Requirements for a High Fidelity Virtual Aircraft Maintenance Training Environment

Glen A. Pfeifer
The Boeing Company
St. Louis, Missouri
glen.a.pfeifer@boeing.com

ABSTRACT

Virtual reality (VR) training applications have attracted worldwide attention and study. One application that shows great potential is aircraft maintenance training. Virtual aircraft maintenance trainers offer the advantage of affordability and flexibility over traditional hardware-based trainers, providing an accurate, 3D representation of the aircraft without the associated lifecycle costs of physical hardware mockups and original aircraft equipment. Moreover, VR permits comprehensive user interaction with the virtual aircraft's systems and components. These range from dynamic navigation using augmented or hypothetical viewpoints to complex manipulation of realistic virtual aircraft components.

The past several years have seen advances in computer and visualization technologies that make the integration of VR with traditional training media more practical. To exploit this new potential, The Boeing Company's Phantom Works group has been investigating high fidelity virtual environments for maintenance training as a more effective means to support its aerospace products. One result of this effort is a matured set of requirements, derived in part from lessons learned creating numerous prototypes. This paper will discuss these requirements, which address many key elements necessary to develop an effective virtual maintenance trainer. Discussion will recognize the challenging issues of source data conversion, geometry management, user navigation and interaction, and functional simulation integration, all associated with the virtual environment. Training systems providers who develop solutions based on these blueprint items will better produce more realistic and effective environments for maintenance training.

ABOUT THE AUTHOR

Glen Pfeifer is a Training Technology Engineer with The Boeing Company's Integrated Defense Systems in St. Louis, Missouri. Mr. Pfeifer joined the Design for Supportability group in January 2000, bringing more than ten years of experience in maintenance trainer design and development. Mr. Pfeifer has been investigating, developing, and demonstrating 3D virtual environments to support Boeing's search for aircraft maintenance training solutions.

Requirements for a High Fidelity Virtual Aircraft Maintenance Training Environment

Glen A. Pfeifer
The Boeing Company
St. Louis, Missouri
glen.a.pfeifer@boeing.com

BACKGROUND

Maintenance training for aircraft programs has traditionally been accomplished using a disciplined mix of training media, including life-sized hardware trainers. Hardware trainers have evolved into high fidelity, computer controlled functional mockups of the aircraft. They provide an effective platform on which to learn not only practical, hands-on motor skills, like those necessary to remove and replace an aircraft component, but also conceptual skills, such as those used to troubleshoot an aircraft system.

Hardware trainers, however, possess a number of shortcomings that make them less than an ideal training media. Their increasing complexity, pricey hardware content, and ultra-low production rates make these devices expensive to design, build, maintain, operate, and upgrade. They require a large facility's footprint dedicated to their operation. Hardware trainers also tend to be a bottleneck to large student throughput, because of their restricted accessibility, and the extra care and safety that must be exercised when using functioning hardware. Safety issues often place even more demand on Instructors' time because an additional safety observer is required. Moreover, these devices are ‘fixed’ assets, requiring the additional costs of travel and temporary relocation for personnel to acquire the necessary training. Over a twenty-year trainer lifetime, the total of these life cycle costs can be enormous.

An alternative training media, virtual reality, is now being explored that could alleviate these many issues. Advances in computer and software capability over the past decade have made it possible for digital 3D environments to eliminate the need for (or greatly reduce the life cycle costs of) many hardware training devices. Computer generated 3D virtual environments permit the visualization of and interaction with an aircraft and its systems without the need for actual physical hardware.

To exploit this new potential, The Boeing Company has been investigating high fidelity 3D virtual environments for maintenance training as a more cost effective means to support its aerospace products. Virtual environment technologies and processes are being developed with the goal of employing them as an effective and economical training media.

Virtual Reality and Virtual Environments

Virtual reality (VR) is commonly thought of as an ‘immersive’ experience. The sense of presence and level of interactivity provided by the computer simulated environment convinces the user that he is fully part of the environment. The human-simulation interface is accomplished with special equipment, such as a head-mounted display device for visualization, and cyber-gloves or a 3D mouse for interactivity. Immersive VR can be compelling under select circumstances, but lag time and low resolution in response feedback continues to be an issue for complex environments. An internal study revealed that among several forms of computer-based 3D training tested, immersive VR scored consistently low, in large part because it did not yet provide the responsiveness required to make it suitable for use as a maintenance trainer (Barnett, Helbing, Hancock, Heininger, & Perrin, 2000).

However, the same study revealed that complex 3D virtual environments can be experienced effectively by way of an ordinary display monitor, and a computer mouse and keyboard. In this ‘desktop VR’ the user's sense of immersion is no longer compelling, but when compared to traditional CBT formats, its ability to enable comprehensive visualization and interaction with complex 3D digital mockups is exceptional. These characteristics bolster the significance of desktop 3D virtual environments as a new and powerful media available to the contemporary Instructional Designer.

High Fidelity Virtual Maintenance Trainer

A subtle but important difference exists between the use of 3D graphics in computer-based training and the concept of a virtual maintenance trainer (VMT). The VMT can be thought of as a virtual 3D aircraft mockup – a digital training ‘device’ made to look and act like the real aircraft. In general, the VMT is not intended to be a training module, authored to present a specific lock-step sequence of actions for a given task or procedure. In the author's view, the VMT is at its best when designed as a total 'free-play' environment. Though a VMT could be created to accommodate only a specific training task, or family of similar tasks, the vision is a complete 3D digital mockup of the aircraft that can accommodate any training procedure.

The Boeing Company has recognized the potential for such a training media, and is investigating how it might be applied to practical use. One result of this effort is a set of requirements believed to be necessary to produce an effective, high fidelity VMT. Derived in part from lessons learned creating numerous prototypes, these requirements address a number of key VMT elements vital to achieving its full potential as a training device.

Specifically, the VMT should:

- Provide the user access to part attributes of the original source data.
- Exploit the original source data assembly structure.
- Enable the user to visualize very large model datasets.
- Exploit as many dynamic model management techniques as practical.
- Exploit the use of linked, or non-embedded, content data.
- Enable the user to efficiently navigate his viewpoint within the 3D environment.
- Enable the user to manipulate individual objects within the virtual environment.
- Enable multiple users to interact within the virtual environment simultaneously.
- Exploit the use of event-driven behaviors.
- Integrate the interactive visual simulation with an external functional simulation.

Each of these ten core requirements will be addressed in the paragraphs that follow. Discussion of these requirements will recognize the challenging issues of source data conversion, geometry management, user navigation and interaction, and functional simulation integration, as they apply to the development of an effective VMT.

SOURCE DATA CONVERSION

On a modern aircraft program, one of the greatest advantages to the VMT developer is the wealth of digital 3D aircraft modeling data already available. The design phase of an aircraft program yields a huge database of design models that define the physical form, fit and function of the parts and systems on the vehicle. These design models provide a rich source of digital 3D data that can be reused to efficiently and accurately populate the virtual maintenance environment. But creating the visual content for a virtual maintenance environment is dependent on the ability to convert the source data into a form useable by the visualization application.

From the perspective of a computer's CPU, engineering CAD models are very 'heavy' to process, making it impractical to dynamically and interactively view large portions of the digital aircraft at once. The 3D objects in virtual environments, on the other hand, employ much lighter tessellated representations of the original CAD models. A tessellated representation of each object is created by a conversion process that extracts only the mathematical definition of the model's surfaces, and then simplifies it into a 3D mesh of triangles that approximate the shape of each surface. The resulting 3D object is a lightweight depiction of the engineering math model. When rendering tessellated representations, the computer is now only burdened with computing the vertex coordinates and vectors of each triangle to provide a rendered image of the object on the screen. To be certain, there are many inexpensive software and hardware optimizations for rendering triangles so that objects made of triangles are 'computationally friendly' and fast.

Today the conversion of engineering CAD model data into tessellated geometry for visualization is a relatively routine process. However, the extraction of non-geometric data is more ambiguous. Engineering design models can yield an abundance of non-geometric data during conversion – recycled data that can provide great benefit to both the VMT developer and end-user. In particular, when converting source data to build a VMT, the ability to harvest part attribute and assembly structure information is of key interest.

Part Attributes

The VMT should provide the user access to part attributes of the original source data.

Most 3D modeling applications allow the aircraft designer to designate data attributes for each part model. These attributes, or metadata, can include information as common as part name and part number, or as obscure as logistics control number and part serial number. The value of making reused part attribute information available on the VMT is clear. It provides the end-user useful and specific aircraft information at little or no additional cost.

During development of the VMT prototypes, one of the most useful attributes recycled during the conversion process was the part name. The conversion script assigned the name attribute of each CAD part model to the name of each resulting virtual geometry file. When loaded into the virtual environment, these unique part names provided the developer an easier means to distinguish individual parts in a part tree that might be thousands of parts large. This example illustrates how engineering metadata, recycled for VMT use, can be a useful and valuable asset.

Assembly Structure

The VMT should exploit the original source data assembly structure.

Organization of large assemblies of parts is often accomplished in 3D applications by defining them in a hierarchical arrangement, that is, components combine to form sub-assemblies, which combine to form assemblies (see Figure 1).

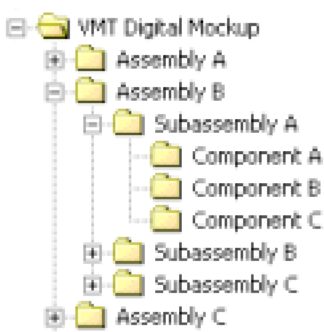

Figure 1. Sample Hierarchical Assembly Structure

Many 3D design applications utilize an assembly structure. Preserving that assembly structure during file conversion is important because of the considerable time saved in organizing the objects that make up the virtual geometry content. Also, assembly structure is important because visualization applications often associate a part's characteristics, like 3D position or color, with respect to its parent component. In many of the VMT prototypes, it was easy to open an entire door assembly, comprising many individual parts, by rotating their single common parent about a hinge line.

GEOMETRY MANAGEMENT

A typical aircraft can easily contain 100,000 individual parts or more, each defined as a 3D digital model rigged in a vast product data manager. When visualizing the virtual aircraft environment on the computer display, the presentation of so many parts offers tremendous assembly fidelity and spatial realism. But to smoothly render a very large set of objects places a tremendous burden on a PC's CPU and graphics card. As a user dynamically navigates through the 3D environment, the positions of all the objects change with respect to the viewer's perspective, forcing the CPU to calculate new positions so that a new image, or frame, can be rendered. As sequential frames are calculated and displayed, the illusion of motion is generated. For smooth, natural motion within a virtual environment, a frame rate of 30 frames per second or better is desired.

Frame rate is dependent on the number of objects being rendered, and the speed and memory capability of the computer equipment being used. Frame rate is important because if it is too slow, scene dynamics, such as viewpoint changes, become jerky and more difficult for the user to control or interpret. The lag between user input and visual response becomes a considerable distraction and inhibits training value. Empirical results suggest that between 15-20 frames per second are considered minimal to achieve satisfactory results.

Adequate frame rate, faster load times, and stable object management are but a few objectives when processing complex VMT environments. Fortunately, current software applications used for 3D visualization apply a number of techniques that are useful in managing very large assemblies of 3D objects. Three common techniques are very large model management, dynamic model management, and the use of non-embedded geometry.

Very Large Model Management

The VMT should enable the user to visualize very large model datasets.

Significant computer horsepower and a high-end 3D visualization application are necessary to accomplish this. Only within the past few years have standard PCs and their graphics cards been capable of rendering very large assemblies with satisfactory frame rates.

One of the author's earliest VMT prototype efforts was designed to familiarize the user with a complete nose to tail layout of an aircraft fuel system (see Figure 2).

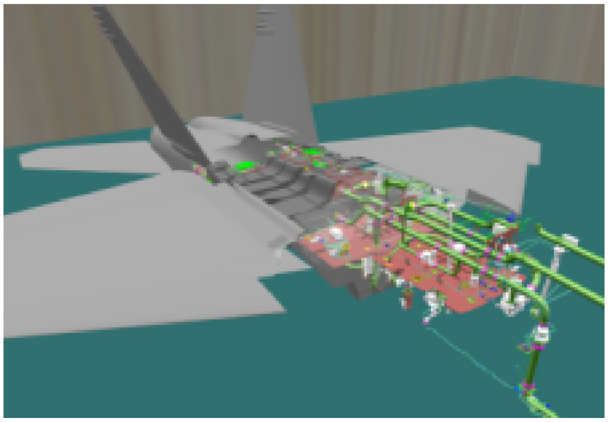

Figure 2. Virtual Fuel System

Using this prototype, the end-user is able to fully visualize not only the many fuel subsystems and how they interconnect, but also how they are positioned in the aircraft's moldline. The prototype environment reveals over 3300 fuel system components, and can be effectively demonstrated on a commercially available high-end laptop PC.

To further illustrate computer advances in visualizing very large model datasets, recent internal trials have been completed that demonstrated an assembly of over 100,000 objects (represented by 21 million triangles) on a standard Dell 650 technical PC. Interaction with the environment was unacceptable at a refresh rate of less than one frame per second, but it is clear that high-end desktop PCs are showing vastly improved capability compared to even a couple of years ago.

Dynamic Model Management

The VMT should exploit as many dynamic model management techniques as practical.

The purpose of dynamic model management is to actively maintain a tolerable frame rate while interacting with the VMT. In practice, this is accomplished by reducing the workload on the CPU by controlling the total number of objects (i.e. triangles) being rendered on the computer screen at any one time. Several powerful techniques are commonly used by rendering applications, including level-of-detail management, dynamic load management, and occlusion culling.

When level-of-detail (LOD) management is applied, the total triangle count of each object being rendered in a virtual scene is made dependent on the distance of that object from the user's eye point. Objects farther away are rendered at a lower LOD. Consequently, as the user moves farther away from objects in a scene (and more objects come into view on the screen), the far objects are automatically transformed into or replaced with representations composed of fewer triangles (having less detail). This is reasonable because in a 3D perspective view, objects farther away are smaller and cannot be viewed in full detail anyway.

PTC's dvMockUp, a powerful VR application used during the author's investigations, assigns three levels of detail to every object during the conversion process. The developer is allowed to manually set the viewing range for each level, say, best detail within one meter of the object, moderate detail between one and ten meters, and lowest detail beyond ten meters. This technique became especially helpful as we transitioned our prototype demonstrations from our desktop PC to our somewhat less capable laptop PC. The active LOD transitions that occurred while navigating through the environment helped maintain an acceptably smooth sense of motion.

Dynamic load management is another means used by some visualization applications that can improve the frame rate of large, high fidelity VMTs. In this technique, the virtual environment monitors the context of the end-user's actions, and actively loads and unloads geometry data based on those actions. Most typically the load status is dependent upon the end-user's vantage point. For example, an aircraft virtual environment can be authored so that the system components behind an access door are loaded for viewing only as a result of a specific user action, such as opening the door by selecting its latch, or navigating into a pre-defined bounding box in which the components are visible. Since frame rate is affected by the number of visible triangles, limiting the environment to only the objects that can be seen (thus minimizing the total number of objects loaded at any one time) is a very effective way to improve frame rate.

Finally, a technique called occlusion culling is also employed in rendering applications. Occlusion culling is the computational process of identifying surfaces that are hidden from view and eliminating them from the rendering queue. Though more CPU intensive, this technique can significantly reduce the total number of triangles that are rendered in every frame.

Non-Embedded Geometry

The VMT should exploit the use of linked, or non-embedded, content data.

There are two ways in which a VR application file can define the visual (and other) elements of a virtual environment. In an embedded-type application, the data elements such as 3D geometry, texture images, and audio files are all integrated within a single definition file. It is not uncommon to find 3D visualization applications that use an 'embedded' file format. Many proprietary 3D environments such as Shockwave3D embed all the objects in the scene into one file which, for complex, high fidelity virtual environments, yields a runtime file that is very large and difficult to manage. Likewise, scenes created in uncompressed Virtual Reality Markup Language (VRML) format can result in extremely large files if not created correctly.

As an alternative, high-fidelity 3D applications can employ a non-embedded format. The non-embedded format features a relatively lightweight 'scene' file that specifies links to individual geometry (and other) data files. When launched, the scene file manages the creation of the virtual environment by loading only the content data files that it defines.

Although the number of files necessary to generate a single virtual environment balloons from one file to potentially thousands of files (one for each unique aircraft part), each individual file is relatively small, and easily reusable by different VMT scene files. Moreover, many instances of a single part, such as a screw, can be represented in a scene at multiple locations by using a single master geometry file. Envision a training program that might require dozens of different virtual environments, one for each different training need (or perhaps even one for each lesson), and the advantage of each VMT scene reusing the same geometry files becomes clear.

In addition to more efficient file management, the non-embedded format offers other key advantages. One is improved VMT configuration control. Upon creating a master VMT geometry database, any follow-on aircraft part or assembly configuration changes could be individually processed and added to the database, allowing many configurations to be represented within a single data store. Given a well-conceived filename syntax and the ability for the Instructor to specify the necessary parameters when loading a VMT scene file, it's reasonable to envision the ability to tailor the training environment to a specific aircraft tail number.

Another advantage of implementing a non-embedded format is improved trainer-to-aircraft concurrency. More timely updates could be made to the VMT(s) by forwarding updated geometry data files and scene files via the Internet to the end-user's database. Only the changed files would need to be replaced.

USER NAVIGATION AND INTERACTION

One of the distinguishing features of a 3D virtual environment is its high degree of user interactivity. The interactive nature of the VMT is important to its potential as a learning tool. Seeing graphical depictions of objects is one thing, but exploring and manipulating those object in 3D is quite another. This control offers the end-user a sense of engagement with the training material, affecting a keener interest and improved understanding of the subject, thus making the training more effective.

In essence, the user can engage the VMT in two ways: by controlling where he is and which way he looks, and by interacting with the objects encountered. Additional realism is experienced when multiple users are allowed to interact freely within the same environment.

Scene Navigation

The VMT should enable the user to efficiently navigate his viewpoint within the 3D environment.

As a minimum, the end-user must be given the ability to navigate throughout the 3D scene with a full six degrees of freedom. When viewing the VMT on a computer screen, six-DOF navigation can be accomplished a number of different ways. User-input can vary from simple mouse-clicks on a graphical user interface, to more complex manipulations of common PC interface devices.

No matter what the means, however, the interface must be intuitively easy to learn and use. Otherwise the intended learning experience is muted because the user's focus is on the equipment, not the training. One particular device, the 3D Spaceball®, may be an optimal solution. This device provides an intuitive and natural user interface -- push the ball forward, the viewpoint moves forward; twist the ball clockwise, and the viewpoint rotates right, and so on. Navigation executed by the 3D Spaceball is independent of the mouse and its cursor, leaving the mouse dedicated to other more basic tasks, such as selecting 3D objects, or picking menu options. Spaceballs are also relatively inexpensive, and most 3D applications have drivers available to operate them.

Finally, in addition to being able to manually navigate in the 3D environment, the ability to 'auto-navigate' is often more valuable. Training time can be made more efficient with the use of selectable menu options or icons that activate pre-defined viewpoints. These views of interest can be a discrete point (like the pilot's viewpoint in the crew station), or a view that is automatically transported along a specified path.

Object Manipulation

The VMT should enable the user to manipulate individual objects within the virtual environment.

The ability to select a component of interest and then move or rotate it independent of its parent assembly is an important element to the utility of the virtual trainer. Object manipulation provides an easy means to inspect and better understand aspects of a component that might normally be hidden from view. This capability is an enhancement over physical hardware trainers.

Multi-User Collaboration

The VMT should enable multiple users to interact within the virtual environment simultaneously.

This capability is important because it opens the Instructor's options to team-oriented training tasks that more closely resemble real life maintenance activities. The procedures necessary to perform typical aircraft system troubleshooting tasks, for example, are accomplished more efficiently with two or more maintainers positioned at different parts of the aircraft and communicating over a headset.

In general, true collaborative capability allows multiple users to exist simultaneously in the same virtual environment. Each user's interaction and vantage point is independent of the others. If desired, the presence of each participant can be represented as a virtual body. However, since digital manikins are generally triangle-heavy, in practical application it is more efficient to display only part of a body, like a head or a hand.

In a desktop virtual environment each user is able to see the system responses affected by the other users' interactions. An example of collaborative training in a virtual aircraft maintenance environment might have one user controlling crew station switches to apply hydraulic power and another user checking the resulting hydraulic fluid quantity on a gauge in the landing gear bay.

Collaborative training in a virtual environment is more difficult to achieve but well within the capability of modern PC equipment. The VR application must be specifically designed to handle multiple scene channels, each tied to an individual user interface, or the individual VR applications need to be able to communicate and exchange information.

FUNCTIONAL SIMULATION INTEGRATION

The virtual maintenance environment provides the end-user the means to visualize and interact with a digital 3D mockup of the aircraft. Previously, discussion has focused on requirements associated with the visual content of the VMT. But equally important to the utility of a VMT is its ability to realistically simulate the aircraft's functional behavior. However convincing the virtual aircraft may look, its true value as a training tool will be in how it acts.

Event Driven Behaviors

The VMT should exploit the use of event-driven behaviors.

Event-driven behaviors, including those driven by conditional parameters, are programmed instructions that effect changes in the virtual environment. They are triggered by predefined events, like user interactions, or the execution of other behaviors. The VMT developer can apply behaviors to specific aircraft parts or assemblies, or to the virtual environment in general.

In previous prototyping efforts, dvMockUp, which incorporates dozens of ready-made behaviors, has been used with effective results. Some newer applications now exist that allow the developer to edit existing behaviors, or create new ones from scratch, to meet unique functional capabilities. A typical VMT library of behaviors must include, as a minimum, several associated with changing the position or orientation or motion of an object. Others will be needed to affect changes in the appearance of an object, like color or transparency or texture. In addition, behaviors will be needed to control the lighting or audio characteristics of the environment.

A recent VMT prototype, designed to illustrate how a jet level sensor in an aircraft fuel system works, demonstrated a number of integrated behaviors. The end-user can initiate fuel system operation, and then can interactively select various components near the jet level sensor to better visualize and understand its theory of operation. Behaviors are effectively used to see inside fuel lines and other components, to see flow direction by animating ‘flow balls’ through the fuel lines, and to see other physical changes in the system as fuel is consumed (see Figure 3).

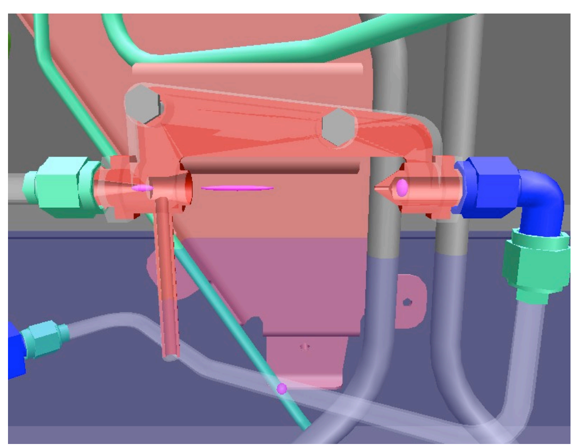

Figure 3. Functioning Jet Level Sensor in a Fuel Tank

Another example of how behaviors have been used in several VMT prototypes is the application of conditional motion behaviors to an aircraft access door and its latch. Selecting the closed door with the mouse cursor (event) results in the door hinging open, but only if the latch has already been selected (unlatched). Selecting the open door then results in the door hinging closed. Another combination of behaviors commonly used in several prototypes influence changes in motion and appearance – selecting a toggle switch on a control panel (event) causes the 3D switch to snap to the ON position, which in turn causes an indicator light to change color.

When implementing conditional logic for behaviors like the examples above, local or global variables are defined for multi-state objects, like switches, indicators, doors, and even the pins in an electrical connector. The current value, or state, of these variables can then be scrutinized by active behaviors to determine the resulting end behavior. VMT prototyping experience has shown that this works well for objects with simple state relationships, such as doors (opened/ closed) or connectors (connected/ disconnected). However, simulating with high-fidelity realism the function of a complex multi-state system (like a crew station's electrical system) using event-driven logical behaviors becomes an overwhelming task. This difficulty has been overcome by controlling object behaviors in the virtual environment's visual simulation with the logic management capability of an external functional simulation.

External Simulation Interface

The VMT should integrate the interactive visual simulation with an external functional simulation.

The functional characteristics of a complex system can be simulated more accurately and developed more efficiently using standard simulation software coding practices, rather than by authoring the complex system behaviors in the visual application.

A VMT prototype has been created where the visual simulation is coupled with and controlled by a simple functional simulation. While performing the step-by-step procedure of a representative troubleshooting task, the user can interact with several switches in the virtual crew station and can observe the realistic system responses depicted by indicators and gauges. During the same task, the user disconnects a connector in an equipment bay and uses the mouse cursor to probe its pins. The functional simulation controls the expected responses, which are observed on a virtual Fluke multimeter in the 3D environment.

For this prototype, the objects' motions and appearance changes are accomplished with behaviors local to the 3D visualization application, but the external functional simulation manages the conditional state logic. Interfacing the visual simulation with the functional simulation was accomplished using the DoD-standard High Level Architecture / Run-Time Infrastructure (HLA/RTI) protocol. An application plug-in based on HLA specifications had to be created for the specific visualization application being used.

One aspect of this VMT architecture is especially important. For the VMT developer, it is likely that the simulations necessary to make the virtual aircraft function realistically already exist – a byproduct of other aircraft program activities. On modern aircraft programs, significant software code is developed for use in flight simulators, or other testing or support activities. The reuse of this aircraft simulation code provides yet another opportunity to substantially lower VMT costs, and improve development lead-time and upgrade cycles.

CONCLUSION

Many training tasks traditionally assigned to expensive hardware trainers can now be accomplished on digital mockups. VR and computer technology advancements have enabled the development and use of complex, high fidelity virtual aircraft maintenance environments on commodity PC platforms. The 3D virtual maintenance trainer shows considerable potential as an effective training media, and to further the implementation of this new media, ten basic requirements considered vital to VMT development have been presented. Virtual training systems providers who develop solutions based on these requirements will better produce more realistic and effective environments for maintenance training.

REFERENCES

Barnett B., Helbing, K., Hancock, G., Heininger, R., & Perrin B. (2000). An evaluation of the training effectiveness of virtual environments. Paper presented at the Interservice/Industry Training, Simulation, and Education Conference, Orlando, FL