished by improving the hand functionality, on-line guidance and GUI. The hand functionality included more hand fidelity, grasping algorithms and an interface to a hand tracking device. On-line guidance included a maintenance human factors checklist from MIL-STD-1472D [9]. The improved GUI was needed to decrease the learning curve for simulation users.

1.3 Phase IV

The Phase IV objective was to evaluate the effects of environmental conditions on task performance, provide flexible control over anthropometry, and support multiple human models. The environmental conditions included effects of ambient and object hazards including temperature, noise, and radiation. GUI refinements continued, and three new tasks were added including the development of the motion model library, a demonstration of DEPTH on an actual design problem, and a feasibility study regarding technical manual generation.

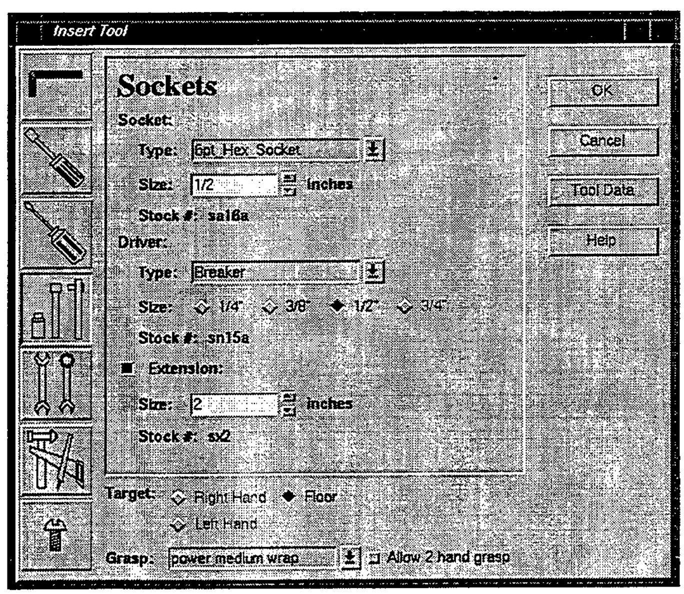


Figure 2: Hand Tool Dialog Box

1.4 Phase V

In Phase V, the final phase, output of logistics data and training media were added. An interface was created to send simulation results to a logistics database. The capability to create PC-capable movies (i.e., QuickTime) from the simulation was added. In addition, improvements were made to the Transom™ Jack® human modeling system (herein referred to as Jack).

1.5 Code Statistics

DEPTH is a combination of the Tcl/Tk [10], Lisp, C, and C++ languages. The number of lines of code required for implementation is summarized in Figure 3. The data is presented for each release of DEPTH since Release 2.0. Note that a majority of the code is written in C/C++. Tcl/Tk was used to create the GUIs and is described in more detail later in this report.

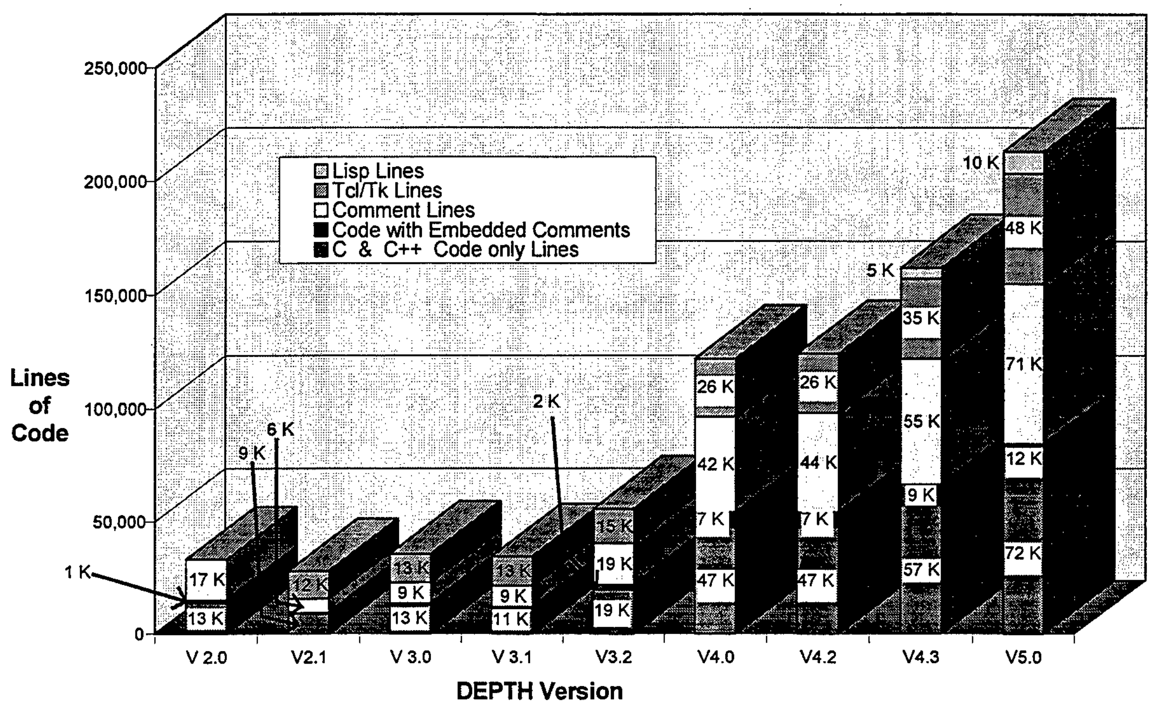


Figure 3: DEPTH Code Statistics

2. Major Accomplishments

2.1 Task Simulation

Objective

A maintenance task simulation should allow more accurate and less costly assessments to be made on proposed weapon system designs. If simulations are too difficult or time consuming to create, few people will use them – no matter how technically proficient they are. Thus, DEPTH should ease the creation of complex simulations and should provide primitives necessary to create and animate complex tasks. DEPTH should also simulate several humans working cooperatively or independently as appropriate. In addition, DEPTH should evaluate body fit, reach, and strength requirements, as well as predict overall task performance time.

Approach/Discussion

In Phase II, an automated task composition capability was added to DEPTH. The intent was to portray purposeful behavior according to a logical plan of action. It was intended to be capable of acting out task sequences in realistically timed motions, appear to react, plan, detect obstacles, avoid uncomfortable or inefficient postures and movements; and so on. The capability was to allow detailed and accurate simulation of complete maintenance tasks so human performance requirements could be estimated.

The Human Operator Simulator (HOS) by Micro Analysis and Design, Inc. was selected to create task sequences. Although HOS was a good tool for task composition, it was not well suited to control low level Jack motions. Thus, HOS was removed in Phase III to allow a more appropriate task composition tool to be integrated. The government determined that the functionality already integrated into Jack would better fulfill the automated task composition requirement.

The first motion models created in Phase IV were simple but provided the foundation for more complex simulations. These models were implemented with Jack animations which were easy to work with and robust. The Jack animation window could be used to reorder motions; change start and end times (duration); and run two or more motions concurrently. Unfortunately by the end of Phase IV, only part of the motion model library was complete and some models did not run correctly. None of them were performing the kind of high level tasks envisioned and the use of multiple humans was not supported.

At the start of Phase V, it was recognized that DEPTH's requirements could not be met with the animation functionality. The most significant problem was that analyses could not be performed as animations ran. Thus, a Lisp-based simulation capability created for Jack called Parallel Transition Networks, or PaT-Nets, was used [6]. A network-based editor for PaT-Nets (called Visual Jack) was also under development. However, late in Phase IV, we found that it fell short of expectations, so the DEPTH team decided to develop a simple step sequencing capability (Figure 4). To minimize software complexity, only one step could run at a time; one step had to complete before the next step could start.

As the motion models and DEPTH code were further developed, it became obvious that significant user interaction would be needed to create complex tasks. For example, we hoped to provide a utility to simulate the removal of any Line Replaceable Unit (LRU). For this to be possible, information about the quantity, size, and location of fasteners and cables would be needed. Unfortunately, CAD files translated into DEPTH usually contain only geometric information. Even if all of this information was extracted, DEPTH does not currently understand how parts are assembled nor where parts should be grasped. To calculate this automatically requires knowledge of where the human is located, the direction from which he or she is approaching, whether to use a handle, and which way is up. It also requires a significant amount of human behavioral information.

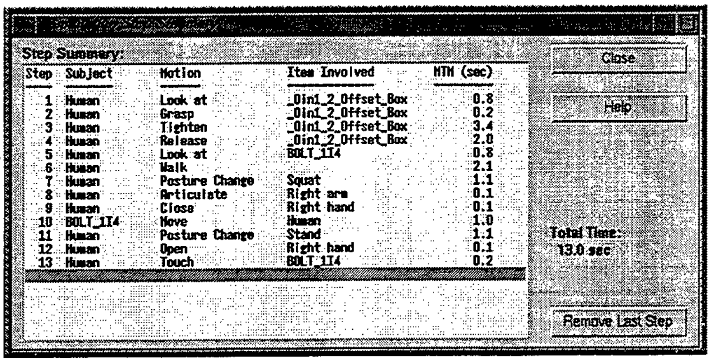


Figure 4: Simulation Script

As dialog boxes were designed, it became obvious that it was impractical to enter all of the information needed to define complex tasks, so an alternative approach was devised where users could use tasks to build tasks. Each series of tasks could be logically grouped and saved as individual files. Once saved, they would be available for future procedures requiring the same task to be performed on the same object.

It was observed that this concept could be extended. If the steps were saved independently from specific objects, the simulation steps could be applied to different objects. If the object has a different name, the user could be prompted for the new name. Thus, the DEPTH workspace file – containing information about the environment and all of the figures in it – is saved independently from the simulation file. Additionally, simulation functions become part of a separate menu structure.

Another goal was to prevent erroneous data input. Extensive field disabling was implemented on the dialogs and on the menus to force the selection of only valid options. For instance, if the human has grasped a tool, another object cannot be grasped with the same hand in the next step. Similarly, the user cannot create, delete, or rename figures while creating a simulation. Such modifications could alter the state of the environment expected by the simulation. For example, if a tool were used in the first step, then later deleted, the simulation would not be able to find the tool. This restriction was generalized to prevent environment alterations (other than the view) while in simulation mode. Further, when a simulation file is opened, the state of the environment is scanned. Thus, before the simulation ever starts running, the user can be prompted to choose a comparable figure to replace a required figure.

After these restrictions were in place, automatic tool selection (an existing DEPTH function) was studied to determine if it should be incorporated into the simulation capability. With this function, tools are auditioned by automatically creating them, checking their sweep range, and then deleting them. Since this is an alteration to the environment, it could cause simulation inconsistency. The user can still seek tool selection guidance independent of the simulation, or simply perform a loosen fastener simulation. The latter is a more meaningful sweep evaluation than the tool selection function since the human is actually holding the tool. The automatic tool selection does not factor in the space required for the hand and arm.

As Phase V progressed, lifting, lowering, pushing, pulling and cooperative work (multiple person) models were added. Since more than one human was moving simultaneously and motions involving single humans were only allowed to run one at a time, humans would not have to actively avoid each other while working on independent tasks. The walk model allowed humans to avoid other figures (including humans), but this capability was not made available to avoid burdening the user with more parameters.

Enhancements were made to the motion models included the following items:

- Collision checking between humans and objects
- Checking for blocking of the human's line of sight
- Checking for the human being unable to reach the desired location
- Strength analysis, using MIL-STD-1472D information
- Calculation of elapsed task time using Motion Time Measurement (MTM)

Lessons Learned

The implementation of generalized high level motion models was greatly hindered by the fact that attributes (other than geometry) are usually not transferred in from CAD. Without such information, the user must prepare the environment – such as placing fasteners and handles – before creating the simulation.

When working with objects such as hand tools, inverse kinematics makes it necessary for the arm to follow the tool rather than the other way around. This approach often does not produce human-like movements. Transom is developing dynamic articulation into Jack which should produce more natural movements.

2.2 Strength Analysis

Objective

In addition to reach accessibility, it is also important to determine that people are strong enough to perform a task. Lifting, pushing, pulling, carrying and torquing (turning with a wrench) are tasks that often require a considerable amount of "muscle." Strength depends greatly on the individual and his or her posture. For example, a maintainer who is forced to lean into an access opening would not be able lift as much weight as he or she could in a more comfortable posture.

DEPTH should provide a way to assess the strength based on the best information available. MIL-STD-1472D [9] and algorithms from the Crew Chief program [7] are two excellent sources of such information. Studies performed for Crew Chief were particularly well suited for maintenance task strength. These analyses should be performed automatically as a simulation runs.

Approach/Discussion

In early DEPTH versions, Crew Chief strength analyses were not integrated with the simulation. Instead the user had to input parameters required by the strength algorithms in a dialog box (Figure 5). The objective of the final development phase was to calculate strength limitations with little or no user input. Thus, the strength function parameters should be extracted and automatically passed from the simulation reducing analysis time, errors and frustration.

With funding from the Logistics Research Division, the University of Dayton Research Institute created a C interface for the Crew Chief strength functions. The interface was primarily created to allow the strength functions to be used independently of the Crew Chief model, and also added flexibility in the data input. For example, the previous interface only allowed a rectangular object and the parameters included the height, width, and depth of the figure. With the new interface, any shape object is allowed and the parameters include the dimensions of the bounding box needed to completely surround the object. In addition, the data look-up algorithms for Crew Chief were modified to interpolate between data points, extrapolate beyond data points, and flag out-of-bounds parameters.

The lift, push, pull, and carry models were changed to incorporate these strength calls. During a simulation run, DEPTH has information about what type of motion is being attempted and how the figure is oriented relative to the human. Other information is also available, such as the human's stature and the figure's mass.

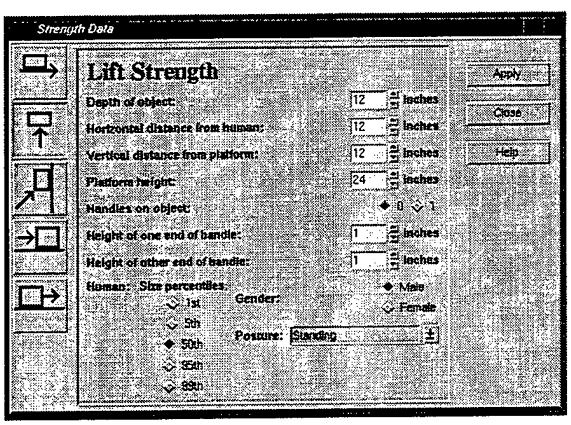


Figure 5: Lift Strength Dialog Box

A number of assumptions had to be made to supply the strength functions with all of the required information. Each parameter uses a different reference point on the human for distance measurements. For example, lift used a point between the feet. The parameters for each function are described below.

<u>*Lift*</u>

The lift function was also used for lowering motions. As shown in Figure 6, the *position reference point* for lift depends on posture. For example, when standing the posture reference point is located on the floor between

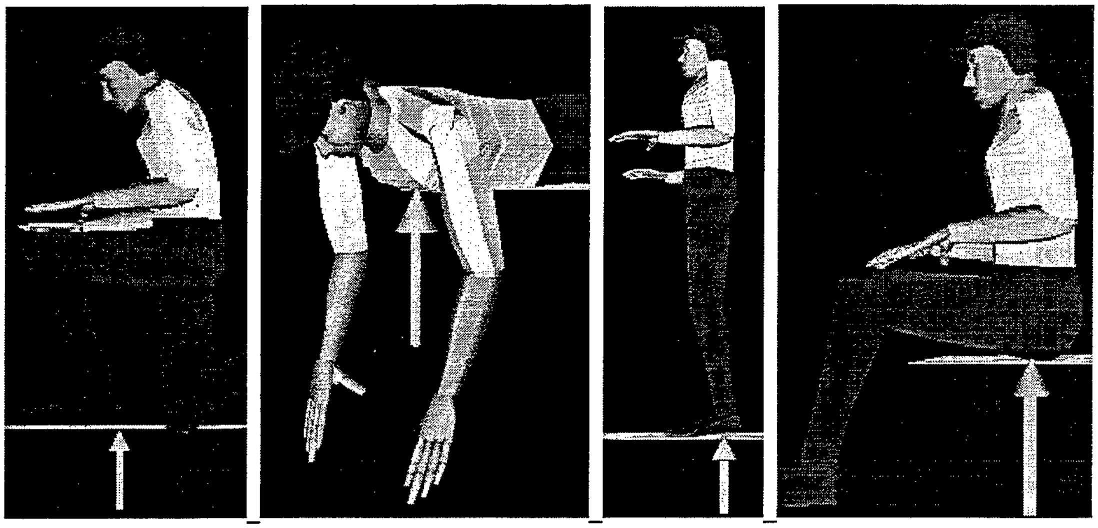


Figure 6: Example Position Reference Points

the two heels and when prone it is the point on the floor in the middle of the shoulders. The position reference point is always a point on the "floor" (or the human's platform). As with all of these functions, the output was safely conservative. The arguments to this function are:

- *dist_away* was the horizontal distance from the position reference point to the object. Since it was not possible to determine where the object's front face was, the grasp site was used. If both hands were used, the midpoint between the two grasp sites was used.
- *elevation* was the vertical (y axis) distance from the human's support platform to the bottom of the object at the end of the lift. The target location – where the object was lifted to – was considered the bottom of the object.
- *height*, *width*, and *depth* were the dimensions relative to the human of a bounding box surrounding the object. The pelvis orientation was used to determine the direction the human was facing.
- *num_handles* was "0" if two hands were used and "1" if one hand – with the handle on top of the object. We could not find a reasonable way to determine whether the handle was on top.

<u>*Push and Pull*</u>

DEPTH calculates the force as the product of the object's weight and the static coefficient of friction between the object and the surface under the object. This equation only applies to horizontal motion on a non-lubricated flat surface. The arguments to this function are:

- *dist_away* was the horizontal distance from the human's reference point to the center of the human's hand or hands. If only one hand was used, DEPTH provided the distance in the x-z plane from the human's reference point to the grasp site. If both hands were used, the midpoint between the two grasp sites was used to calculate the distance from the reference point in the x-z plane.
- *elevation* was the vertical distance from the human's support platform to the center of the human's hand or hands. The hands are required to be at approximately the same height. If only the left hand only was used, DEPTH provided the distance in the y direction between the support platform and the center of the human's left hand. Otherwise, the distance in the y direction between the support platform and the center of the human's right hand was provided.

- *force_type* was controlled or uncontrolled. Controlled means that if the object can move vertically, the human performing the task was required to only apply horizontal forces. DEPTH always assumes uncontrolled force.
- *elbow_angle* could either be straight or bent. Straight means that the elbows are locked straight. If the left hand only was used, DEPTH checks the left elbow angle. Otherwise, the right elbow angle was checked. If the elbow angle was less than or equal to 5°, then straight was provided, otherwise bent was provided.
- *friction_coef* was the friction coefficient between the maintainer's boots and the support platform. This value, as well as the friction needed for the force calculation, was user specified. Defaults of 0.8 and 0.3 respectively, were provided for coefficient and force calculation.

<u>Carry</u>

The arguments to this function are listed below.

- *height, width,* and *depth* were the dimensions of the bounding box surrounding the object and oriented relative to the human. DEPTH takes the human's pelvis orientation as the direction the human is facing. The bounding box surrounding the object was calculated using this orientation.
- *num_handles* was "0" if two hands were used in a standing posture, "1" if one hand grasped a handle on top of the object, and "2" if the object was to be carried with two hands in the crawling posture. DEPTH provides "0" if two hands are used and "1" if one hand is being used. DEPTH had no way to determine whether a handle was on top.
- *ceiling_hgt* was the vertical height of the ceiling relative to the human's support platform. DEPTH provided the human's stature, making the ceiling unlimited. DEPTH had no concept of ceilings.

<u>General</u>

- Each strength function had different way to assess posture. DEPTH simulation limits the human to either standing or squatting. Therefore, when a posture was required, DEPTH either provided standing or squatting.
- Since the presence of gravity was assumed, object mass was converted to weight.

Lessons Learned

Some assumptions and complicated calculations were needed for this implementation. The calculations were difficult to program and often required significant processing time – sometimes several seconds for a single simulation step. In any case, it should be noted that DEPTH provides a *reasonable estimation*, but actual strength could be affected by factors not simulated.

2.3 Hand Model

Objective

Almost all maintenance involves working with the hands. Some tasks require powerful grasping and others require precise work with the fingers, thus, DEPTH should provide realistic, articulated hands. Reach planning, strength analysis, collision avoidance, automatic reaching and grasping should be available for realistic simulations. Reach planning based on a grasp location and planned use of the object should be provided. Finally, hand tracking devices should be supported since they provide a powerful method to control hand motion.

Approach/Discussion

Prior to Phase III, Jack did not provide a detailed hand, so in Phase III, graduate students at Penn conducted studies to determine the best way to model hands. The results of these studies were used in the design of the hand model currently in Jack. Each finger including the thumb was modeled with three segments and three joints. The phalanges include proximal, middle and distal segments while the thumb includes the first metacarpal. The palm geometry was shortened to consider the proper connection sites with the digits; the webbing between the fingers was not visually depicted. Although this made the fingers appear long and palm appear short, it made the simulation of accurate motion limits more practical.

In addition to hand anthropometry, R&D was also conducted in hand motion. For reach planning, a sequence of collision-free, yet strength feasible, motions can be orchestrated to permit movement of the hand to the goal position. Grasping involves several simulations (PaT-Nets) performed in parallel that are reactive to the surface geometry of the grasped object. Specifically, the fingers and thumb search for an optimal position as they touch the target object. This grasping algorithm determines which of sixteen standard manufacturing grasps – falling under two major categories of power and precision – should be used.

An interface was developed to both the CyberGlove by Virtual Technologies and the DataGlove by VPL. The CyberGlove interface was far more accurate and reliable (probably because the CyberGlove was a more mature product) thus this interface was maintained throughout the remainder of the DEPTH program. The DataGlove interface was not supported in later software versions.

Lessons Learned

Implementing the enhanced hand model greatly improved the appearance and functionality of the hands [4]. However, more development is needed to improve hand motion for autonomous simulation. The CyberGlove interface worked well, but the tracking accuracy was not perfect. Nevertheless, a tracking glove provides a useful approximation of hand motion.

2.4 Flexible Anthropometry Definition

Objective

In designing equipment that humans will interact with (practically all equipment), one should consider differences in body dimensions. A person with short arms, for example, will not be able to reach everything that someone with long arms can. Conversely, large people wearing bulky clothing may not be able to fit into tight spaces. So DEPTH should provide a straightforward method to create humans of varying sizes. It should be able to appropriately handle incomplete or even inaccurate data and a statistical analysis should be performed to assess accuracy.

In addition to physical dimension information, DEPTH should allow strength, mass, kinematics, and other such characteristics to be modified. Some of the data used to evaluate a maintenance task will be a function of characteristics such as size or gender, and may vary across populations. In addition to generating a human from population data, DEPTH should allow specific attributes to be input directly. This could be useful in creating a model of a specific person or for characteristics that may change over time such as strength due to fatigue.

Approach/Discussion

Defining accurate anthropometry is complex, so a multiple-tabbed GUI was designed to simplify the task for the user (Figure 7). The main tab provides selection of the most generally-used characteristics to allow quick selection of a suitable human. Other tabs allow the more experienced operator to customize a human as required for the task being simulated. Additional features incorporated on the GUI include joint manipulation, hiding segments to give the appearance of missing limbs, and adjusting color attributes. Collision detection dramatically degrades simulation performance, so the user was allowed to determine what segments of the body are checked for collisions.

Defining initial postures with varying anthropometry was not simple. One approach considered was to generate a different human for each size variation, but this was found to be too onerous. This approach was used in early versions since the Crew Chief model files were read in, but it could not be used to

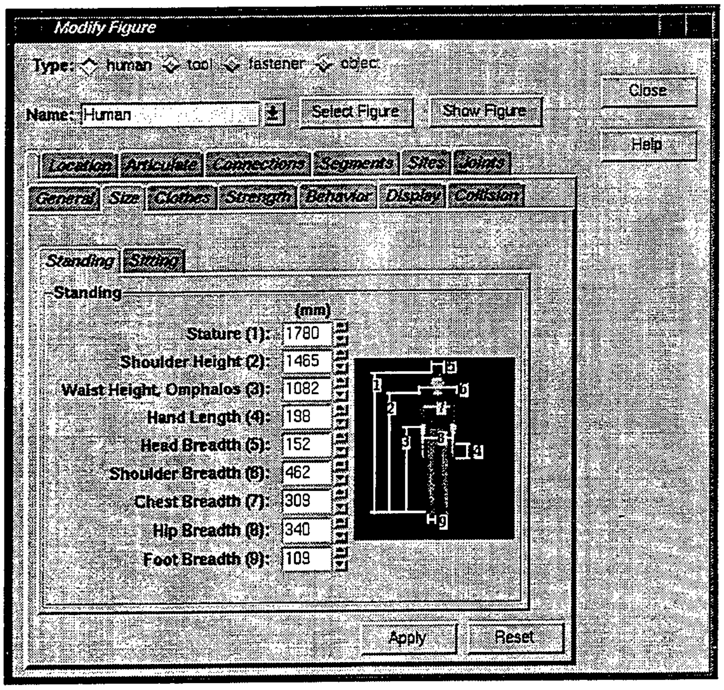


Figure 7: Anthropometry Dialog Box

flexibly scale anthropometry. A better approach was to define postures with predefined joint angles independent of segment lengths. In addition to flexible scaling of anthropometry, this approach would be used for batch human factors analyses. Specifically, a simulated task could be automatically repeated with varying anthropometry. Unfortunately this approach forced us to consider new methods to model bulky clothing.

The BodyShop anthropometry scaling system – created by Penn for this project – supports creation of diverse anthropometry using several different methods. Before BodyShop, Jack used segment measurements from joint centers to generate human files. While convenient for software implementation, this did not relate well to normally-accepted anthropometric measurements. BodyShop used actual anthropometric landmarks as opposed to joint centers. Since landmarks were not standardized, a set was selected from MIL-HBK-759B.

To increase flexibility, an interface was provided to allow anthropometric databases to be installed. This included subject-based data as well as percentile-based data.

Lessons Learned

Jack's original method to define anthropometry was actually quite flexible – almost too flexible. For DEPTH's functionality, we attempted to make this functionality more logical to the user. It is now possible to define a human model based on widely accepted body measurement landmarks. A user can use percentiles[2] or enter dimensions from individuals for a multivariate approach.

The use of widely-accepted anthropometry measure points to scale a template figure worked out well for several reasons. From the user's standpoint, it provided a flexible yet logical way to represent humans. Template figures also made the software implementation more logical. Instead of numerous special models to access, the developer only needs to maintain one template. It minimizes storage requirements since only template figure files are required. It also expands the options that are available for enhancements such as the batch processing discussed above. The interface for batch processing and design analysis was designed and included in the DEPTH documentation.

[2] Many human factors experts no longer consider percentiles a valid approach. This is because a human made up of all 95th percentile segments (for example) does not necessarily have a 95th percentile stature. It is considered more appropriate to choose individuals from the population who are nearest the target boundary. This is accomplished using a multivariate approach where the variable data are combined or reduced into fewer component values.

2.5 Ambient and Object Properties

Objective

An important part of task analysis is safety. Many types of existing hazards can be simulated. These hazards may either be ambient or associated with objects [8]. Ambient factors – extreme ambient temperatures, vibration of floor, humidity, precipitation, and noise – can affect fatigue, stress, safety and strength. Because object hazards such as heat, sharp edges and hazardous chemicals can cause serious injuries thus protective gear may be required. DEPTH should recommend appropriate protective gear and simulate the restrictions it imposes.

The user should be notified before attempting to perform an unsafe task. If the human model is directed to handle toxic materials, for example, protective clothing should be recommended. If there is a limited duration for exposure to the substance, the simulation should require another technician to complete the task or require a timed break. Repetitive motion injuries such as Carpal Tunnel Syndrome should be assessed based on body position and number of iterations.

Approach/Discussion

Since the information was not available from CAD, we allowed the user to set levels for each factor. Ambient factors are applied to the entire environment. For instance, when noise is set to a particular level, the entire workspace is assumed to have the same noise level. In the case of object properties, the values are applied for each object instance created. For instance, one box can be 50°C, while another 10°C.

DEPTH displays safety warnings as soon as appropriate. Ambient factors are calculated when a human is created. If the human wearing fatigues is inserted into an extremely cold environment, a dialog box appears stating that cold weather clothing is recommended. Object factors, on the other hand, are calculated when a touch or grab is performed.

A study was performed to determine what information was available regarding object hazard exposure and how this should be presented in DEPTH. The resulting report required data to be in the form of a "yes" or "no" answer, such as "What noise level is unsafe?" We also researched the best message to provide in the warning dialogs. Some of the data gathered satisfied these requirements, but no data could be found for some factors. The data available was sometimes dependent on exposure time. Unfortunately, these times ranged anywhere from fraction of a second to over a year. This made implementation prohibitively difficult.

Ideally some of the ambient environmental factors, such as noise and radiation, would have been part of object properties. These factors emanate from a source and diminish when moving away from the source. To fully simulate this effect would require measuring the distance between the human and the source, and then calculating the level at that distance. Determining the cumulative effect would require continuous calculations. The effects of shielding (complete or partial), reflection, and the additive effects of multiple sources can affect results. However, it should be possible to develop reasonable estimates in most cases without such elaborate calculations.

Protective clothing, such as parkas and chemical defense suits, limit the mobility of a human's limbs. Unfortunately, we were unable to locate data to quantify mobility limitations with protective clothing. Once the data is available, joint limits in Transom Jack can simply be set according to the clothing type.

Lessons Learned

This implementation turned out to be more difficult than expected, so we only achieved limited functionality.[3] Surprisingly, data to support full hazard analysis was not readily available. For example, no

[3] Note that when the simulation portion of DEPTH was rewritten for Phase V, the object properties to check for grab and touch were lost. Due to budget and time constraints, re-implementing this functionality was not possible.

data could be found to assess how fatigue or stress affects performance. Perhaps this lack of data was due to the fact that exposure effects often depend on many variables. These complex interdependencies make it nearly impossible to precisely simulate these factors, however, we feel that *estimating* hazardous conditions is useful. Considering the *most significant* conditions is better than nothing at all.

2.6 Logistics Data Capture

Objective

One of the most important aspects of any design process is analyzing and documenting support requirements. In the past, the government placed rigid requirements on what analyses were performed and how they were documented. This process, formerly called Logistics Support Analysis (LSA), was made less stringent to reduce costs and increase flexibility during acquisition. Although the latest evolution of this process is less stringent, many contractors find it necessary to perform most of the original process anyway.

DEPTH should be able to exchange information with the Enhanced Automated Graphical Logistics Environment (EAGLE) database using a Structured Query Language (SQL) interface. EAGLE, commercially available from Hughes Aircraft, is a MIL-STD-1388-2B validated software product that works in conjunction with the Sybase database management system. EAGLE provides a method to organize and store logistics data needed throughout the life of a large system. With some modifications to the SQL, it should be possible to exchange data with similar logistics databases.

Approach/Discussion

The focus was to populate EAGLE fields directly from DEPTH, and when possible, without user input. A user interface was designed to obtain key field values required to make updates and queries. A goal was to make this functionality readily available yet unobtrusive if it was not needed.

Once a connection was established with EAGLE, the interface to DEPTH could be used. The user made selections from information in the database when establishing the workspace environment. Some input fields were provided for more direct database input. Figure 8 shows the source for the information used to populate the database relative to the data element definitions (DED).

The user was assisted in describing the task as needed for logistics data capture. Once the simulation was in process, the tasks and subtasks were analyzed for performance times. Upon completion of the simulation, the entire workspace was analyzed for crew size and item quantities. All of this information was then sent to EAGLE. Figure 9 shows the table relationships used by DEPTH and the DED numbers for the fields populated. A supplement log is generated to review database changes.

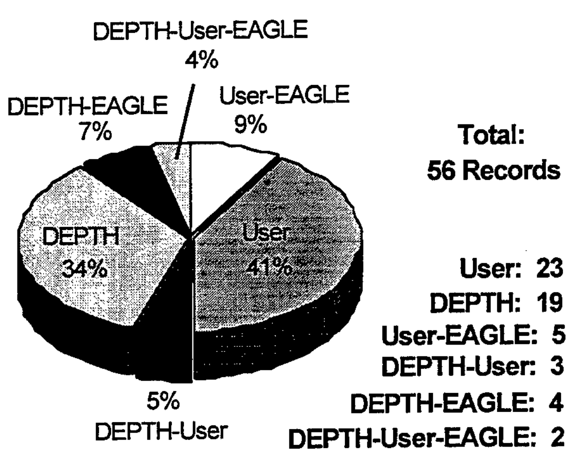
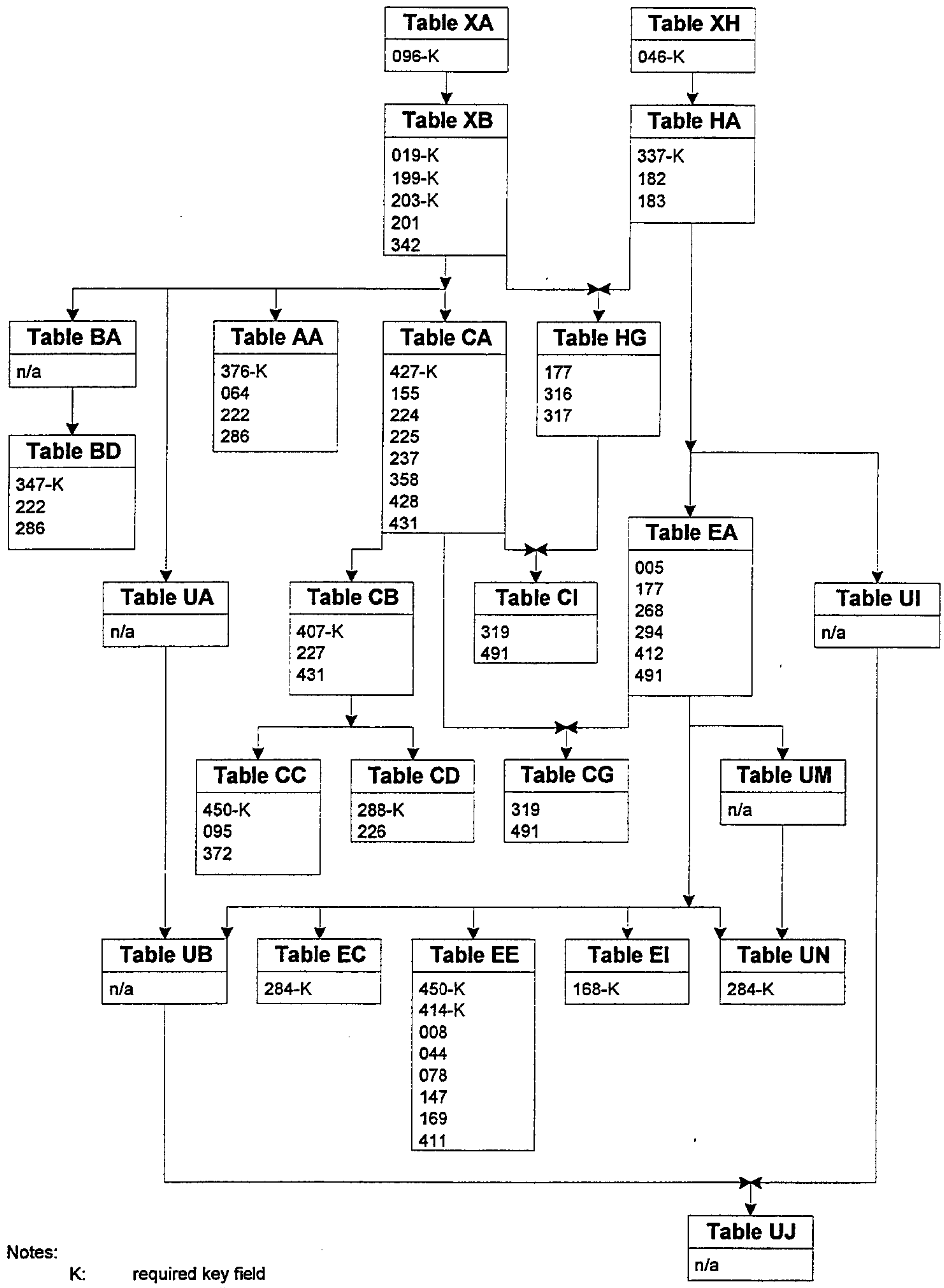


Figure 8: DEPTH Code Statistics

Lessons Learned

When integrating this new logistics functionality into DEPTH, it was found that the Jack software terminated whenever EAGLE was accessed. The problem was traced to differences in the UNIX shells. DEPTH was started in the C shell and Jack was started in the Bourne shell. Both Sybase SQL commands and Jack used the SIGIO signal for communications, causing the Bourne shell process to be terminated. When DEPTH was started in the Bourne shell, there was no problem. Another alternative was to mask the SIGIO signal. While no problems were encountered, such practices were considered



Notes:

K: required key field

n/a: no new DED item but table is required

Figure 9: DEPTH LSA Table Relationships

risky by Sybase's Support Group. The alternative solution that was to start DEPTH in the Bourne shell within the script.

The interface between DEPTH and EAGLE was limited to one module. If a database other than EAGLE would be used, only this module would need to be changed. If the logistics interface were made a part of the DEPTH software, a Sybase code development license would be required to obtain the libraries for linking with the object code. Without the license, it would not be possible to modify this functionality. To overcome this obstacle, the logistics analysis code was made into a library linked with DEPTH. Thus, changes could be made independent of the Sybase libraries.

When populating the database, key field information is required to modify table contents. One major key field is the Contractor and Government Entity (CAGE) code. Rather than having the operator complete this field, a fictitious CAGE code of "DEPTH" was used. This choice has two purposes: minimize the input fields to be completed by the user and provide a convenient way to locate updates made by DEPTH. The logistician could then verify that the entries made from DEPTH were correct .

Unfortunately, we were not able to transfer pictures from the simulation to the EAGLE Z-tables. This feature would have been useful for the automated generation of interactive electronic technical manuals. The problem encountered was a bug in the software that converts the Silicon Graphics, Inc. (SGI) RGB pictures to a format compatible with EAGLE.

The next logical step was to "reverse the process" by generating workspaces and simulations based on the information contained in a logistics analysis database. Since the interface was already created and key components already existed, a workspace could be populated with minimal effort, but creating simulations from the logistics information would not be as simple. In short, it would be far easier to change the model than the simulation.

2.7 Maintenance Training

Objectives

Limited resources are available for classroom maintenance training. Hands-on experience using actual aircraft is ideal for the trainee, but requires aircraft to be available for them to work on. Simulation could provide an easy and safe method to teach maintenance tasks. DEPTH could be used to demonstrate tasks for maintenance training. These simulations – captured prior to production of hardware or mockups – should allow training media to be ready when the system is ready without the expense of video production. As a minimum, DEPTH should make it easy to generate movie files that can be played on personal computers. DEPTH should also be used to create simulations "on-the-fly" or allow interactive training using virtual reality.

Approach/Discussion

The ability to create movies during simulation playback was added to support DEPTH's training aspects. Once a simulation is created, the user can easily record the simulation run to a movie file that can be replayed with a QuickTime® Viewer. SGI and Moving Picture Expert Group (MPEG) movie files can also be created. The various movie file parameters such as file type, size and frame rate can be set using the preferences menu. Movie parameters such as format, compression algorithm, frame width and height, frame rate and the directory to use for temporary files may be specified on the "Movie" tab of the "User Preferences" dialog box.

Because of its close ties with simulation, "Record..." was added to the simulation menu. The record function makes the simulation run and periodically captures frames using the Jack "save_screen" command. Capturing of movie frames is accomplished with a PaT-Net so that it may be run in parallel with the simulation. The user's preferred frame rate dictates how often the "save_screen" command is issued. After the simulation ends, the frames are "packaged" into a user-specified movie file and then discarded. The ".mov" file is created using the SGI utility *dmconvert*.

A prediction of the resulting movie file size can be made from the "Movie" tab of the "User Preferences" dialog box. These predictions are based on a small case study and are most likely inaccurate, especially because the amount of compression attained depends upon the data being compressed. The inaccuracy of these predictions prevented the design goal to stop movie file capture when low on disk space from being implemented.

Lessons Learned

Capturing frames during a simulation run could significantly affect performance, especially for high frame rates and large frame sizes. However, since movies will often not be recorded until a simulation is perfected, this performance degradation may be acceptable. Movie recording also requires disk space two to three times the size of the resulting movie file because frames are stored in a graphics file before they are compressed. Real-time compression in Jack would help reduce disk storage requirements.

2.8 Body Tracking Devices

Objectives

Immersive technology refers to the hardware devices that may be used to immerse a person into DEPTH's virtual environment. The benefits to becoming a player in the simulation by using body tracking devices and head mounted displays are significant. One such body tracking device is Ascension Technology's Flock of Birds®. DEPTH should allow the body tracking devices to control the limbs of the virtual human, thus allowing the virtual human to mimic movement of a user.

Approach/Discussion

Because the Hughes development team did not have access to the Flock of Birds hardware, the body tracking code was written at the Logistics Research Division. The Flock of Birds utility was written as a separate executable which could be run in parallel with DEPTH. The transmitter and receiver properties and placement could easily be configured using a "wizard" style setup dialog box. Receiver position information is streamed directly to Jack using the documented Remote Procedure Calls (RPCs).

Lessons Learned

Integration of the Flock of Birds functions was simplified by the fact that it was kept as a separate executable and the whole development team was involved in the design. Only minor changes were needed to the dialog boxes to maintain DEPTH's look and feel.

2.9 Cable Generation

Objectives

One of DEPTH's users noted that the disconnection and connection of cables to LRUs is a very typical aircraft maintenance task [5]. As detailed in this report, the user should be able to easily generate a cable of a desired length, diameter, and flexibility. In addition, the user should be able to select a TNC-Knurled, TNC-5/8", TNC-9/16", electrical, or no connector for each end and select from a straight, 90 degree, or 45 degree backshell for each connector. DEPTH should provide simulation primitives for connecting and disconnecting the cables. Strength and collision feedback should be given during the execution of those motion primitives.

Approach/Discussion

The cable generator was developed by AL/HRGA and provided to Hughes for integration. Most of the cable parameters documented by Lockheed Martin plus resolution (number of facets in each cable segment) and color were implemented. Hoses with definable compression and wave guides with width and height parameters were also added.

After final receipt of the cable generator code, several changes were required. Because the code would be compiled and linked with the main executable, all function names and global names had to be prefaced with "CABLE_" to prevent global scope naming conflicts for the linker. All sites created by the cable generator for connector attachments had to be created through the site creation functions so they could be tracked in the software. The figure file created by the cable generator had to be inserted into the DEPTH workspace at the proper location using the DEPTH insertion functions. The attachment and constraint of the cable had to be changed to use DEPTH's connection functions so that DEPTH would be aware of the connections and the user could delete them if necessary.

Motions to connect and disconnect cables were added toward the end of the Phase V. These motions can be found on the "Object Use" dialog box. Collisions will be reported for figures identified as obstacles for the human working with the cables, but strength analysis is not yet available. Because DEPTH provides cable connection and disconnection as motion primitives, the connection and disconnection order feature suggested by Lockheed Martin was not implemented.

Lessons Learned

The functions used to generate the cables, hoses and wave guides were robust and created cables as specified. The GUI was also well organized. However, unlike the Flock of Bird code, integration was required.

2.10 User Interface

Objective

Modern software packages should provide a combination of novice-oriented interfaces as well as shortcuts for experienced users. Novice interfaces are important for DEPTH users who tend not to use the software on a daily basis. All commands should be accessible through menus, icons, and dialogs. The dialogs should also lead the user to valid parameter entry with graying logic (disabling irrelevant fields) and entry limits. On-line help should be available for every dialog box. Quick access "hot keys" and macros should be provided for power users. Cascading menus and the number of choices available on each menu was minimized.

Approach/Discussion

Early versions of DEPTH assaulted the user with multiple, nonstandard windows. In Phase III, the DEPTH team (Air Force, Penn, and Hughes) designed a new approach based upon widely accepted interfaces. The result was a more intuitive and useable GUI.

At the same time, Penn was developing a new application programmer's interface (API) for Jack to eliminate the need for two-way communication sockets. A socket is a communications port that allows inter-process communication between software modules. It had been determined that the incorrect usage of sockets in Version 2.0 created many instabilities that caused crashes. By replacing sockets with RPCs and a newly-developed C-API, communication with Jack was much more reliable. The C-API supported the Lisp calls that were previously accessed via sockets; therefore, integration was transparent. The C-API Lisp calls eliminated the large character buffers needed to construct the Lisp commands and parse the returned strings.

In the past, Jack was controlled using a command line by typing one or more key words. In addition to other benefits, DEPTH added friendly menus, buttons and dialog boxes while maintaining the command line. The Tcl/Tk scripting language was used to build the interface. Command processor procedures were added to provide DEPTH commands to the Tcl/Tk language. All dialog boxes (except error, warning, and informational) include a "Help" button which displays a HyperText Markup Language (HTML) help file in the Netscape browser. Dialog box items are disabled (appearing gray) or unpacked (disappear) if they do not apply. Entry widgets have pre-set upper and lower bounds to prevent input from being out of range. Color icons provided by the customer allow quick access to many of the common commands. Much of the user interface can be customized by the user with the *preferences* dialog.

At the start of Phase V, Air Force representatives met with the design team in a two-day menu design meeting. The objective was to build on lessons learned to create an even better design. The result was the GUI for the final release (Figure 10).

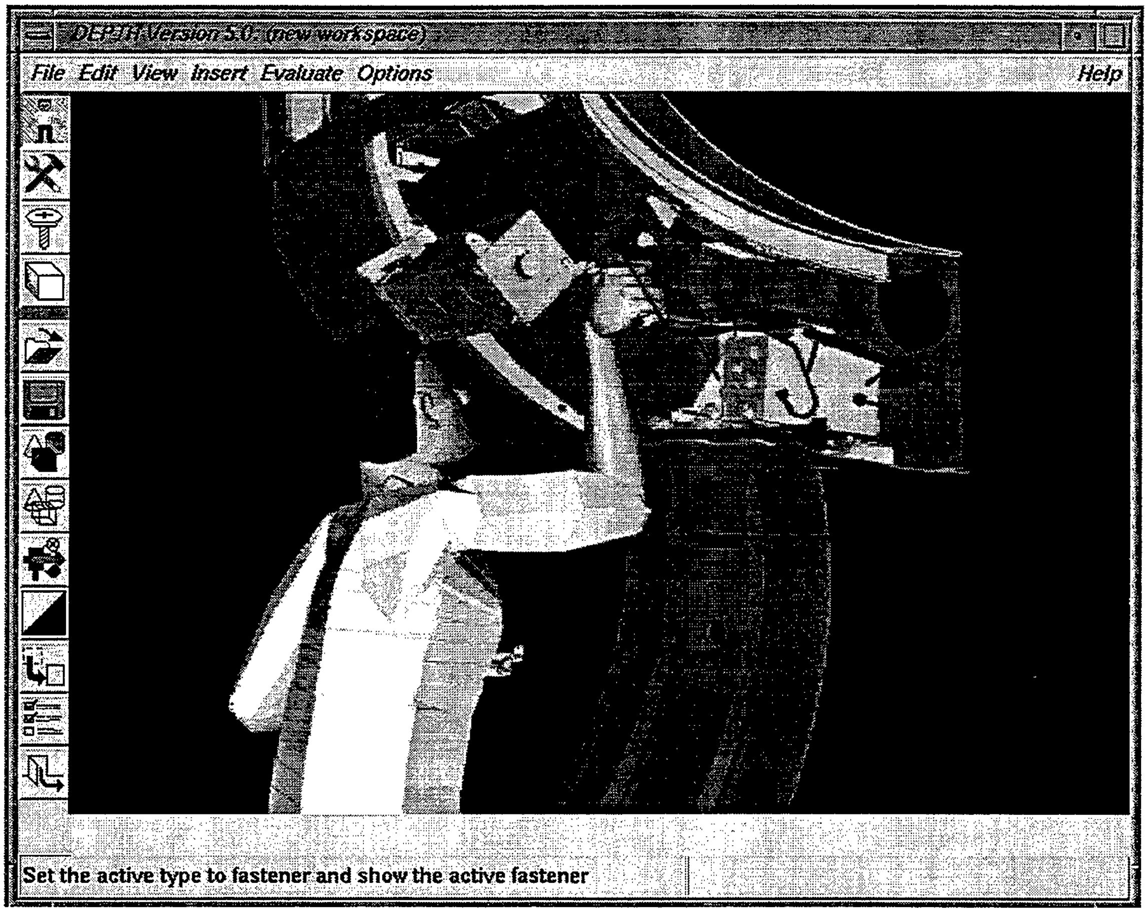


Figure 10: DEPTH Screen Layout

Two major modes are available in DEPTH, each with its own menu structure. The main menu for environment modeling provides the categories *file*, *edit*, *view* (manipulation), *insert*, *evaluation*, *option*, and *help*. When simulation mode is entered, the menu structure changes to provide appropriate functionality.

Lessons Learned

The goal was to make DEPTH and Jack look like a single application, but this was complicated by the fact that there were two independent processes. Jack was started and controlled by DEPTH using RPCs. To appear as one application, DEPTH positions the Jack main window and message window (both with borders removed) within the DEPTH main window. This is most noticeable when moving the DEPTH window because the Jack window appears to chase the DEPTH window rather than moving with it. Other windows may also slide between the two windows. At startup, the user will notice that the Jack display cannot be controlled and various windows appear. While these problems are annoying, they are not necessarily critical. All of them could be eliminated by developing a better interface between the two processes or by merging DEPTH and Jack into a single process.

The implementation of Microsoft Windows style hot keys was also hampered by the existence of the two processes. Tcl/Tk allows the use of hot keys. However, since the Jack window is positioned on top of the DEPTH main window, the key strokes may be intercepted by Jack and never received by DEPTH. This characteristic prevented the selection of Jack figures for manipulation by DEPTH making a number of GUI operations less intuitive.

Several GUI items were not implemented due to resource or technical constraints:

- A tool bar in simulation mode
- A uniform look and feel of all dialogs
- Meaningful color icons in place of the prototype icons
- User-defined macros that can be invoked by pressing a function key

The Tcl/Tk language [10] used for the GUI is a great prototyping language. Sample GUIs can be designed quickly for prototyping. However, Tcl/Tk lacks the robustness and flexibility provided by an X-window language such as Motif. The available Tcl/Tk widgets required customization to address our complex implementation requirements. For instance, since the tabbed GUI widget was designed for a single row of tabs, the multiple row tabbed dialog required a novel toggling scheme to give the appearance of switching between the rows. Occasionally, the dialog boxes did not fully expand to show all the fields. Some widgets invoke commands twice when a selection is made. In addition, since Tcl/Tk is interpreted during execution (rather than compiled), it tends to be somewhat slow.

To reduce menu choices and limit cascading menus, tabbed widgets were added to most of the dialog boxes. This shifted the balance from a complicated menu structure to complicated dialog boxes. Often dialog boxes contain rows of tabs. The Modify Figure dialog box (Figure 11) includes a set of four radio buttons which change the set of tabs that appear at the bottom of the dialog. Also, the number of tabs required to support editing of information about humans mandated expanding the row of tabs to two tiers. The complexity of the dialog boxes significantly increased the complexity of the code required to implement them.

2.11 CAD Translation

Objectives

One of the key features of simulation software is the capability of working with existing data sets. The graphical systems should be able to import 3-D CAD drawings. This is the ideal method to obtain "virtual mockups" since the CAD data will already be available from the design process.

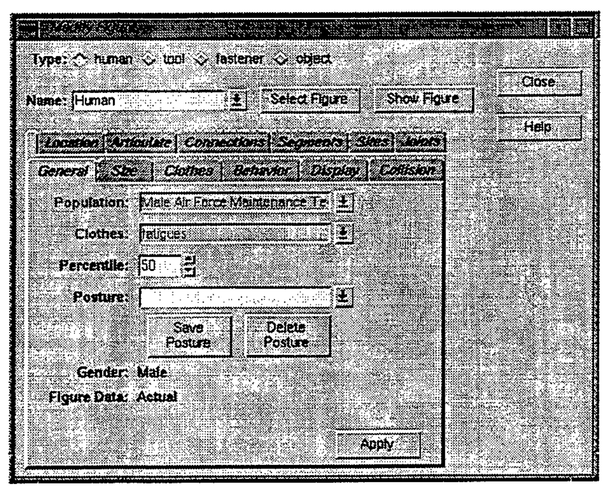


Figure 11: Modify Figure Dialog Box

Approach/Discussion

During Phase II, a CAD translator was incorporated directly into DEPTH. This unsupported translator quickly became obsolete and was removed in Phase III. During Phase III, access was incorporated to an external IGES translator provided by Penn. This IGES translator, however, suffered from frequent crashes and was removed in Phase IV. At that time, a command line driven Pro-Engineer translator, developed at Penn, was integrated directly into the Insert File dialog box and remains in place today. In Phase V, a command line Inventor translator, again developed at Penn, was also incorporated directly into the Insert File dialog box. SGI provides a suite of translators from various CAD systems into their Inventor format. Using these translators, the DEPTH analyst can import most CAD data through a two-step process.

Lessons Learned

With the coming Standard for the Exchange of Product Model Data (STEP), it was hoped that CAD files would provide attributes (such as OBJECT_TYPE = "3/4 inch fastener") as well as geometry. But since STEP never seemed to gain widespread use and since many CAD systems did not store such information anyway, we never were able to make the process totally seamless. The user must still define attributes (such as fastener locations) within DEPTH. This increases the time required to simulate maintenance on new systems but is not a critical limitation in most circumstances.

CAD geometry is often much more detailed than needed by DEPTH. This can cause excessive memory allocation and degraded simulation performance. DEPTH does not directly assist in reducing this detail however some stand-alone utilities are provided that can reduce faces and nodes. When possible, the user should eliminate geometry that is unnecessary for the simulation.

The developers learned that not all files can be translated. Often the input data must be "massaged" before inserting into DEPTH. If Jack cannot completely interpret the input, it terminates ungracefully. Now that Jack has become a commercial software package, it is expected that problems like this will be resolved.

2.12 Workspace File

Objective

DEPTH simulations use a set of virtual humans, tools, fasteners, CAD objects, cables, and generic objects arranged in specific configurations. The user should have a convenient way to position, articulate and assign properties to figures. These configurations and properties should be saved in the DEPTH file.

Approach/Discussion

The DEPTH main menu provides access to editing functions such as modify, rename, cut, copy, paste, and delete. There is also an insert menu which makes figure creation easy and intuitive. Together, these features provide all the tools necessary to create an environment suitable for analysis. A file menu includes all of the commands necessary to store and retrieve the environment as well as export it for use in other applications. Other aspects of the workspace may be specified using the options menu.

The DEPTH workspace file stores all information about the environment. Information about all of the figures is kept in the Object Control Module (OCM) – an object class hierarchy from which humans, tools, fasteners, and all other objects are generated (Figure 12). Most of the commands used to manipulate figures in the process of building an environment are funneled through OCM. The simulation commands are discussed in another section.

OCM is a true object-oriented design that includes encapsulation, data hiding, inheritance, and polymorphism. A suite of list classes complete with interrelationships is used to keep the lists of figures, segment, sites, joints, and connections. Data persistence is achieved by employing read and write methods for each OCM class that chain a call to the parent's read or write method.

The DEPTH workspace file is actually an archive consisting of multiple files. These files include the environment information as written by Jack, the additional information as written by OCM, the report log file, and all of the the geometry (Psurf) files that have been inserted into the environment. The archive is packaged using the UNIX "tar" command.

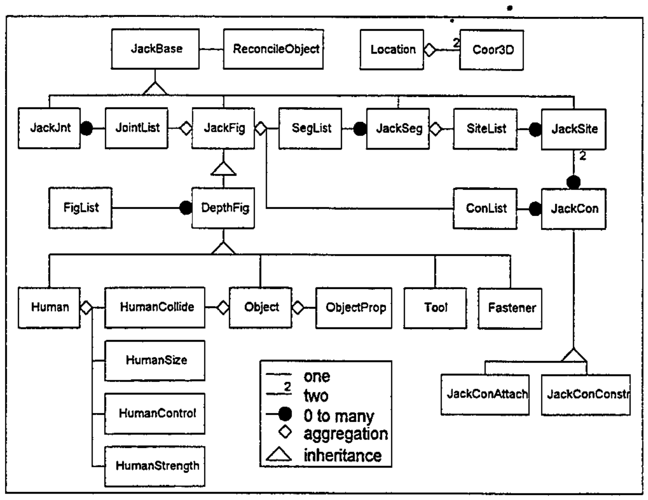


Figure 12: Object Control Module Class Diagram

Lessons Learned

The OCM was required because DEPTH needed to extend the information recorded by Jack. The figure, segment, site, and joint lists were duplicated in DEPTH to provided faster access. Merging the DEPTH and Jack processes would eliminate the need for this duplication of information. Also, this duplication requires that DEPTH obtain the unique ID of each figure, segment, site, and joint which is assigned by Jack at load time. Retrieval of this information slows load time for large files.

The OCM also covers some holes in the Jack file format including figure shading, figure display and human behaviors. Because of this, loading of DEPTH files is slowed because each segment must be shaded.

DEPTH files provide a convenient package of information. The files, however, can become quite large, especially if a significant amount of CAD data is inserted. To alleviate this problem, a file linking option was added, allowing CAD data to be shared between DEPTH files.

DEPTH creates a hidden directory in the user's home directory for unpacking files and storing user customizations. This can cause problems if they do not have enough disk space allocated. A better method would have been to extract information directly from the archive instead of having to unpack it. Jack, however, does not support packed geometry (Psurf) files.

During the testing of DEPTH 5.0 with Transom Jack 1.2, several compatibility problems were encountered. Specifically hand tools could no longer be loaded into DEPTH.[4] This made it necessary to fall back to Transom Jack 1.1 for the final delivery. An earlier delivery could have made it possible to work through the problem.

2.13 Task Analysis Assistance

Objective

Task Analysis Assistance consists of information made available in the DEPTH software. These capabilities would provide practical guidance and support to the design analyst. Guidance such as human factors checklists, prescriptive help or reference documentation is needed while conducting design reviews.

DEPTH version 3.0 supported four checklists:

- MIL-STD-1472D (Figure 13)
- Maintenance Skills
- Safety - Physical Involvement
- Safety - Recommendations

For DEPTH Phase V, the government decided to add the capability to create custom checklists.

Approach/Discussion

As a result of a study performed in Phase III, Hughes Training, Inc. recommended that MIL-STD-1472D and parts of the Human Factors Handbook [11] be provided in HyperText format. Unfortunately, legal and copyright issues prevented inclusion of the Human Factors Handbook and technical problems prevented the conversion of MIL-

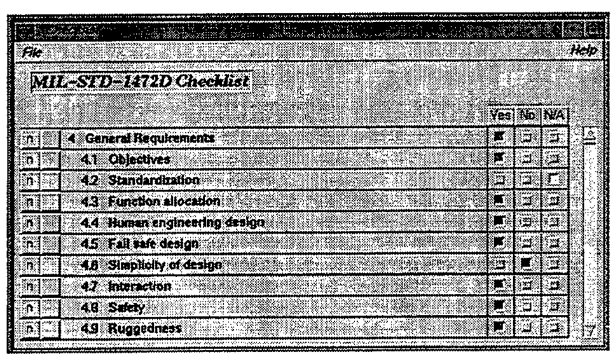


Figure 13: MIL-STD-1472D Checklist

[4] See Software CR number 11409 for a detailed description.

STD-1472D from a document containing Macintosh graphics to a document that could be displayed on an SGI workstation. The initial work to convert the MIL-STD-1472D on-line determined that a significant and costly effort would be required to provide the manual in a useful format. An alternative option that provided the MIL-STD-1472D checklist in a fashion similar, but expanded, to the hard copy generated by Lockheed was implemented.

Human factors assistance exists in the form of checklists. The user may check off criteria for the MIL-STD-1472D, Maintenance Skills, Safety: Physical Involvement, Safety: Recommendations, or any customized user added checklists. Access to on-line human factors documentation was not added due to budget constraints.

The code to create user checklists was written by the in-house team at Wright-Patterson and provided to Hughes as Government Furnished Equipment (GFE) for integration into DEPTH. The Checklist menu item originally cascaded into the four included checklists. This function was changed to open a dialog box that allowed the desired checklist to be selected from the list available. An *add* button on this dialog invokes an editor to create a new user-defined checklist. Similarly, buttons were added to edited and removed checklists.

Lessons Learned

There are a number of reasons why we found it unproductive to implement task analysis assistance to the extent originally envisioned. The process of converting paper-based documents to interactive electronic media can be quite involved. Copyright issues can be difficult to resolve and conversion issues between hardware platforms can make the task difficult. Further with the rapid increase of Worldwide Web resources, there is no longer a strong need to integrate such functionality into a simulation system. Only minor changes were necessary to integrate the user checklist GFE code.

2.14 Reporting

Objectives

The wealth of information provided by these simulations is most useful when consolidated in a report for review by the user. DEPTH should have a rich set of reporting functions complete with graphs and charts. The user should be able to easily obtain the information of interest without having to sift through pages of irrelevant data.

Approach/Discussion

It was recognized, however, that all analysis data should be logged in even without a full reporting mechanism. DEPTH currently logs changes to insertion and modification of figures, environmental factors and object properties, safety alerts, changes in constraint satisfaction, access violations, strength violations, reach failures, and simulation steps.

A "Reports Entry" dialog box (Figure 14) was written to allow the filtering of certain reportable events. Currently, reports can only be written to a text file and include no graphics.

Lessons Learned

The logging function records all the data returned by the simulation. However, additional work could be done to improve the format and readability of the report.

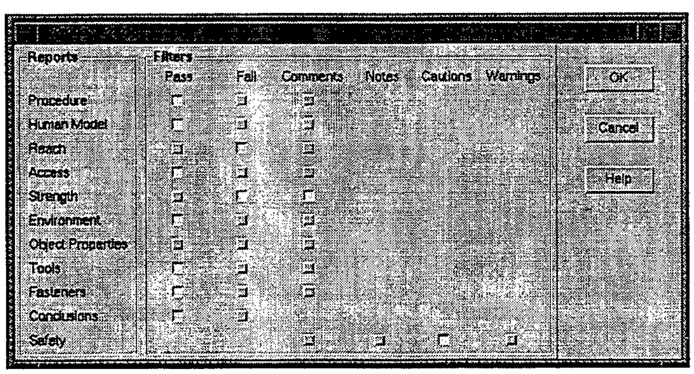


Figure 14: Report Filter Dialog

2.15 Demonstration

Objective

DEPTH should be tested with a real-world design scenario to demonstrate improved design evaluation for Air Force weapon systems.

Approach/Discussion

DEPTH provides a rich set of functions that allow maintenance tasks to be simulated in a cost-effective and speedy fashion. The improved capabilities for rapid development of simulation and animation were provided in a demonstration that used actual 3-D CAD models.

Several maintenance scenarios were considered for the demonstration. Since the goal was to find a weapon system that was under design using CAD, the F-22 aircraft seemed to be an ideal candidate. However, the CAD data were not readily available to any contractor on the DEPTH team, and even if they were, many drawings were classified.

In place of the F-22, Hughes recommended to use F-15 radar that was being upgraded.[5] Again, CAD data was not easily obtained by the DEPTH team, so pictures of the existing configuration and upgrade descriptions were used to create the virtual mockup with DEPTH. The B4 Multipurpose Aircraft Maintenance Stand used in this task was only available in 2-D CAD drawings and conversion to 3-D was nontrivial. The stand was modeled using the Pro-Engineer CAD package. Finally, the F-15 exterior was obtained from Viewpoint Data Labs.

The following maintenance activities were selected to demonstrate DEPTH's aircraft maintenance task simulation capability:

- A maintenance technician opened the radome. A small stature human was initially used to demonstrate an unsuccessful reach. As expected, the simulation reported that the technician failed to reach the radome. After a human of average stature was substituted, the simulation showed that the task could be successfully accomplished.
- Next, the radome was secured with the lockout bar. This illustrated object handling from the technician's viewpoint. The original plan was to perform the visual search for the lockout bar and show that the view was obstructed. This was not possible because of a code bug. Since the collision list was used for both visibility and collision checking, the adjust joint function reported erroneous collisions. However, the proper placement of the lockout bar was demonstrated.
- A visual inspection of the radar antenna was conducted next. Again, we simulated this through the technician's eyes.
- The next task was to disconnect the Analog Signal Converter. This activity was intended to demonstrate collision detection when oversized connectors were substituted in the environment. However, the collision detection could not be performed because user-defined motions could not access the collision list. Additional code to provide this capability was not possible given time constraints. However, disconnecting the Analog Signal Converter enabled the removal of cables and fasteners.
- The Analog Signal Converter was removed and replaced from the aircraft bay. This demonstrated the ability of one person to pull and lower an object. Since the mass of the LRU was intentionally chosen to exceed the weight limits for one person, the ability to detect and present a strength warning was demonstrated. After the weight (mass) was readjusted, the simulation was rerun. The replace maintenance task demonstrated the lift and push motions and the reconnection of cables and fasteners. Similarly, the transmitter disconnection, removal and replacement demonstrated a two man push, pull, and lift.

---

[5] Specifically this was the APG-63 (V)1 upgrade on the F-15 C/D. A similar upgrade had already been performed on the F-15 E.

The following motion models were used to create the simulation maintenance tasks included in the demonstration:

- Walk
- Look At
- Use Tool
  - Grasp and Loosen
  - Release
- Acquire
  - Grab
  - Release
- Transport
  - Lift/Lower
  - Push/Pull
- Object Move
- Adjust Object Joint

Additionally, a number of custom PaT-Nets were created specially for the demonstration:

- Detach and Attach Converter Cables
- Detach Transmitter Cables
- Detach Transmitter Wave Guides
- Change Floor Level
- Change Human Location (2)

The last item created for the demonstration was a narrated video to show DEPTH's rapid development environment. The video covered how to import, size, and place LRUs; how to create, position, and attach cables; and how to execute a motion model.

Lessons Learned

The F-15 demonstration applied the DEPTH software in a real-world design scenario and showed its utility. The most significant technical challenge was the CAD data itself. CAD data is often difficult to obtain, especially for older systems. Since the CAD data provided more geometric detail than was actually required, the models were difficult to manipulate and computer response times were slow.

We also had difficulty obtaining valid, quantitative cost/benefit data due to the lack of comparison baseline data from other maintainability analysis methods. However, we were able to show notional improvements from working with soft mockups versus physical mockups, and from working with simulated humans versus real humans.

2.16 On-line Help

Objectives

On-line help – descriptions of the functionality, menu structure and dialog boxes – should be available and should provide background technical information when appropriate.

Approach/Discussion

On-line help written in HTML has been provided for all non-informational dialog boxes. The help pages that are displayed with Netscape contain links to related help pages. The pages were structured to mimic the workspace and simulation menu structures.

Lessons Learned

Use of Netscape for display of help files has some caveats. It takes time to load the Netscape application and display the help file. Also, if Netscape is already running, an error message appears that must be confirmed before the help page is displayed.

Standard Graphical Markup Language (SGML) was studied as an alternative to HTML for the markup language for the DEPTH help files. HTML was selected because the tools to create and display the text were widely available on numerous hardware platforms. Most were available at no additional cost to the contractor or the government. Also, we hoped to post the help files on the DEPTH Web site. The SGML tools were available in much smaller numbers and were expensive.

2.17 Error Logging

Objective

A software package should have a centralized mechanism to log and report internal software errors. This mechanism should handle errors caused by the user, operating system, or the hardware. The error message should indicate how to resolve the problem. Information should also be logged to assist developers in analyzing and correcting more serious problems.

Approach/Discussion

All non-recoverable errors due to invalid parameters or an unexpected system state are handled through a single "postError" function. This function looks up the error code in an error description file and then presents an appropriate error message. The calling function provides the module name and optional debugging text. This information is logged to the file "DEPTH_Error.log" in the user's home directory. Depending upon the severity of the error, the user may either exit or attempt to continue.

Lessons Learned

Providing error logging and reporting with a common function makes it easier to handle new errors. It also makes the task of writing (and translating) user documentation easier, since the text for all errors can be found in one place in the development library.

3. Recommendations For Further Research

Although a great deal was accomplished in the DEPTH program, more work can be done to meet more of our user's needs. These recommendations are listed below in several categories. Functionality development constrained by technical problems are noted with an asterick (*). The others were simply advances that could not be developed due to time constraints.

Hazard Analysis

- Hazard analysis is an important area with a significant amount of R&D still needed. The hazard study [8] indicated that this could be implemented where object attributes (noise, temperature or radiation) are user-defined.
- Proximity effects can be represented as semi-transparent spheres. Once the human contacts this sphere, the user can be notified in a variety of ways (sound, dialog box, icon changes and/or color changes).
- Users should be able to define their own hazards. Examples of user-defined hazards include jet engine intakes and heavy objects hanging from a crane. As with predefined hazards, the ranges of user-defined hazards can be represented as a semitransparent field with the size and shape defined by the user. The set of shapes should include spheres, cylinders, boxes, cones, and scaled objects (larger versions of the hazardous object). The warning notification should also be user defined.

- Simulation of shielding or cumulative radiant effects would be quite difficult to implement.
- Implementing lighting analysis as specified in the hazard report [8] would also be quite difficult. This would require calculating the distances from user-defined light sources to the work location. Additionally, it may be beneficial to consider shading and accumulative lighting effects.
- Electrical current and voltage should be combined for simplicity.

Cognition

- Interfacing DEPTH with a cognitive modeling system would be useful to create simulations automatically and simulate cognitive factors. This "marriage" between simulations of mind and body would be useful in the human simulation industry as a whole.
- A standardized interface could allow any cognitive model to add a body and any body model to add a brain. The Society of Automotive Engineering (SAE) Human Modeling Committee is developing such a standard.

Reporting

- Statistical analysis of simulation results.
- Ability to add comments.
- Conclusions about torso, pelvis, foot, and arm articulation.
- Improved format with various fonts, color and graphics.

Simulation

- Effects of gravity and other forces. Without gravity, objects can seem to suspend in mid-air.
- The simulation software needs to "understand" floors, walls, ceilings and other barriers. Also, the human should not be able to reach through their body.
- Only render detailed objects when they are relevant to the simulation.
- Support of multiple processors to improve real time graphics performance.
- Efficient collision detection.

Human Motions

- Heel articulation.
- Provide a crawl motion with a target location.
- Simulate visual inspections with user-defined target and duration.
- User-defined handles and grasps.
- Walking backwards or sideways while carrying a load.*
- Cooperative carrying.
- Avoid obstacles (including other humans) during walk and crawl motions.*
- Evaluate tool sweep with hand attached.*
- Reach planning for touch, grab, and grasp motions to detect collisions and to assist in clearance analysis.*
- Allow concurrent (simultaneous) motions.*
- Enhanced user supplied simulation capabilities
- Improved joint movement, for example, the human should not reach through their body. Also, joint interdependencies should be recognized, since some joints cannot bend as much when adjacent joints are also bent.
- The motion simulation capability could be improved by changing the way Jack manipulates behavior settings. Currently, Jack modifies behaviors without notifying DEPTH, and further, the initial settings are not restored. Many behavior settings are undocumented.
- Concurrent (parallel) motion models are not supported but are needed.

Anthropometry and Strength

- Provide clothing templates for females.
- Associate strength with anthropometric populations.

- Include strength calculation during combination motions.
- Simulate performance degradation due to exposure and fatigue.*

Logistics Database

- Database input to associate support equipment with the unit under test.
- Do not require a complete database so the update capability can be demonstrated.

Safety Warnings Based on MIL-STD-1472D (paragraph numbers listed)

- Environment temperature – para. 5.8.1.8.
- Object temperature – para. 5.13.2.1.
- Lighting – para. 5.8.2, Table XXI.
- Noise – para. 4.2, 4.3 and MIL-STD-1472C: 5.1.1.1.
- Hazardous material* – para. 5.13.2.1, 5.13.7.4.1, 5.13.7.4.2.
- Radiation* – para. 5.13.2.1, 5.13.7.5; MIL-HDBK-764, para. 10-8.1; and AFSC, Design Note 3D1 2.2.1.

Other

- Site resolution. This is similar to figure resolution where objects are selected in a logical order (front to back using ray tracing).
- An icon bar for simulation mode.
- Hypertext style access to reference material relating to human factors, design and maintenance.
- Allow the user to perform more rapid analysis by stripping functionality – even the graphical visualization.
- Gracefully halt movie capture if disk space runs out.*

Some of these issues are common among 3-D simulations and are being addressed by SGI and others. However, most are unique to human activity simulation and maintenance evaluation.

4. Conclusions

The DEPTH program had many successes. Most importantly, we demonstrated how maintenance simulation could be used for design evaluation without costly physical mockups. We spent considerable effort to find an optimal way to make these simulations easier to use. Three-dimensional simulation of human activity is extremely complex, but we developed software that made it relatively easy. A number of other project objectives were successfully achieved including:

- Detailed simulation of the human hand.
- A practically seamless interface between Jack and DEPTH. Despite being two computer processes, they act as one integrated product.
- A user-friendly interface that flows naturally from task to task.
- Geometry translation from most popular CAD systems.
- Anthropometry can be defined from population data or specific landmark measurements.
- An enhanced tool set originating from the Crew Chief program.
- Automatic tool selection based on fastener information.
- A wide variety of fasteners, cables, hoses and wave guides
- A library of motion primitives to construct simulations.
- Automatic task definition via menus and dialog boxes. Task steps are displayed in a list.
- Image and "movie" capture for multimedia applications.

- Strength warnings displayed as appropriate.
- Driving human model motion with a noun-verb syntax.
- Automatic update of a logistics database from simulation results.

References

[1] Air Force Contract F33615-91-C-001, *Design Evaluation for Personnel Training and Human Factors (DEPTH)*, Statement of Work, WPAFB, 1991

[2] Badler, N. I.; Barsky, B. A.; and Zeltzer, D., *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*, Oxford University Press, 1991

[3] Badler, N. I.; Phillips, C.; and Webber, B., *Simulating Humans: Computer Graphics, Animation, and Control*, Oxford University Press, 1993

[4] Badler, N. I.; Douville, B. J.; and Ting, B., *Center For Human Modeling and Simulation Quarterly Progress Report No. 54*, University of Pennsylvania, 1995

[5] Barron, M. K., "Recommended DEPTH Enhancements - Cable Generator and Connector Analysis," Lockheed Martin Tactical Aircraft Systems, 6 September 1995

[6] Douville, B. J., *PaT-Net User's Guide*, University of Pennsylvania, 1995

[7] Ianni, J., "Crew Chief: Present and Future", *Human Centered Technology for Maintainability: Workshop Proceedings*, 1991

[8] Ianni, et al. "Maintenance Hazard Simulation: A Study of Contributing Factors," *Human Interaction with Complex Systems Conference Proceedings*, 1996

[9] Military Standard 1472D, *Human Engineering Design Criteria for Military Systems, Equipment and Facilities*, 1994

[10] Ousterhout John K., *Tcl and the Tk Toolkit*, Addison-Wesley Publishing Company, Massachusetts, 1994

[11] Woodson, Wesley E.; Tillman, Barry; and Tillman, Peggy L., *Human Factors Design Handbook, Second Edition*, McGraw-Hill, Inc., New York, 1992

[12] Wymore, A. Wayne, *Systems Engineering Methodology for Interdisciplinary Teams*, John Wiley & Sons, Inc., New York, 1976

List of Symbols, Abbreviations, and Acronyms

3-D .... Three Dimensional

AL/HRGA....... Armstrong Laboratory, Logistics Research Division – now the Air Force Research Laboratory, Sustainment Logistics Branch (AFRL/HESS)

API .... Application Program Interface

CAD .... Computer Aided Design

CAGE .... Contractor and Government Entity

DED .... Data Element Definition

DEPTH .... Design Evaluation for Personnel, Training, and Human Factors

DRaW .... DRaW Computing Associates, Inc.

EAGLE .... Enhanced Automated Graphical Logistics Environment

GFE .... Government Furnished Equipment

GUI .... Graphical User Interface

HMSC .... Hughes Missile Systems Company

HOS .... Human Operator Simulator

HTML .... HyperText Markup Language

IPT .... Integrated Product Team

Jack .... Transom Jack Human Modeling Software

LRU .... Line Replaceable Unit

LSA .... Logistics Support Analysis

MIL-STD .... Military Standard

MTM .... Motion Time Measurement

MPEG .... Moving Picture Expert Group

OCM .... Object Control Module

PaT-Net .... Parallel Transition Network

PC .... Personal Computer

Penn .... The University of Pennsylvania

RPC .... Remote Procedure Call

SGI .... Silicon Graphics, Incorporated

SGML .... Standard Graphical Markup Language

SOW .... Statement of Work

SQL .... Structured Query Language

Tcl/Tk .... Tool Command Language/Tool Kit

RGB .... Red, Green, Blue

Unix .... A common operating system used on workstations