REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate only, other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (07804-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(LEAVE BLANK)*	2. REPORT DATE 1998	3. REPORT TYPE AND DATES COVERED Professional Paper

4. TITLE AND SUBTITLE
A Cross-Platform Multi-Simulation Software Executive

5. FUNDING NUMBERS

6. AUTHOR(S)
James Nichols , Thomas J. Magyar, and Eric C. Schug

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Air Warfare Center Aircraft Division
22347 Cedar Point Road, Unit #6
Patuxent River, Maryland 20670-1161

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS

15. NUMBER OF PAGES

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	SAR

A Cross-Platform Multi-Simulation Software Executive

James Nichols[+], Thomas J. Magyar[*]
Naval Air Warfare Center Aircraft Division
Patuxent River, Maryland

Eric C. Schug[**]
SAIC / Simulation and Research Services Division
California, Maryland

Abstract

The Manned Flight Simulator (MFS) was created to provide high fidelity simulation capability and support for the Navy's fleet of aircraft. The facility contains five simulation stations, which share a common interface to the computer networks, visual image generators and actual aircraft flight hardware. Any cockpit at MFS can be used in these simulation stations using a "roll-in, roll-out" concept to easily transfer a cockpit from one simulation station to the next. With this interface, a simulation executive was required to run the multiple simulations at any simulation station. The Controls Analysis and Simulation Test Loop Environment (CASTLE) was developed to meet this requirement. Since this initial requirement, CASTLE has greatly expanded and includes many tools for use during a real-time piloted session or for desktop development and analysis use. With new development, popularity and increased performance of computers, CASTLE now operates on a variety of platforms and operating systems using the same source code and graphical user interface (GUI). These operating systems include SGI-UNIX, HP-UX, DEC VMS and Windows95/NT.

Background

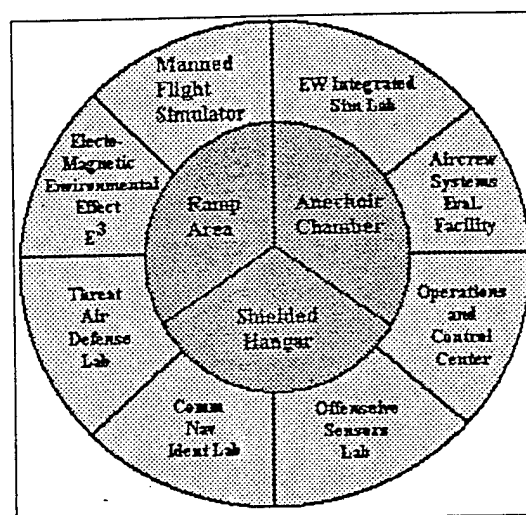The Manned Flight Simulator (MFS) is one facility within the Air Combat

Figure 1: ACETEF

Environment Test And Evaluation Facility (ACETEF) at the Naval Air Warfare Center Aircraft Division (NAWCAD) in Patuxent River, Maryland. The ACETEF complex is a collection of laboratories that are used to emulate the complete battlefield environment, including both friendly and hostile elements. These laboratories (see Figure 1) are often connected to various other facilities and test ranges around the world for joint exercises, and may also be operated separately for local testing.

[*] Aerospace Engineer, Member AIAA
[**] Aerospace Engineer
[+] Aerospace Engineer, Senior Member AIAA

MFS provides the high fidelity, piloted airframe simulations for such tests.

MFS is also used to directly support flight test, control law development, avionics development and integration, and new aircraft technology. Both rotary wing and high-performance fixed wing aircraft are modeled.

The simulation stations within MFS include a motion-capable platform, a 40ft diameter dome, a helmet-mounted virtual display and two other engineering development stations. The simulation stations share a common interface to the computer networks, visual image generators and actual aircraft flight hardware enabling them to be switched among the simulation stations. MILSTD1553 busses are used to interface mission computers and flight control computers with the lab.

Each simulation cockpit was developed with a "roll-in, roll-out" capability with common structural footprints and electrical interfaces. This allows each cockpit to be transferred from one simulation station to another in a matter of minutes along with its actual flight hardware and desired visual system and processors. Cockpits that reside at the facility include the F-14, F-18A/D, F-18E/F, V-22FSD, V-22EMD, and a Multi-Reconfigurable Cockpit (MRC) that can accommodate the throttle, collective, and stick of most aircraft.

The multiple simulations, cockpits and simulation stations required a "generic" open-architecture simulation executive that could run all of the existing and future hi-fidelity simulations at the facility. The Controls Analysis and Simulation Test Loop Environment (CASTLE) was developed to meet these requirements. CASTLE can interface with the labs network of computers and control any real-time (piloted) simulation, or be used at the engineer's desktop for development and analysis with its built-in tools.

First Generation Architecture

The first generation of the CASTLE simulation executive ran solely under the VMS operating system on DIGITAL VAX computers using the FORTRAN programming language. However, the software was created with a modular design so it could grow with the changing technology. It is still in use today and supports the premier Navy aircraft including the F-18 and V-22 in their development phase and F-14 life cycle support. This version of CASTLE has been used for thousands of hours of piloted simulation sessions annually, and for uncounted hours of desktop engineering analysis.

Next Generation Design Goals

Changing technologies in computer operating systems and increased user requirements made it necessary to update CASTLE to use industry-standard features. The most important criteria was to write code that would be platform-independent, using one code set for multiple platforms and computer operating systems. This was accomplished by using the standard programming languages as appropriate. FORTRAN77 and FORTRAN90 were used for the math model of the
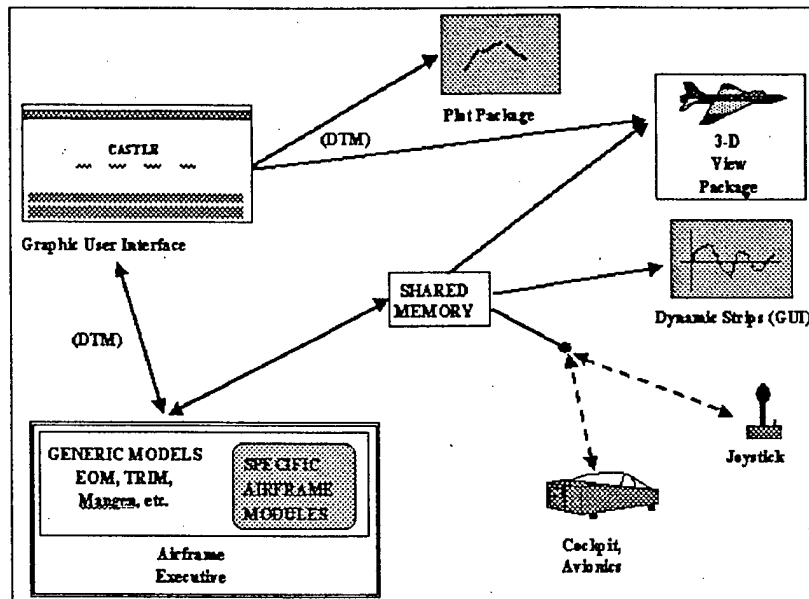
Figure 2: XCASTLE Design

airframes, and C/C++ was used for the CASTLE architecture and user interface. Another important consideration was retaining all desirable functionality of the earlier CASTLE versions.

A graphical user interface environment was built into CASTLE using the standard X-Windows/Motif user interface tools. This gives the software a familiar user-friendly interface environment and also allows non-intrusive operations during simulation sessions. This GUI code is independent of the airframe executive and is executed as a separate task from the airframe. It can also execute distributed tasks and can accept drop-in tools for expanded capability. This GUI-based software environment is now referred to as XCASTLE, although references to CASTLE are considered to be synonymous.

Overview Of XCASTLE Design

The design of the XCASTLE software is very modular on several levels. It consists of a collection of tasks running simultaneously, with the XCASTLE GUI being the controlling entity. Other processes include the actual airframe executive, plotting packages, a spatial visualization tool, interfaces to the piloted laboratory environment, and various other utilities. These tasks communicate via several methods, which include shared memory and a TCP/IP protocol. Figure 2 describes some of these tasks and the communications methodology. It should be noted that both TCP/IP and shared memory communications can be utilized by any particular process. The TCP/IP protocol used is the Data Transfer Mechanism (DTM) designed by the University of Illinois National Center for Supercomputing Applications (NCSA), and is hosted on a variety of computer platforms. The use of this protocol allows the XCASTLE tasks to be running on separate machines. Data word format conversions of the information packets transferred between these tasks are performed automatically. The shared memory can be resident on one machine for tasks running solely on that machine, or can consist of "global"

shared memory such as ScramNet. The shared memory read/write interface will also perform all necessary data word format conversions. These conversions are an important consideration since the large variety of computers used rarely has the same physical data representation.

XCASTLE GUI Description

The design of the graphical user-interface of CASTLE consists of re-usable C++ classes built on top of the X-Windows/Motif user interface. All the CASTLE facilities are built on base classes that handle commonly performed tasks such as defining the GUI, loading and saving settings and sending information via DTM to the appropriate sub-process. New facilities can be quickly generated by using these base classes. The use of settings files enables the user to restore previous sessions and to save the current session settings. A scripting language is available for performing automated tasks and is also used to perform "macro" operations, which is quite useful for running the jobs in the background.

Besides acting as the master process, the XCASTLE GUI allows the user to interactively define aircraft initial conditions settings, create data flow between airframe variables and cockpit controls, determine data storage requirements, and control the various analysis tools built into the XCASTLE system. Digital data storage and plotting is also controlled from the GUI. A Data Dictionary is available that allows the user non-intrusive access to all documented variables in the current airframe database. Variable values can be modified during a simulation run without suspending operations.

Design Of The Airframe Executive

The airframe executive consists of a "shell" of common (or "generic") modules, and a collection of airframe-specific modules linked into a software image referred to as the airframe simulation. The shell code is split roughly into two parts, the first being the looping architecture executed during airframe motion propagation, and the second being an extensive analysis tool set. The majority of this shell code is written in ANSI-C, with the equations of motion and related software still in FORTRAN77. The airframe developer can replace any XCASTLE generic module with a customized version, although this is not a recommended practice for configuration control reasons.

The specific modules that describe a particular airframe are supplied by the airframe developer, and may be written in any language. The most common language is FORTRAN, although several models hosted at MFS are in ANSI-C and Ada. The airframe executive was designed to be flexible enough to rapidly accommodate widely varying model structures. The airframe developer may choose to convert the model into a standard format that can then be operated on by the various XCASTLE development tools, or to merely encapsulate an unmodified model with interface routines. The preferred method is to reformat the model as this allows the user full access to all of the XCASTLE variable tools and provides self-documentation.

One of the most powerful features of the XCASTLE airframe executive is the ability to define the loop architecture interactively. This allows linking in

development versions of code and switching back and forth between development and baseline code without re-linking the executive. It also allows the user to create multi-loop frames, and have positive control over execution order. A different loop structure can be used for the run mode, trim mode, and other analysis modes.

The core of the XCASTLE structure is the variable database management system. It consists of a graphic-based editing tool (EDITCOM) and a pre-compiler (CASCOMP). Once the developer has defined a variable database, the pre-compiler is used to generate equivalence statements (FORTRAN) or macro definitions (C/C++) in the target source code. A special assignment routine is generated during the link procedure, which is used to equate the ASCII-based variable database with the address of the variable in the software. This provides a flexible, robust, completely device-independent method of variable access.

Another development tool available with XCASTLE is the Function Table Processor, which is used to convert tabular data into executable source code. This software can interpret function data in many different industry formats, and uses linear interpolation to return function values.

XCASTLE Analysis Tools

XCASTLE encompasses many tools to perform engineering analysis on a simulation during both piloted and non-piloted sessions. It should be stressed that the exact same code set is used for the cockpit-in-the-loop piloted sessions and for non-realtime engineering sessions so analysis procedures are valid

for either mode. New tools can be added easily due to the modular design of XCASTLE.

The data storage capability of XCASTLE saves the values of selected variables during a simulation run into dynamically allocated internal memory. Multiple data sets can be stored at varying rates and output to files in pre-defined formats, including ASCII text column format, Matlab file formats, and user-defined formats. Any variable available through the variable access database in the simulation can be saved, and as many variables as desired can be saved during a run. This is only limited by the memory constraints of the host computer.

The Maneuver Generator (ManGen) is used to overdrive variables in the simulation. The data used to drive these variables can come from several sources. Several pre-defined functions are available to the user in the Maneuver Generator, including step inputs, ramp inputs, varying sinusoidal inputs, and doublets of these functions. Time history data from flight test or other simulation runs can be loaded into XCASTLE and used to drive any variable during a simulation run. The user can also interactively define a function by creating curves with the plotting package. Multiple functions can be summed into one variable driver.

The Trimming facility uses a one-sided perturbation of each user-defined control until the defined state derivatives have reached a target value. The user can set control limits and error tolerances on the target values. The algorithm is robust enough to allow under-controlled sets as long as the uncontrolled state derivatives are already within tolerance. The

simulation can be trimmed to straight and level flight, steady climb/descend and specified angle of attack conditions. The user can also select non-zero target values for the state derivatives and "trim" to accelerated conditions. Non-standard trim controls, such as engine thrust, can be used to artificially trim the aircraft to normally unobtainable conditions in order to extract linear models and perform similar tasks.

The Linear Model Extraction (LME) facility allows the user to define a linear model structure by specifying the states, state derivatives, inputs and outputs. The user can designate which airframe modules are called, and in what order they are executed. The Linear Model Extraction facility will perturb the states and inputs with user-defined perturbation values and obtain the best approximation of the partial derivative for each matrix element in the A, B, C and D matrices for the state-space equations:

$$\dot{x} = [A]x + [B]u$$
$$y = [C]x + [D]u$$

The outputs can be saved to an ASCII text file or saved as a MATLAB file for further analysis. This tool has proved to be invaluable, as it operates directly on the same airframe code set that is used for piloted sessions and control law development in their natural element. Several airframe projects have made extensive use of this feature with excellent results.

The Sweep facility allows the user to exercise the model to look at trends and discontinuities in the model database. The user specifies a range of variable values and increments as inputs and also specifies output variables. The simulation will run through these inputs and display the outputs for each. The output can be saved to a file, or loaded back into XCASTLE for plotting and further analysis. When used in conjunction with the scripting facility, the sweep facility is a powerful tool that can be used to automate repeated tasks using different parameters for each iteration. This can be especially valuable for creating a "trim" map of the flight envelope, or extracting linear models at various points throughout the envelope.

The Real-Time Processing (RTPS) Link is used at MFS to drive a simulation and its visual systems with flight test data during an actual flight test. This is useful for providing parameters that are difficult or impossible to compute at the RTPS facility, and for simulation validation. The real-time link can also drive engineering test stations (ETS) with data from a piloted simulation at MFS for ETS display development and configuration or for pilot familiarization, test envelope expansion and flight test engineering training.

The Integrated Data Evaluation and Analysis (IDEAS) is a separate package that is designed to enhance productivity in the area of advanced flight test data analysis and model development. It is an open architecture system that includes comprehensive database management tools, various System Identification tools, and other user-supplied analysis tools. The primary focus of IDEAS is to rapidly utilize flight test data to identify and improve simulation performance, but the versatility of IDEAS lends itself to many other applications.
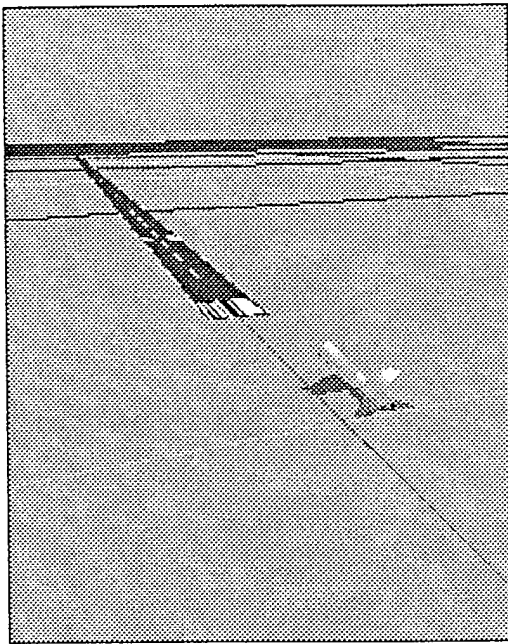
Figure 3: CasTools Chase View

CasTools is an OpenGL graphics software package that was developed to visualize aircraft trajectories and attitudes in a 3D virtual world. This visual aid may be attached to any external data source, but is normally tightly coupled to a specific XCASTLE airframe simulation session. An unlimited number of viewing positions and orientations can be defined and saved by the user. These options include a pilot view, a fixed viewpoint, and a moving viewpoint attached to the aircraft model. Other features include a Heads-Up Display (HUD), a runway, wingtip ribbons and trails, flight path and ground trajectory paths, and altitude bars. In addition to viewing the simulation run in a passive mode, the simulation can be flown with PC-style joysticks, rudder pedals and throttle controls using the built-in joystick capability that exists in the SGI and Windows NT versions of XCASTLE. Trajectory history data files can be loaded into CasTools for viewing and multiple data sets can be viewed simultaneously. Using CasTools,

simulation variables can be plotted during a real-time run on dynamic strip-charts that update as the simulation is running. Multiple data sets and multiple strip-charts can be added to view an unlimited number of variables.
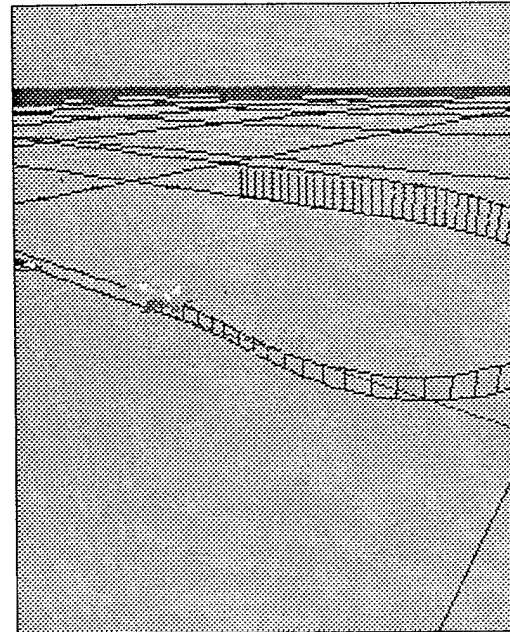

Figure 4: CasTools Flight Path Markers

Computer Platforms

CASTLE was developed initially on Digital Equipment Corporation (DEC) computers using the VMS operating system. Due to external customer demand and the popularity and increased performance of other computer and operating systems, CASTLE was completely re-designed to run on various computer systems. By using ANSI-standard and POSIX-compliant source code CASTLE is available on DEC Alpha computers running the VMS or Windows NT operating system, Silicon Graphics Inc. (SGI) workstations and Hewlett Packard computers running the UNIX operating system, and Pentium class PC's running Windows95 or Windows NT Workstation. XCASTLE

is currently being developed for use on the Apple Macintosh operating system. The Motif X-Windows system was chosen for the Graphic User Interface (GUI) because of its wide usage on various computer systems. The current XCASTLE software is identical for all of the above platforms. The specific computer system or a third-party vendor provides low-level code, such as X-Windows.

XCASTLE Users

XCASTLE was originally developed primarily for use within the MFS facility to support piloted simulation sessions, as well as for engineering analysis and research and development in the desktop environment. The diverse and extensive requirements that MFS is responsible for have ensured that XCASTLE has evolved into a versatile tool with many users and applications. Over the years, XCASTLE has been used to support a large number of US Navy flight test programs, develop and validate control laws, assist accident investigations, and provide an invaluable tool to the warfare simulation evaluators. MFS has provided aircraft simulations to many external users as well, including many universities, contractors, US government agencies, and foreign governments. XCASTLE has supplied piloted aircraft simulations during the Joint Theater Missile Defense (JTMD) exercises and the Joint Combat Search and Rescue (JCSAR) exercises, where a high-fidelity aerodynamic model was required. Several US Marine deployable flight trainers utilize the XCASTLE software, and it is being installed on several other military flight trainers. The XCASTLE software supports inter-player operability through the use of the High Level Architecture (HLA) protocol.

Conclusion

XCASTLE is a very powerful flexible simulation executive that contains over 10 years of requested capabilities and lessons learned. It is capable of running the many airframe simulations at Manned Flight Simulator, from simplistic linear models to the most complex aircraft. The ability to rapidly incorporate external airframe models has proven to be invaluable. Using the exact same code for piloted simulation sessions and at the desktop greatly reduces the development and life-cycle costs of airframe simulations at MFS. XCASTLE has proven to be easily adaptable to external operating environments. XCASTLE contains standard coding practices that enable it to operate on multiple platforms and operating systems. Its modular design allows an unlimited number of tools and applications to be integrated into XCASTLE. With this vast capability and proven technology at Manned Flight Simulator, XCASTLE is an excellent candidate for use as a standard simulation executive for Navy trainer simulations, and across branches of the Armed Forces.

References

[1] Nichols, J.H., "The Generic Simulation Executive At Manned Flight Simulator", AIAA-94-3429-CP, 1994.

[2] Burton, R.A., Miller, C.C., Mills, R.E., "Manned Flight Simulator and the Impact on Navy Weapons Systems Acquisition", AIAA-94-3420-CP, 1994.