

# **Generating Human Machine Interfaces: How to Avoid Development Pitfalls**

**By:**

**Scott Ariotti**

**October 5, 2006**

Copyright © 2006 by DiSTI

A large, light blue version of the DiSTI logo is centered on the page. It features the word "DiSTI" in a bold, sans-serif font, with the "D" being the largest character. A registered trademark symbol (®) is located to the right of the "I".

**DiSTI**®

## Introduction

The generation and integration of Human Machine Interfaces (HMI) can be a challenge for Developers and Project Managers. The relatively simple concept of generating a real-time interactive interface can quickly derail development timelines and negatively impact design function and reuse.

Engineering professionals need to look at software solutions that can satisfy a variety of needs while reducing deployment times. This white paper explores today's driving changes and how development of HMI can be done in a rapid a reusable fashion.

## Market Drivers affecting HMI Development

The proposed Department of Defense FY 2007 R&D budget is \$1.9 Billion (16%) below the FY 2006 levels<sup>1</sup>. This trend is reflected in the overall Federal R&D budget proposed for FY 2007; a four year decline<sup>2</sup>.

The reduction in R&D dollars generally trickles down to force Project Managers to do more with less. This can also lead to an onus on engineers to reduce development times and find ways to leverage previously built assets in order to increase profit margins.

Analysts at Gartner's Application Development Summit 2006 focused one of their four themes on developing applications better and faster. The keynote speaker, Matt Hoyle, stressed that future application development is about building functionality from components where "assembling, buying, and extracting is an increasing part of what you need to do"<sup>3</sup>.

Another market driver is a shift away from paradigms utilizing proprietary data. Today's solutions need to support inputting and outputting recognized open formats and standards. A roadmap plan for the Department of Defense indicates that these shifts "enable a business process migration from proprietary products" to a marketplace that can "extend and adapt to capabilities on demand"<sup>4</sup>.

---

<sup>1</sup> Holmes, K., (Sept. 11, 2006), *EngineeringPolicy.org*, Department of Defense Science, Engineering & Technology (SET), [http://www.engineeringpolicy.org/Department\\_Defense.cfm](http://www.engineeringpolicy.org/Department_Defense.cfm).

<sup>2</sup> (Feb. 8, 2006), AAAS, Guide to R&D Funding Data – Historical Data, <http://www.aaas.org/spp/rd/guihist.htm>

<sup>3</sup> Schindler, E. (Sept 26, 2006), *DevSource*, What Gartner Is Telling Your Boss, <http://www.devsource.com/article2/0,1895,2020837,00.asp>

<sup>4</sup> Scott, J., Lucas, M., Herz, JC, (April 2006), *Office of the Deputy Under Secretary of Defense*, Open Technology Development Roadmap Plan, [http://www.oss-institute.org/NCOSPR/OTDRoadmap\\_v3\\_Final.pdf](http://www.oss-institute.org/NCOSPR/OTDRoadmap_v3_Final.pdf)

## Challenges for HMI Development

Areas of concern for HMI development are listed in the different sections below. Each problem area is wide and varied and in some cases crosses over into other problem areas.

### *Pitfalls of HMI Development in Simulation and Training*

The umbrella of Simulation and Training is extremely broad and covers all aspects of learning by way of replicating real world elements in controlled environments. Obtaining content for use in simulation and training applications can be performed in one of two ways.

First is to use the hardware from the real device in the training arena. This can be a costly solution but for certain applications is quite necessary and mandatory.



The second is a virtual representation of the hardware device developed in a software solution, rendered by low cost computers and driven by any number of input methods. This solution can lead to the following problems if a developer is not careful about the implementation:

1. The rendering performance of the virtual hardware device may not meet specifications demanded by the training environment. Performance issues may not arise during initial development phases or prototyping but later as complexity increases, and subsequently severely impacts program milestones.
2. Virtual hardware devices might be generated as a “one-off” solution, in a proprietary architecture, or be difficult to modify once initially developed.
3. Visual quality of the hardware devices may not meet the fidelity required for training.

Improvements in computer performance and new software development tools are geared at scalable solutions and work towards eliminating these problems in simulation and training virtual devices.

### *Safety Critical Application Displays*

Displays for Safety Critical applications in industries such as Aerospace, Defense, and Medical have long development times due to their inherent complexity. Typically vendors will use tools to create a new design's look, and mimic partial functionality, merely to obtain prototype sign-off.

Efforts put forth to create the prototype are then discarded as developers write the final display application by hand. Progressing in this fashion can lead to the following problems:

1. Hand coding the new display to match the prototype can take months or years.
2. The hand code can be very difficult to change or modify as new requirements come along.
3. The solution is often proprietary or in an architecture that cannot be reused elsewhere.
4. There are significant costs in managing, testing and certifying all of the code developed for the displays.



The FAA has developed a four volume handbook for OOTiA (Object Oriented Technology in Aviation). Software tools are taking advantage of this and other methodologies to improve the development cycle<sup>5</sup>.

### ***Classroom Training on Physical Equipment***

When training personnel on how to use a piece of real world equipment it is often necessary to have that device in front of them so they can operate it during the learning process. Classrooms of equipment are set up in order to facilitate training the appropriate numbers of personnel but the following problems arise from this approach:

1. Procuring all of the equipment to facilitate training can be expensive.
2. Maintaining the equipment to keep it functional for training can be expensive.
3. Monitoring *how* the student is using the equipment is difficult.
4. Travel costs to the training facility and time away from the student's primary job can have a negative impact on an organization's bottom line.



Using virtual HMI content can resolve these issues allowing students to have measured training anytime anywhere, but can lead to areas of concern that are detailed in the next section.

### ***Adding Interactive Content Adds Security Concerns***

Interactive simulation based learning content can solve many problems. However, the content can introduce problems when deployed in today's secure network environments. The delivery mechanisms to "run" the simulation content in web browsers, and other deployable learning applications, may violate corporate or government security directives.

<sup>5</sup> [http://www.faa.gov/aircraft/air\\_cert/design\\_approvals/air\\_software/ooot/](http://www.faa.gov/aircraft/air_cert/design_approvals/air_software/ooot/)

Corporations and government agencies have introduced security directives on mobile code to address these issues<sup>6</sup>. Tools to develop HMI need to address these directives to ensure that delivered content is not malicious.

## The Next Generation of HMI Development

The next generation of HMI development should take the form of a Commercial-Off-the-Shelf (COTS) development tool that can alleviate previous problems with the development of virtual devices. Application developers can leverage newer technologies and architectures such as Object Oriented designs, WYSIWIG interfaces, and applications generated in C++ or Java rendering in OpenGL.

Utilizing this type of COTS solution to develop the HMI application framework should:

1. Reduce development times
2. Decrease application complexity
3. Increase virtual device performance
4. Increase device reusability
5. Increase device fidelity
6. Broaden target platform accessibility

At its core, the tool should generate human readable object oriented code in a user desired standard format (i.e. C++ or Java). The code can be compiled and used however the developer sees fit. A tool with this type of functionality reduces development times, decreases complexity and increases reusability while avoiding proprietary formats.

HMI code should run in the OpenGL graphics environment to ensure the virtual devices run on a computer's GPU and not the CPU to achieve optimum rendering performance and minimum application overhead. A solution geared towards OpenGL rendering, and trivial operating system dependencies, allows generated code to be run on virtually any platform.

A WYSIWIG interface that supports development in both 2D and 3D environments and importing common high resolution image and model formats is critical for reducing prototyping and development times and ensuring proper device fidelity.

Final considerations to a COTS tool are licensing and execution of deployed objects. Developers should have an option to deploy applications without licensing restrictions and the

---

<sup>6</sup> Money, A., (Nov. 7, 2000), *Department of Defense*, Policy Guidance for use of Mobile Code Technologies in Department of Defense (DoD) Information Systems, <http://www.dod.mil/nii/org/cio/doc/mobile-code11-7-00.html>

ability to have content run in non-proprietary rendering solutions (especially in web deployed or distance learning applications).

## **The GL Studio<sup>®</sup> Advantage**

There are choices for developing HMI content and applications. Some have been around for a very long time. The GL Studio tool by DiSTI was born out the market demands for a better way to develop HMI applications.

GL Studio has no proprietary formats, flexible licensing options and is able to produce reliable, safe, efficient, reusable applications in a rapid and easy fashion. The tool has been extremely successful since its inception and has gained great market share because of the needs it addresses.

### ***Code Generation***

GL Studio exists to be a code generator. The editor produces highly encapsulated C++ code, Java code, or DO-178B code qualifiable to Level A for safety critical applications. All generated code is object oriented so GL Studio components can easily be encapsulated into a diverse range of applications as well as reused back into the GL Studio workflow to build larger designs.

### ***OpenGL Library Support***

GL Studio works with OpenGL libraries so all of the virtual components run on the GPU, freeing up the CPU for performing other mission critical tasks. Since the GL Studio solution relies primarily on the GPU, applications can be compiled and executed on a variety of Windows, Linux, IRIX, Solaris, and any Real-Time operating systems.

### ***Ease of Use***

The GL Studio editor is an intuitive 2D or 3D GUI development environment for creating virtual devices and interfaces. Tools include drawing, texturing and hierarchy controls along with pre-built behaviors for a variety of common objects such as push-buttons, knobs, sliders and switches. Users can also add their own complex behaviors. GL Studio can be used to replicate the look and feel of any real world device.

### ***Deployment Options***

A single GL Studio design can be built to deploy applications based on the C++, Java, or DO-178B Qualifiable code generators. These applications have the option to be built with no runtime licensing dependencies at all (i.e. they will never check for a license).

Applications can be stand alone executables that run in their own environment, components to plug into 3<sup>rd</sup> party applications, or web deployed. GL Studio objects can run in an ActiveX based web player or in the Java Runtime Environment. For developers who need to run on the Naval Marine Corp Internet (NMCI) the GL Studio web player is a Certified NMCI Plug-in and the Java Runtime Environment is currently on the NMCI Gold Disk.

### **Try GL Studio for Yourself**

GL Studio is available at [www.glstudio.com](http://www.glstudio.com) for 30-day trial periods. We encourage you to try the tool and learn more about these advantages. More information about GL Studio is available in the GL Studio Primer along with a variety of downloadable applications on the web site.

If you would like to contact a member of our staff to answer any particular questions, arrange an on-site presentation, or ask about a web based meeting please feel free to call us at 407-206-3390.