MH60S/R Helicopter Multi-Platform & Web-Based Trainer with Dipper Acoustics

Robert Richards, Ph.D.
Stottler Henke Associates, Inc.
San Mateo, CA
richards@stottlerhenke.com

ABSTRACT

The US Navy's PMA-205, in conjunction with the training and simulation industry, has benefited the Navy by developing and deploying the *Operator Machine Interface Assistant* (OMIA) part-task trainer for the MH-60S and MH-60R helicopters. This flexible, multi-platform, web-based crew trainer is currently in use by HSC-2, HSC-3, HSM-41, Navy helicopter maintenance organizations, and by fleet squadrons at port, home and when deployed at sea. Earlier versions have been in use for over nine years. OMIA benefits the warfighter by providing a simulation of the MH-60S/R Common Cockpit that includes a FLIR capability with optional hardware FLIR hand-control unit and acoustics simulations. This paper will describe the design decisions made to OMIA during the development of the training application so it would not require any special user rights to install. It will also discuss the unforeseen side benefits the Navy has realized beyond its original development intent. Design decisions included porting the OMIA to Java enabling deployment as both a portable stand-alone application for PC, Mac and Linux, as well as a web application that can be run through a web browser. Side benefits include the device being picked up by maintenance organizations, as they need to understand the cockpit as part of their activities. Currently, OMIA continues to benefit the Navy by providing training that can be accessed wherever and whenever it is needed, and thanks to this flexibility its use has expanded far beyond its original scope via ongoing funding. A recent capability expansion is the addition of sophisticated acoustics, whereby acoustics training previously only available in limited availability TOFT trainers is now available via the Acoustics Training System (ATS) version of OMIA.

ABOUT THE AUTHORS

Robert Richards, Ph.D. is a Principal Scientist and Project Manager at Stottler Henke. Dr. Richards is managing and has managed multiple projects for both commercial and government clients, including various intelligent-tutoring-system-based training projects. He is the principle investigator for OMIA, a PC-based desktop training system that teaches crewmembers the Navy's new MH-60R and MH-60S helicopters. Dr. Richards has taken the project from a Research and Development SBIR project to a deployed training tool that has been awarded multiple IDIQs totaling over $12 million. Dr. Richards received his Ph.D. from Stanford University in mechanical engineering with an emphasis on machine learning and Artificial Intelligence.

MH60S/R Helicopter Multi-Platform & Web-Based Trainer with Dipper Acoustics

Robert Richards, Ph.D.
Stottler Henke Associates, Inc.
San Mateo, CA
richards@stottlerhenke.com

INTRODUCTION

The US Navy has introduced two new helicopters, the MH-60S and MH-60R. Both of these helicopters utilize Lockheed-Martin's Common Cockpit design. The Common Cockpit includes all the flight and mission instrumentation in both of the helicopters and enables both the pilot and co-pilot to share workload through dual flight and mission instrumentation. As can be seen in Figure 1 the pilot and copilot each have two LCD screens, one of which is the Mission Display (MD) and the other is the Flight Display (FD). The pilots interact with these displays primarily through a set of bezel keys around each display and a keypad located in the center console. This keypad contains a set of fixed function keys (FFK), a set of context-dependent programmable keys (PK), and a small joystick known as the "hook".

For more than eight years the US Navy's PMA-205, in conjunction with industry, has developed, deployed, and updated a flexible, low-cost PC-hosted crew trainer for the Navy's MH-60S (Sierra) and MH-60R (Romeo) helicopters called the *Operator Machine Interface Assistant* (OMIA).

Figure 1. MH-60 Common Cockpit

OMIA PART-TASK TRAINER

The purpose of this paper is to discuss 1. The architectural decisions of the trainer in order to maximize training accessibility; and 2. The unanticipated no-cost benefits that have come out of the deployment of the training device. The core OMIA (Figure 2) is a standalone Java program that operates under any standard Windows (XP, Vista, and 7), Linux or and Macintosh computer that is installed with a Java Runtime Environment (JRE); it is also compatible with Navy/Marine Corps Intranet (NMCI) computers. The standalone OMIA provides an introduction to the Common Cockpit, including the Mission Display (Figure 3), the Flight Display (Figure 4), the Center Console's Fixed Function and Programmable Keys (Figure 5), and several helicopter control panels. A major benefit of the standalone core OMIA trainer that the Navy requires, is that it application uses no external licensing, and therefore it can be distributed freely to anyone in the US Navy via compact disc or the Web. However, the core system also supports a number of optional extensions to meet additional training needs.

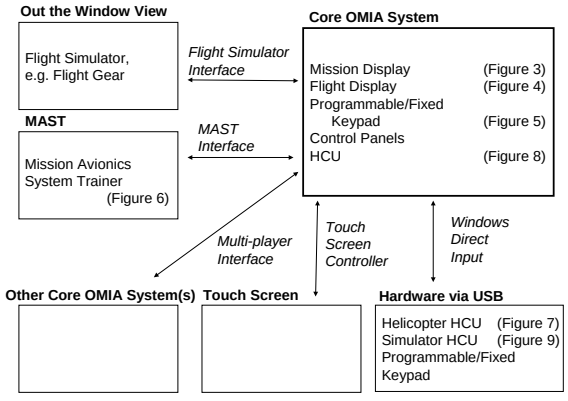

Figure 2. The Core OMIA System and Optional Extensions

The core OMIA program can be used to teach Navy pilots how to operate both the Sierra and Romeo versions of the helicopter. A different executable is created for each configuration. Presently there are two for the MH-60S (armed and non-armed) and two for the MH-60R (pilot and sensor operator). In addition, the user can also run in standalone mode (the default) or in network configuration. In a network configuration, one operator can be the pilot and another operator can be the co-pilot or sensor operator. In this scenario, both operators will see the same world, including changes made by each other.

Figure 3. OMIA Mission Display with Menu Visible

The architecture is designed to allow the trainer to automatically configure itself to deliver the most optimum training environment possible. This includes the ability to sense when multiple monitors are connected, presence of a touch screen display, and connection to a supposed flight simulator environment. OMIA and Windows are able to detect if optional hardware is installed and will work correctly with new hardware automatically. The simplest example is multiple monitors: by attaching two displays, the Mission Display, shown in Figure 3, and the Flight Display, shown in Figure 4, can be displayed on separate monitors, with one of the monitors also displaying the Center Console. Another option is to have one or more of the screens made a touch screen as is done in the Mission Avionics System Trainer (MAST), described below and shown in Figure 6**.** The bezel keys on the flight display and mission display in the MAST are operated using finger pushes on a touch screen in order to more accurately emulate the ergonomics of the actual helicopter. Of course, the third screen containing the Center Console panels could also be a touch screen so that the user could push the buttons in a way more similar to how it is done in the aircraft instead of using the mouse.

A new software addition for OMIA is the ability to integrate with a third-party flight simulator. Every time OMIA starts it checks to see if a compatible flight simulator (e.g., Microsoft Flight Simulator, Flight Gear) is already running. If it is running, OMIA attaches itself to the flight simulator and then gets its position, speed and other flight information.

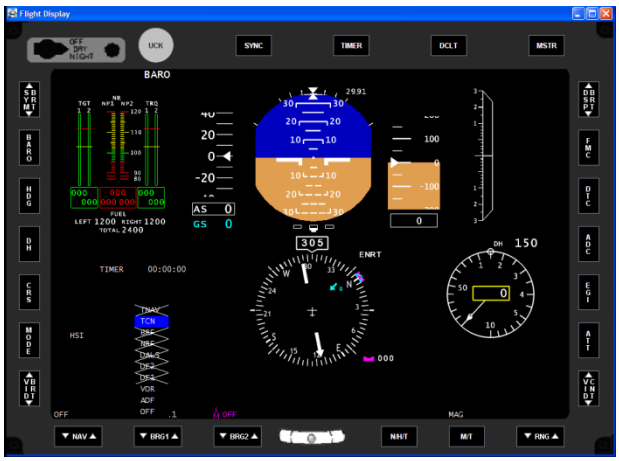

Figure 4. OMIA Flight Display

In this configuration, the user is able to have the external view be completely generated by the flight simulator. While the Flight Display, Mission Display and all of the other panels are still being used from the core OMIA. However, any other information such as ground speed, latitude/longitude location, or motion is all taken from the flight simulator. This is very beneficial if a user wishes to fly or see the terrain while navigating a search and rescue pattern. As one navigates, the helicopter may be guided along the search and rescue pattern on the Mission Display, and as search and rescue points are reached or captured the pattern will update appropriately. When using a flight simulator, other hardware can be used if desired, such as a joystick, or a head mounted display with head tracking may be added to improve the means for emulating the full field of view.

Flying can be performed solely using a joystick, or a joystick and a separate control for the collective, or COTS pedals could be added. More information on the details of interfacing with Microsoft Flight Simulator is provided in Richards and Ludwig (2007).

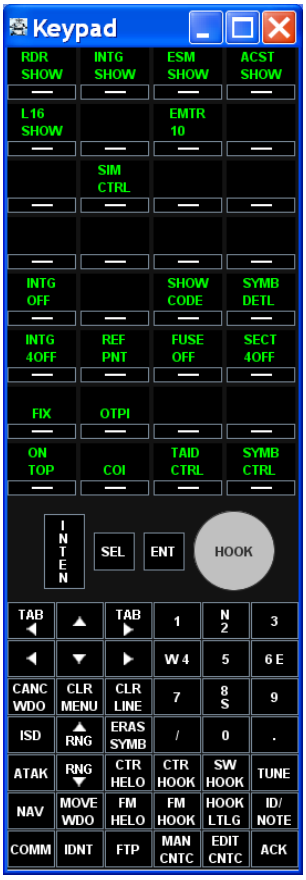

Figure 5. Programmable/Fixed Function Keysets

Converting From C++ to Java

Until 2007 this software was written mostly in C++, which was the best programming language for the challenges at the time. A description of the C++ version is provided in Richards and Ludwig (2007) and Ludwig (2006). However, due to various circumstances OMIA was converted to Java in 2007. This work is described in more detail in Richards and Ludwig (2008).

There were five main factors that drove the decision to convert OMIA from C++ to Java. First, the flexible design for evolving requirements is necessary because the Common Cockpit continues to evolve. Even though the MH-60S and MH-60R both use the Common Cockpit, the helicopters have different capabilities and missions, thus many operations are different on the two platforms. However, a programmable keyset (PK) supports the differences. In addition, the software for the two platforms is not always at the same version. The Navy supports these differences in OMIA. Since this process will be continuing for years it is always best to have the software in the most flexible language for this task. Advances in the Java language and tools have made Java a great choice for rapid modification.

OMIA has been able to, and must continue to work with and control Microsoft™ Flight Simulator when it is available, to use COTS and/or custom hardware when attached, and to still function as a complete standalone application. In addition, the C++ version of OMIA software was the software component of the MAST; this has been replaced by the Java version.

Second, the option to go to Java was facilitated when a graphical user interface tool supplier that OMIA uses released a Java version of their tools. Without this option, the rest of OMIA could have been converted to Java and some of its benefits could have been realized, but it would still not be able to run on Navy/Marine Corps Intranet (NMCI) machine.

Third, one of OMIA's main goals has always been its accessibility by computers on both land and at sea. The default computer configuration in the Navy, NMCI (Navy/Marine Corps Intranet), have restricted access and due to which, certain aspects of the C++ version of OMIA could not be used easily on NMCI machines. But with a Java version, NMCI compatibility could be provided as the required Java Runtime Environment is already installed on the NMCI machines.

Fourth, besides keeping up with the helicopters' changes, OMIA is constantly being enhanced. One of the enhancements is the client-server design change, so that multiple users can operate OMIA concurrently. For example, a set of users can match different seats in one helicopter, and multiple helicopters can also be handled so everyone is playing in the same world.

The fifth advantage of the new OMIA is that it can be easily ported to other platforms (e.g., Linux). The Navy requested a Linux version of OMIA in 2009 as part of an acoustic systems trainer enhancement. The fact that OMIA was written in Java allowed it to fill this unexpected request rapidly.

Mission Avionics System Trainer (MAST)

The MAST, as shown in Figure 6, includes actual hardware in the Center Console that is the exact aircraft hardware or a very close facsimile of it. The MAST hardware was procured by the Navy from an industry supplier. One can actually push physical buttons, change actual knob positions, feel feedback, open covers, etc.

There are two seats, the pilot and co-pilot. Each seat has two screens just as in the helicopter, one for the

Flight Display and one for the Mission Display. There are individual screens for the pilot and co-pilot showing the outside view, generated by Microsoft Flight Simulator. There is a simple cyclic in the MAST and the screens for the Flight Display and Mission Display are touch screens. Another feature of the Center Console hardware is the actual hook hardware, used to control the cursor in the Mission Display.

Figure 6. Mission Avionics System Trainer (MAST)

The MAST is a medium resolution trainer driven completely by OMIA and Microsoft Flight Simulator software. Again, there is only one version of OMIA, and it can work with or without MAST hardware. The MAST can be used for many different types of operations, including coordinated operations, because, as described above, the two seats can be used independently or in conjunction (client/server mode) so that the pilot and co-pilot are flying the same mission. They are mainly intended for Sierra training; however, since they are being completely driven by OMIA software, they can be quickly reconfigured as Romeo stations via restarting the programs in Romeo mode.

OMIA as a Portable and Web Application

A portable application, or 'portableapp' is a software program that does not require any kind of formal installation onto a computer's permanent storage device to be executed, and can be stored on a removable storage device such as a CD-ROM, USB flash drive, flash card, etc.; this enables it to be used on multiple computers. **T**he portableapp reads its configuration files from the same storage location as the software program files.

Most software for Microsoft Windows is not portable, because the installation requires using the Windows Registry, etc. That is, if one installs an application that is 'installed' in a folder and copies the folder to another computer, usually the software will not run on the second computer (where by contrast, a portableapp would). However, it is worth noting that since OMIA is written in Java, in some regards it is NOT completely portable because there needs to be a Java Runtime Environment (JRE) on the computer that OMIA runs on. The JRE is freely available and many machines already include it, including all NMCI machines. However, the default JRE is not portable. Nevertheless, work is being done to make a portableapp JRE. For the rest of this discussion, OMIA Java will be referred to as a portableapp.

By making OMIA a portable application, the Navy receives many benefits. A major advantage that has already proven itself very valuable is the ease of distribution and 'installation'. Since a portableapp does not require formal installation it can be used directly from the distribution media as long as the media is writable. Therefore, a USB drive containing OMIA can be plugged into a computer and the OMIA software will run directly from it. This is not the case for a CD disk, since the disk is read only. However, in both cases the OMIA directory can be simply copied to anywhere on the computer and then run from that location. This has made distribution to training classrooms effortless compared to the previous C++ OMIA, which required an installer, especially when automatically pushing installs to networked computers.

NMCI compatibility, as already mentioned, is a huge benefit to the Navy provided by OMIA's being a *portableapp*. Previously, users would have to go to a lab when they wanted to utilize OMIA even though many have an NMCI computer at their desks and/or a personal computer. There is a huge benefit in making OMIA available at a user's desk or on their personal computer; in addition to anywhere they have access to a browser.

Another benefit to the Navy of OMIA being written in Java is that it could be run as a Java web-start application. That is, not only is OMIA a *portableapp*, it is also a web-based application.

FLIR ENHANCEMENTS

OMIA continues its earlier history, where each version not only matches the evolving helicopter software, but functionality progressively expands as well. Thus, for the most current version of OMIA, it has added a FLIR (Forward Looking Infrared) capability.

The FLIR user mainly controls the FLIR operations via a Hand-Control Unit (HCU), as shown in Figure 7. The Navy has developed a portable HCU that uses the actual helicopter's HCU, but connects to a USB controller with a USB connector. This portable training HCU has been interfaced to OMIA.

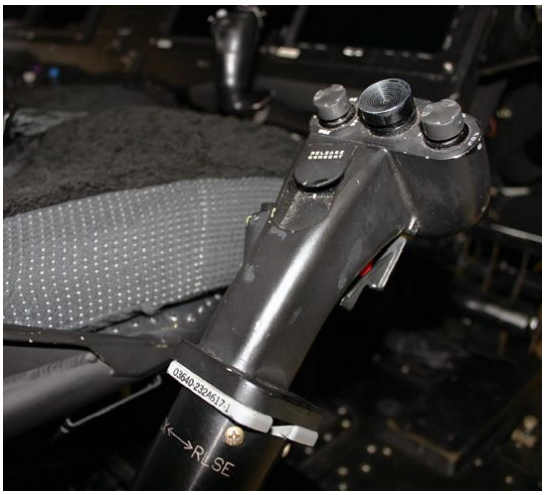

Figure 7. FLIR Hand Control Unit (HCU)

OMIA reacts the same way to the HCU hardware as it does to the presence/absence of other hardware units; when OMIA starts up it seeks whether a FLIR HCU hardware is attached. If it is attached, the software will read input from it, if it is not detected, then a software equivalent is provided (Figure 8).

However, neither of these HCU solutions meets all of the training requirements. The actual hardware is too expensive per HCU unit, while the simulated hardware does not provide the tactile feedback and muscle memory of a physical hand control unit. To address these issues, OMIA has been integrated with a second, low-cost, HCU based on technology originally developed for simulators rather than using an HCU from the helicopter (Figure 9). Due to the modular design of the OMIA software, adding support for an additional USB HCU input device required relatively little development effort.

Figure 8. Simulated HCU in OMIA

Figure 9. Low-cost HCU built by Metters Industries

An example of a FLIR, with the MH-60 overlay, as shown on a Mission Display in OMIA is shown in Figure 10. The generation of FLIR images is a difficult task in real-time. Usually FLIR simulators are very expensive units incorporated into multi-million dollar simulators. For OMIA a simpler solution is created to provide a high level of learning benefit sans the cost.

The FLIR implementation in OMIA uses a 2D FLIR image. Much of the learning related to FLIR concerns the operation of the FLIR menus and other operations that are part of the software overlay. Through the combination of the hardware FLIR HCU and the overlay menus and other functions, a great deal of learning is facilitated. For example, users can zoom in and out, slew, adjust image polarity, cycle through camera modes, and navigate through on-screen menus.

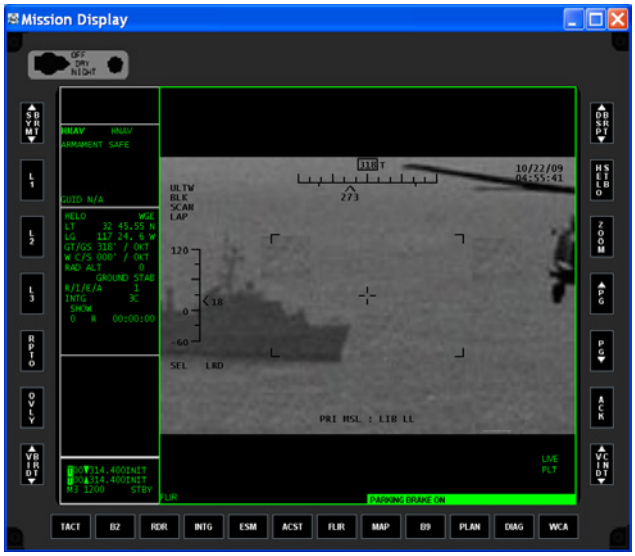

Figure 10. Screenshot of 2D FLIR in OMIA

DIPPER ACOUSTICS

During 2009 and 2010, OMIA was enhanced to include MH-60R Airborne Low Frequency (ALFS) Sensor Modeling and Simulation Training. This ALFS Trainer Simulator (ATS) enhancement resulted in OMIA-ATS. This capability is mainly used by the Sensor Operator (SO) that sits at the SO station; see Figure 11. OMIA-ATS requires capabilities currently only available under Linux, so OMIA-ATS is restricted to use on the Fedora Linux OS presently. This enhancement supplied the Navy with a complete hardware/software solution, in that fully configured laptops with an external touch screen and OMIA-ATS installed were provided. The generic OMIA benefited as much as possible from the acoustics enhancements, so OMIA allows the learner to get an introduction to acoustics operations and menus, but does not allow for pinging and viewing realistic acoustic returns as OMIA-ATS does. A screenshot of OMIA running in the SO mode is shown in Figure 12 with the Cable Angle dialog open (the region with the concentric circles).

Figure 11. Sensor Operator Station

Figure 12. OMIA ATS Running in SO Mode

OMIA-ATS operates in four different modes:

Simulated Noise Mode
Create Scenario Mode
Play Scenario Mode
Analyze Data Mode

- Simulated Noise Mode: a student can exercise all acoustic keyset and display functions. The acoustic return in simulated noise mode is shown in Figure 13.
- Create Scenario Mode: Allows an instructor to generate a scenario to train a student. The mode records all instructor key presses and cursor movements; processes and displays a recorded

acoustic data file; and records annotations made by the instructor to "explain" the lesson.

- Play Scenario Mode: Allows student to train from an instructor's generated scenario; a student can complete the lesson with or without an instructor present; a student can complete the lesson at his own speed; lessons expose the student to recorded acoustic data (real target signatures); and choice of playback options – Automatic or Guided.
- Analyze Data Mode: Enables an instructor to search through recorded data sets to find pings of interest.

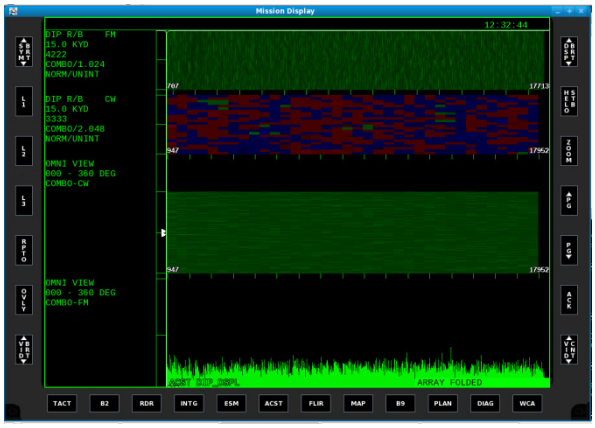

Figure 13. Acoustic Return in Simulated Noise Mode

The purpose of the ATS system is to provide a training tool for MH-60R sonar operators. The ATS can support any of the following:

- Provide system familiarization for displays and keysets (with or without instructor support)
- Support independent skills review in a Training facility environment
- Prepare operator to take full advantage of TOFT training sessions
- Provide a platform for acoustic return recognition training
- Support independent skills review for deployed squadron personnel
- Support immediate Post Mission analysis (once the 60R platform gets a flight recorder)

This new OMIA-ATS capability is a departure from most other functionalities developed for the Navy, in that it provides in-depth capability. Usually OMIA is used as an introduction to an aspect of the helicopter so as to prepare for and thus save simulator or actual helicopter time. This benefits the Navy by saving millions of dollars of simulator/flight time. However OMIA-ATS provides even more benefit to the Navy because, now, for active dipper operations, OMIA-ATS is actually replacing simulator and helicopter time.

CONCLUSION

The complexity as well as the number of the sensors under control by the crew on the MH-60S and MH-60R helicopters, pose a difficult training task for the Navy. To meet this challenge, the US Navy's PMA-205 in conjunction with industry support has developed and deployed OMIA, a flexible, low-cost PC-hosted desktop crew trainer. OMIA has evolved with the changing helicopter software; with every iteration, it has become an ever more functional trainer providing greater flexibility and benefit to the Navy.

The latest version of OMIA demonstrates the utility of designing a flexible system that allows for quick responses to ever-changing demands. As presented in this paper, the OMIA program has been able to adapt to new challenges with a minimum amount of additional development. One such challenge was an unexpected request to run on Linux; another challenge was the addition of a new physical hand control unit for FLIR capability.

OMIA is available for training anytime on both land and at sea. To learn more regarding the past, present and future of OMIA, as well as on access for Navy personnel, please contact one of the authors.

REFERENCES

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Ludwig, J. (2006, April). *Comparing helicopter interfaces with CogTool.* Paper presented at the 7th International Conference on Cognitive Modeling, Trieste, Italy.

Richards, R. & Ludwig, J. (2007). PC Rapid Modification Tool for Aircraft Experimentation & Training for the MH-60S/MH-60R Helicopters.

Proceedings of the 2007 IEEE Aerospace Conference. IEEE.

Richards, R., & Ludwig, J. (2008). Training Benefits of a Java Based Part Task Trainer. *Proceedings of the 2008 IEEE Aerospace Conference*. IEEE.