

CONCURRENT DEVELOPMENT: THE C-130J STORY

**Jim Caylor, CAE USA Inc.
Military Simulation and Training
Tampa, Florida**

Abstract

In 1994, the UK Ministry of Defense ordered a suite of C-130J simulators for the Royal Air Force. The aircraft development effort was in the critical design phase – with First Flight still two years away. The aircraft design included a highly integrated, quad-redundant avionics architecture, with nine 1553 Busses linking six Bus Controllers and more than one hundred Remote Terminals. Nearly all the latest communication, navigation, surveillance, control and display systems were utilized. The suite of simulators was to include full flight, part task, and maintenance trainers. Producing these simulators concurrent with the aircraft development posed unique challenges for both the aircraft and simulator manufacturers. This paper describes those challenges, with primary emphasis on the concurrency issues surrounding the avionics systems development. It describes the decisions made, the logic behind those decisions, and lessons learned. This paper also discusses the need for industry guidance, and suggests a process for use on future concurrent development programs (new aircraft and major upgrades).

Biographical Sketch:

Jim Caylor is a Principal Systems Engineer with CAE USA Inc (formerly BAE SYSTEMS and Reflectone). He joined CAE in 1982, following five years as an Electronics Design Engineer with Smiths Industries Aerospace and Defense Systems. Jim was Lead Avionics Engineer on the C-130J simulator development program, and participates in the working group tasked with updating ARINC Report 610, Guidance for Use of Avionics Equipment and Software in Simulators. He is a member of the Institute of Navigation and International Council on Systems Engineering. Jim has a Bachelor of Engineering Technology degree from the University of South Florida and is a registered Professional Engineer.

CONCURRENT DEVELOPMENT: THE C-130J STORY

**Jim Caylor, CAE USA Inc.
Military Simulation and Training
Tampa, Florida**

INTRODUCTION

Concurrent development is the simultaneous design and deployment of a new aircraft and simulator. More precisely, the simulator must be fielded slightly ahead of the aircraft. Concurrent development poses unique challenges for both the aircraft and simulator manufacturers. Using the C-130J experience, this paper describes those challenges, with particular emphasis on the concurrency issues associated with the avionics systems simulation.

Scope

This paper focuses on the methods employed for the concurrent design of the C-130J simulator avionics, the logic behind those methods, and lessons learned. There are many other significant engineering issues associated with concurrent development, such as designing and certifying an aerodynamics model, but they are outside the scope of this paper. The avionics issues have broader relevance, in that they are also applicable to the major upgrades associated with airframe lifecycle extensions and Global Air Traffic Management (GATM). This paper also provides guidance for future concurrent development programs – both new aircraft and major upgrades.

This paper is intended to provide value to a wide audience, so background material is occasionally provided to clarify a point. Also, it is a “case study” in the sense that decisions were made in the context of the circumstances at the time. Applicability to other programs should be judged on a case by case basis.

Concurrent Development and Concurrency

While concurrent development specifically relates to new aircraft, “designed for concurrency” should be a goal for all new simulators (for both new and existing aircraft). Concurrency is a lifecycle objective, and is achieved when the simulator configuration is kept slightly ahead of the aircraft configuration. This allows aircrews and maintenance personnel to train on new systems before they experience them on the aircraft. Designing for concurrency means simulating the avionics systems such that future changes to the aircraft configuration and software can be incorporated into the

simulator with minimal time and resources. Concurrent development places the worst case demands on concurrency, in that there is a constant flow of changes during the design, integration, and initial deployment phases. Therefore, learning to accommodate these changes during the concurrent development inherently supports lifecycle concurrency objectives. In this context, this paper also provides guidance for general concurrency design decisions.

ARINC Report 610A

As noted throughout the paper, ARINC Report 610A, Guidance for Use of Avionics Equipment and Software in Simulators, played a significant role in the development process. The Report primarily targets avionics designers, providing them with guidance for incorporating simulator unique interfaces into their designs. It also provides background information on the unique situations encountered by black boxes when operating in a simulator. Note that there is no equivalent “military” document. Version A was released in 1994, and represents the consensus of airlines, airframe manufacturers, avionics equipment designers and simulator manufacturers.

Abbreviations

While this paper is not intended to be a program overview, reference to key participants is occasionally necessary. The terms “end-user”, “aircraft manufacturer” and “simulator manufacturer” are used when the point is applicable to all concurrent development programs. For C-130J specific points, the following abbreviations are used:

- MoD: UK Ministry of Defense (aircraft and simulator procurement agency).
- RAF: UK Royal Air Force (aircraft and simulator end-user).
- FSLO: Flight Simulator Liaison Officer (RAF liaison to the simulator manufacturer).
- LM Aero: Lockheed Martin Aeronautical Systems (aircraft manufacturer).
- HDAS: Honeywell Defense Avionics Systems (CNI Management System manufacturer)
- CAE: CAE USA Inc. (simulator manufacturer).

SIMULATOR DEVELOPMENT

A brief overview of the C-130J aircraft avionics architecture is presented here to set the stage for the simulator development discussion. Key to this overview are the avionics controllers managing the exchange of data on the MIL-STD-1553B Command/Response Multiplex Data Busses. First released in 1975, 1553 continues to dominate the serial communication needs on military aircraft.

Aircraft Avionics Overview

The C-130J avionics architecture (see Figure 1) includes nine primary 1553 busses: the Left and Right Avionics, Panel, Display and Comm/Nav Busses, and the Interprocessor Communications (IPC) Bus. Each of the nine busses is redundant (A and B twisted shielded pair). Note that the figure is a highly simplified block diagram, and doesn't do justice to the complexity of the architecture. The intent is just to orient the reader on the elements of the architecture that are relevant to the

Display Busses, and MC #2 controls the right side busses. Communication Navigation Identification - System Processor (CNI-SP) #1 is BC for the Left Comm/Nav Bus, and CNI-SP #2 is BC for the Right Comm/Nav Bus. MC #1 controls the IPC Bus. In each instance, the offside unit is the Backup Bus Controller (BBC) for each bus. The Bus Interface Units (BIUs), in addition to performing complex signal conversions, provide an additional layer of bus control redundancy.

Simulator Design Requirements

The MoD's simulator procurement included two Dynamic Mission Simulators (DMS), a Flight Training Device (FTD), and a Maintenance Part Task Trainer (MPTT) Suite. The procurement also included an Airloadmaster - Rear Cabin Trainer (ALM-RCT), Training Management Information System (TMIS), Computer Based Training (CBT), and a Schoolhouse. The DMS is equivalent to a Weapon Systems Trainer (WST), and the FTD is equivalent to a Cockpit Procedures Trainer (CPT). The FTD was designed to

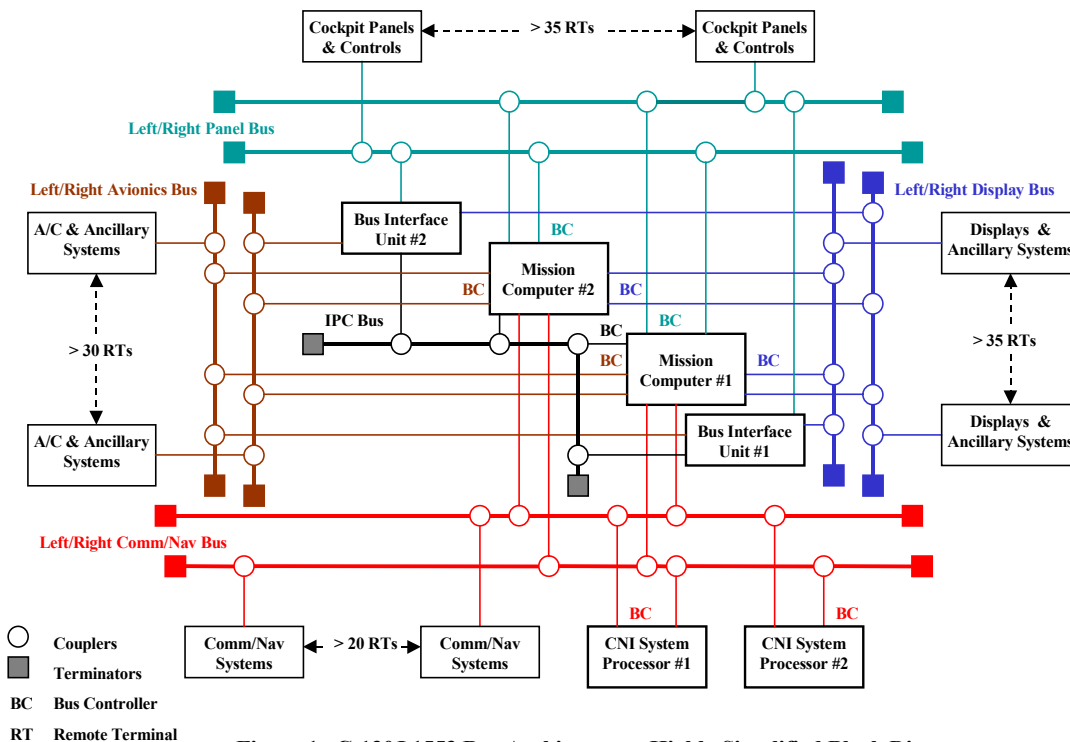


Figure 1. C-130J 1553 Bus Architecture - Highly Simplified Block Diagram

points made in this paper. The design includes many 1553 “smart” panels and Electronic Circuit Breaker Units (ECBUs). Mission Computer (MC) #1 is Bus Controller (BC) for the Left Avionics, Panel, and

FAA AC 120-45A Level 6 requirements and was essentially a DMS, less visual and motion. The MPTT Suite included Avionics, Propulsion and Auxiliary

Power Unit Part Task Trainers, and supported training using the aircraft Portable Maintenance Aid (PMA).

The concurrency specification requirement was “The software structure driving the DMS system shall be such that changes to the aircraft configuration and software shall be capable of being incorporated into the simulator with minimum time and resources”. It was also required that “Avionics simulations shall be compatible with the use of simulator functions such as freezes and resets”.

Following initial technical discussions, the requirements were enhanced to include “...modify the Operational Flight Programs (OFPs) to provide entry points (built-in hooks) in the C-130J Mission Computer OFP, and other OFPs, *as necessary* to provide ...freeze, slew, and reset capability, and to support the 32 simulator functions as defined in ARINC Report 610A.”

Requirements Analysis

Aircraft and simulator engineers met for the first time at the aircraft hardware Critical Design Review (CDR) in June 1994. As it turned out, this would be the first of more than 30 meetings to specifically address avionics concurrency issues. Ideally, this first meeting should have occurred much earlier in the aircraft program but, as is usually the case, the simulator supplier was not under contract. *Lesson 1: Find a way to get simulator engineers involved at the aircraft conceptual design stage.* It takes a long time for training requirements to flow down, and open competition held for the simulator contracts. The opportunity to influence the avionics software design to accommodate training features dissipates quickly – along with the opportunity to minimize cost and schedule impacts.

It was apparent from the CDR meeting that the first step would involve education. The aircraft avionics engineers needed to better understand the simulator environment. Since the engineers already had their hands full with the aircraft design effort, management would also have to be educated in order to squeeze the simulator needs into the priority scheme. *Lesson 2: A simulator advocate, physically located at the airframe manufacturer’s facility, is essential to the education process.* The advocate provides on-going information regarding the aircraft development, a communications link to the aircraft engineers, and leverage in dealing with the aircraft avionics vendors.

Education

The first step in the education process was to generate a paper titled “Issue Paper on Simulator Compatible Software for the C-130J/382J Program”. The term “Simulator Compatible Software” (SCS) was coined to distinguish it from “Simsoft”. Simsoft is an industry term generally accepted to mean special software resident in aircraft black boxes that supports their direct use in a simulator. Simsoft can be embedded in the flightworthy OFP, or in a modified OFP used only on the simulators. SCS has a broader meaning and refers to any software written to support the use of an aircraft black box in a simulator – irrespective of whether that software resides in the box itself, or in a simulator unique computer. Recall that one of the C-130J specification requirements was to provide “entry points (built-in hooks)”. *Lesson 3: Never refer to Simsoft, or SCS, as “hooks” or “entry points”.* This can lead to a false conclusion that the level of effort involved in modifying the OFPs is trivial, and therefore not a significant cost or schedule issue (and therefore a low priority).

Issue Paper

Following a brief program overview, the Issue Paper summarized the questions and factors involved in the decision to stimulate (use the aircraft black box in the simulator), or simulate (produce a software/hardware replication of the black box functionality). These factors are generally well known to simulator manufacturers, but are summarized here for the benefit of aircraft avionics engineers and end-users:

- **Buyer Requirements:** Does the buyer direct a specific approach?
- **Special Interfaces:** Would use of the aircraft black box require extraordinary signal conditioning?
- **Sensors and Emitters:** Does the avionics emit/receive RF energy, sense pressures, volumes, atmospheric conditions, or contain gyroscopes?
- **Data:** Is sufficient data available to simulate the black box functionality?
- **Development Costs:** Costs are always a factor.
- **Lifecycle Costs:** Usually more important than development costs.
- **Schedule:** Schedule is always a factor.
- **Fidelity:** What level of simulation fidelity is required?
- **Maintainability:** Compare the effort to maintain simulator unique hardware against that of maintaining the aircraft black box.
- **Availability:** Are sufficient black boxes available to equip the simulators?

- Life Expectancy: Is the black box scheduled to be phased out (e.g., replaced by a GATM system)?
- Concurrency: Which approach will best achieve the concurrency goals?

The Issue Paper then went on to provide an overview of ARINC Report 610A. 610A was the best available guidance document for the implementation of simulator compatible software. It defines 32 simulator unique functions (e.g., Flight Freeze, Weight Change, Snapshot Take, Speed Times N), and describes how they are used during training. For each avionics equipment type (Flight Management Computers, Cockpit Displays, etc.), 610A designated each of the 32 functions as requiring Full, Passive, or No Support.

The Issue Paper continued by defining the specific design and development tasks, and who had primary responsibility for each task (simulator, avionics, or aircraft manufacturer). The paper also provided a 1553 interface protocol that could support the exchange of data between the C-130J black boxes and the simulator host computer. The functions described in 610A were used to develop the protocol messages. Note that this was a sample protocol, mainly intended to highlight design considerations, and to serve as a starting point for the preliminary design effort. It could have been implemented as presented, but was primarily intended to start the thought process.

Since the design and development tasks defined in the paper were eventually performed, they are summarized here. The first task, primarily the responsibility of the simulator manufacturer, was to narrow the list of C-130J avionics to only those systems which would most likely be stimulated. The factors listed earlier were used to narrow the list. The next task, primarily the shared responsibility of the aircraft and simulator manufacturers, was to narrow the list even further to include only those systems requiring a more thorough analysis. The last task, primarily the responsibility of the avionics manufacturers, was to take the final list of black boxes and evaluate their performance in a simulator environment. The following 5 Step Process was used for this evaluation:

- Step 1: Using 610A, review the expected performance of the equipment for each Full and Passive Support Function.
- Step 2: What is the predicted equipment performance (without Simsoft) for each function?
- Step 3: If the predicted performance does not match the expected performance, is the predicted performance, nonetheless, acceptable?

- Step 4: If the predicted equipment performance does not match the expected performance, and the predicted performance is not acceptable, is there a way to force acceptable performance without Simsoft (e.g., use of simulator host computer software to manipulate the inputs and outputs of the equipment, or use of the simulator host computer to act as a control unit during repositions)?
- Step 5: If the answer to Step 4 is “no”, then what level of Simsoft is required (estimate lines of code, interface requirements, etc.)?

Two goals are apparent in the 5 Step Process: understanding the black box behavior in a simulator environment, and minimizing OFP changes (cost and schedule impacts).

For the C-130J, three systems remained on the final list and required application of the 5 Step Process: the MC, BIU, and CNI Management System (CNI-MS). The logic behind having the avionics manufacturer perform the final analysis is that they best understand the design and operation of their equipment. They are best equipped to predict the response of their box (OFP) to activation of one of the simulator functions. The simulator manufacturer can only assist with the analysis by helping them understand the expected performance. Note the use of the term “predict”. Given this was concurrent development, the MC, BIU and CNI-MS software had not yet been completed, so reaction to the simulator functions could not be fully verified in a lab environment.

CNI-MS Analysis

HDAS performed the CNI-MS analysis. The CNI-MS consists of two System Processors and three CNI Management Units (CNI-MUs). As a result of their analysis, 10 of the 32 simulator functions were judged to have an unacceptable impact on CNI-MS performance. The term “unacceptable”, in this context, meant that activation of the simulator function would cause the CNI-MS performance to deviate from normal behavior and potentially result in negative training, or that recovering from activation of the function would take an excessive amount of training time. HDAS generated a report summarizing the impact of each of the 10 functions. This summary formed the basis for further Technical Interface Meetings (TIMs).

MC and BIU Analysis

LM Aero performed the analysis for the MC and BIU. Since the BIU was BBC for the MC, and LM Aero was developing the software for both, the analysis of these

two systems was combined. Using the 5 Step Process, 9 of the 32 functions were judged to have an unacceptable impact on MC/BIU performance. They too summarized the results for further discussions.

Technical Interface Meetings

Following the initial analysis by HDAS and LM Aero, several TIMs were held with CAE to further analyze the findings. Since this paper is intended to describe a process, a detailed summary of the subsequent analysis is not presented here. A few examples are provided to show how the analysis proceeded. *(Note that the C-130J avionics functionality has evolved, and the equipment operation described here is as it existed at the time of the analyses. Also note that some details have been deleted in order to reduce the space required to make the point).*

CNI-MS Example #1

Problem: With Flight Freeze activated, there will be a wind buildup equivalent to current airspeed, and the internal navigation solution will be corrupted. If the aircraft is off the flight plan track, the CNI roll steering command will ramp up to its max value and, when released from Flight Freeze, result in a step command to the autopilot.

Discussion: When operating in the primary navigation modes, Kalman filtering is performed within the Embedded GPS/INS (EGI). The CNI-MS accepts the navigation solution inputs from the EGI. Since the EGI is simulated in software, the simulator host computer has full control over the inputs to the CNI-MS during normal operation. In degraded navigation modes, Kalman filtering is performed within the CNI-MS. Therefore, an uncontrolled wind buildup during Flight Freeze will only occur in degraded navigation modes.

The group requested a test be conducted on the HDAS engineering test bench. The results were that, when operating in degraded navigation modes, it took 3 minutes for the wind buildup to wash out following a 1-minute freeze.

On the C-130J simulators, the autopilot function is also simulated in software. The CNI generated roll steering command is fed to this simulation.

Conclusion: Since a wind buildup only occurs in degraded navigation modes, discuss the need to protect against this with the end-user. As a workaround, under simulator control, the navigation modes could potentially be switched during Flight Freeze to allow a

rapid recovery from navigation errors (degraded to primary, primary back to degraded).

The software simulated autopilot can smooth the roll steering command from the CNI when Flight Freeze is released - therefore no changes to the OFP are required.

CNI-MS Example #2

Problem: When Speed Times N is activated, violation of the vertical and lateral profile may occur.

Discussion: Speed Times N should only be used to accelerate the cruise phase of flight. The function should only be activated between Top of Climb and Top of Descent.

Conclusion: Place an advisory warning in the simulator instructor's handbook describing when not to use Speed Times N. No changes to the OFP are required.

CNI-MS Example #3

Problem: Internally stored mission data (flight plans, take off and landing data, performance initialization data, communication radio presets, etc.) are erased when the ZEROIZE function is executed by the student, and when FLIGHT COMPLETE is declared. FLIGHT COMPLETE is automatically declared when the aircraft lands at the primary destination and there is no path to an alternate destination.

Discussion: In order to conduct efficient training, a relatively quick method of reloading flight plans and initialization data is desired.

Conclusion: A mechanism to transfer flight plans and initialization data between the CNI-MS and simulator host computer is required. If necessary, modify the OFP to accomplish this.

MC Example #1

Problem: The MC OFP includes Secondary Warning Computer Software Configuration Items (CSCIs) for the Advisory Caution and Warning System (ACAWS) and Ground Collision Avoidance System (GCAS). These systems may generate aural and visual warnings when repositioning or slewing the simulator.

Discussion: The Inter-Communication System Central Switching Unit (ICS CSU) triggers the actual aural warnings. As this system is simulated using trainer unique hardware and software, the aural warnings can be suppressed at will. Some visual warnings (ACAWS) will appear when, for example, the simulator is

repositioned through GCAS algorithm trigger points, however they should disappear quickly once normal training is resumed.

Conclusion: No OFP changes are required.

MC Example #2

Problem: MC input parameters must be logically consistent. If true airspeed is slewed in the simulator, for example, then calibrated airspeed, groundspeed and Mach inputs must reflect the new values.

Discussion: The simulation software models must ensure data coherency. As regards parametric data, the MC is essentially a “throughput” device, and any adverse effects due to simulator functions can be eliminated, or minimized, by carefully controlling inputs to the MC. The MC algorithms generally check for min-max limits, but do not check the rate of change, so step inputs, within the min-max limits, should not be a problem.

Conclusion: No OFP changes are required.

Every issue associated with each of the identified functions (10 for the CNI-MS and 9 for the MC/BIU) was examined. More than fifty OFP performance issues had to be considered in the context of their impact on training effectiveness. Many meetings were required before agreement was reached on the implications of these issues. In some cases, the review was supplemented by empirical data gathered in LM Aero’s Large Aircraft Digital Avionics Simulation and Systems Integration Laboratory (LADASSIL) and HDAS’ engineering test facility. Once the team felt they sufficiently understood the impacts (given the performance of the developmental equipment was “predicted”), they decided to review the findings with the end-user (MoD and RAF). In preparation for the end-user review, the team decided to list the design options, summarize the pros and cons, make a consensus recommendation, and identify contingencies. In the case of the CNI-MS, the options were:

1. Modify the CNI-SP flightworthy OFP to resolve the issues raised during the analysis.
2. Modify the CNI-SP OFP to resolve the issues raised during the analysis, but create a “for simulator use only” version.
3. “Live with” all the limitations identified during the analysis.
4. “Live with” the minor limitations identified during the analysis, but implement “workarounds” for the major limitations.

Lesson 4: Avoid use of the term “live with”. While accurate, the term immediately puts the end-user on the defensive.

To support the end-user appraisal, LM Aero and HDAS estimated the cost and schedule consequences of Options 1 and 2, and CAE estimated the cost and schedule consequences of Option 4. Option 3 was dismissed early on. These estimates were necessary to evaluate the immediate impact on the program and, more importantly, to gauge the lifecycle consequences (OFP updates were occurring frequently during development, and estimated to occur at least every 6 months following initial deployment). Short and long term goals were fighting for top billing on the priority list.

As part of the on-going Integrated Product Team (IPT), the RAF FSLO had been briefed periodically during the equipment analysis phase. Involving the end-user in the thought process early on is indeed the best way to proceed. *Lesson 5: IPTs are a good thing.* If all the parties know all the facts, reason will prevail.

The RAF and MoD were thoroughly briefed on the results of the analyses, and it was nervously agreed that Option 4 was the best way forward. Options 1 and 2 were set aside as contingency plans.

Conservative Design Approach

Once Option 4 was selected, every effort was made to reduce risk. Since it was a concurrent development program and the summary of black box behavior in the simulator was a prediction, every precaution was taken, within the bounds of the selected approach. The following measures were taken to reduce risk, and address the few major limitations identified during the analysis:

- Maximize the performance of the simulator 1553 I/O System
- Connect the simulator 1553 I/O system to all busses
- Place an I/O controlled relay on all relevant 1553 bus connections
- Implement a hardware architecture that could support a software simulation if necessary
- Implement Piped Busses
- Simulate MC #2 and MU #3 during certain modes of simulator operation
- Educate the simulator project design team on black box operations so they can incorporate special software features in their designs (e.g., parameter smoothing)

The fundamental principle was that a simulator hardware/software design that maximized external control over the black boxes would be implemented.

A trade study was performed to select the highest capability 1553 printed circuit cards available on the market (cost was last on the priority list). Not only was the highest capability card selected – its capabilities limit was reached, and the vendor was called upon several times to enhance their design.

The simulator 1553 I/O system was connected to every bus, even if a specific need to trap data from a particular bus had not been identified.

Relays were placed in just about every stub on every bus. This not only supported malfunction insertion, but

malfunction) by opening the relays shown. An individual panel switch failure, however, cannot be simulated with this architecture. Only a total communications failure can be simulated. Lower level malfunctions were required for maintenance training, and to simulate in-flight equipment overheat faults. The Piped Bus approach supported placing the ABI System between the Comm/Nav/Bkr Panel and MC #2. The aircraft Panel Bus became two busses: the Piped Panel Bus and the Native Panel Bus. The ABI System simulates MC #2 on the Piped Bus, and the Comm/Nav/Bkr Panel on the Native Bus. During normal operation, 1553 messages were “piped” directly through the ABI System. If the instructor inserted a panel malfunction, the ABI System, under host computer control, would alter the contents of the 1553 message to MC #2 to reflect the failure. The iteration

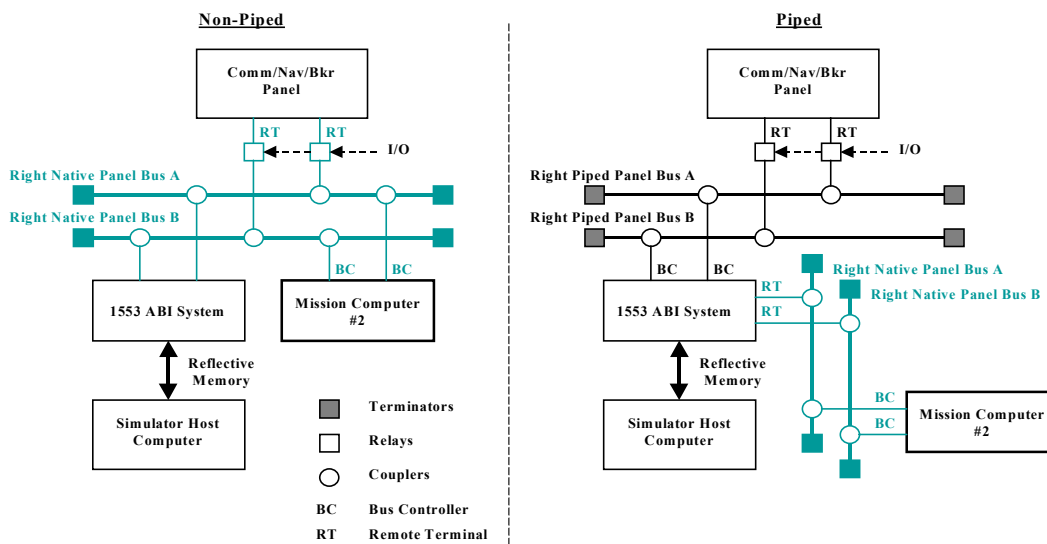


Figure 2. 1553 Piped Bus Architecture

supported switching in a software simulation should the need arise.

A “Piped Bus” architecture was implemented (see Figure 2). This scheme was primarily intended to support secondary malfunctions. In the example shown, without a Piped Bus, the aircraft Comm/Nav/Bkr Panel communicates directly with Mission Computer #2 on the Right Panel Bus. The simulator 1553 Avionics Bus Interface (ABI) System resides on the bus to monitor transfers, and simulate other equipment on the bus. Communications can be interrupted between the Panel and MC #2 (primary

rate of relevant traffic on the Piped Bus was increased within the ABI System in order to minimize latency. Note that the Display Bus was not piped, as it would introduce additional transport delays of critical flight data.

As mentioned earlier, the CNI-MS 610A analysis surfaced the need to store (download) flight plans and initialization data to the simulator host computer for subsequent upload back to the CNI-SP (see CNI-MS Example #3 above). Not having this capability was judged a major limitation. Additionally, the simulator host computer needed access to the information created

on the Mission Planning System (MPS). For the C-130J, missions are planned on the MPS and stored to a Removable Memory Module (RMM). The RMM is taken to the aircraft and inserted into the Dual Slot Data Transfer System (DSDTS). One of the CNI-MUs is then used by the crew to initiate the mission data transfer from the DSDTS to the CNI-SP. Since the DSDTS resides on the Display Bus and the CNI-SP resides on the Comm/Nav Bus, the data is routed via the MC. In addition to retrieving mission data from the RMM, the MC could also write mission data to the RMM.

Initially, two options for getting mission data into the host computer were identified: a) trap the data as it was transferred from the DSDTS to the MC on the Display Bus, or b) trap the data as it was transferred from the MC to the CNI-SP on the Comm/Nav Bus. When the analysis first began, a different data transfer system was planned for the C-130J. The DSDTS was introduced well into the simulator detailed design phase. It would be some time before the Mission Planning Interface Control Document (ICD) was finalized. This ICD would impact the software design of the MPS, DSDTS, MC and CNI-SP. This was considered a significant concurrency issue, so the design team exercised a third, more conservative option.

A partial simulation of MC #2 and CNI-MU #3 was developed (see Figure 3). Using the bus relays, the real units could be switched out, and the simulations switched in to transfer the mission data. The MU

software simulation, via the 1553 ABI System, generated pre-defined keystroke sequences to SP#2 in order to cause the transfers to occur. This technique is well known, and considered low risk. The challenge was simulating MC #2 such that SP #2 would consider it alive and well, and transfer the mission data (recall that the SP is the BC on the Comm/Nav Bus). It was also desirable to minimize, if not eliminate, any cockpit failure warnings during the switching process. It took a fair amount of empirical analysis, using the prototype simulator, to make this happen. This mechanism supported the transfer of mission data to the host computer, where it could be stored in volatile or non-volatile memory - and later transferred back to SP #2. Since there was no fully functional MPS and DSDTS initially, this mechanism also provided interim mission planning to support integrated system tests. Any number of missions could be constructed on the CNI-MU and downloaded to the host computer for future use. Although the SP had the capability to store flight plans in non-volatile memory, these were lost when ZEROIZE was activated, or when a new OFP was loaded (which occurred frequently early on). Later on, this mechanism also supported capturing crew changes to missions once they were transferred from the DSDTS to the SP.

Approach Summary

The conservative design approach outlined above ultimately satisfied the needs of the simulator development program. Dozens of OFP updates were installed in the black boxes during the

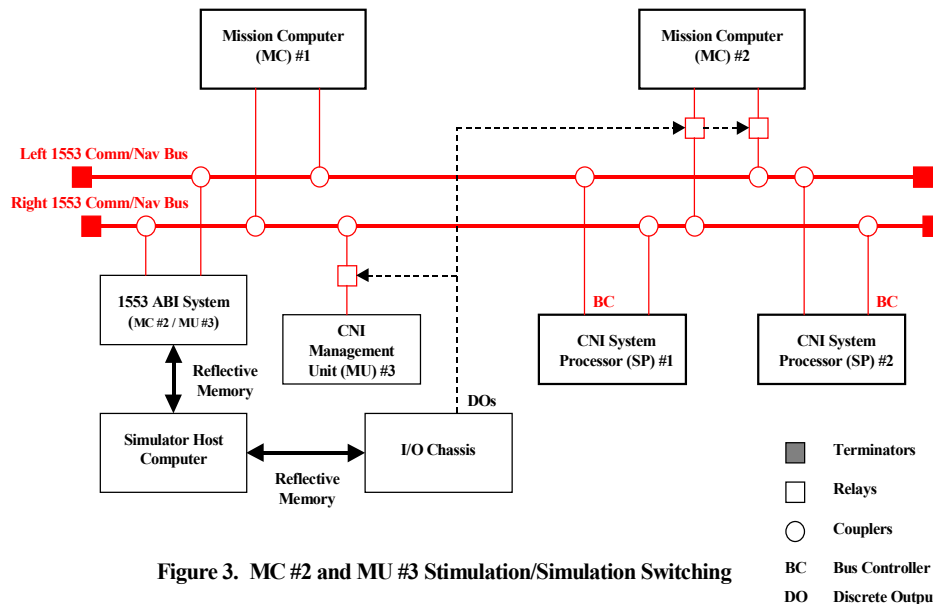


Figure 3. MC #2 and MU #3 Simulation/Simulation Switching

hardware/software integration phase, with little impact. An eleventh hour change in aircraft head down display vendors was also accommodated with minimal impact on the simulator development.

Inherent in this approach is the need for frequent contact between the airframe, avionics and simulator engineers. Well into the integration phase, LM Aero determined that the capability to transfer mission data from the MC to the DSDTS was not needed, and would be removed from the aircraft design. This capability was inhibited from cockpit selection, but the MC/SP protocol was retained in the SP for simulator use only. Without the on-going concurrency TIMs, this function would have been totally eliminated from the design, resulting in the loss of a significant training feature.

Lessons Learned

Five lessons learned were presented in the body of the paper as their relevance surfaced. Three additional lessons are presented here. These three relate to concurrency in a more general sense, and their importance is magnified for concurrent development programs.

Lesson 6: All avionics architectures, and related simulator development programs, are unique. There should be no predisposition to a particular concurrency design solution (simulate, stimulate, Rehost, Retarget). *Perform the analysis.* (Note that “Rehost” is use of unmodified avionics *executable code* running on a different computing platform in a simulated environment, and “Retarget” is use of avionics *source code* recompiled for a different computing platform in a simulated environment.)

Lesson 7: Concurrency is a lifecycle commitment. Concurrency is not something that can be achieved solely within the limits of the initial simulator delivery. Simulator (training) interests should be represented in the subsequent aircraft avionics upgrade process. As the avionics software matures, it should be re-examined in the context of operating in a simulator environment.

Concurrency is often viewed as simply a configuration control problem. “Put the simulator manufacturer (or training center) on the aircraft Engineering Change Proposal (ECP) distribution list, and everything will be OK”. While this step is essential, achieving concurrency requires much more. It normally takes too long for formal ECPs to flow down hill. Simulator engineering must somehow remain cognizant of the on-going aircraft avionics design developments. Simply attending aircraft design reviews can help substantially. This provides an opportunity for training interests to

influence avionics designs, and provides a better understanding of the equipment operation. *Maximizing the time available to determine simulator impacts, and make any necessary modifications, is the objective.*

Lesson 8: Guidance, communications, and education are essential to achieving concurrency. Each element of the military simulation and training community (airframe, avionics and simulator manufacturers, specification and procurement professionals, training center operators, and end-users) plays a key role in achieving concurrency objectives. If all those involved understand the issues driving concurrency, both short and long term goals can be achieved.

RECOMMENDATIONS

A process has evolved in the civil training community that could serve as a model for military simulation and training needs. It includes guidance, in the form of ARINC 610, and communications and education, in the form of an issues conference.

Guidance

When the C-130J simulator development effort began, ARINC 610A had just been released. It incorporated updates to cover new developments (such as Integrated Modular Avionics and On-Board Local Area Networks), and provided additional general guidance to the civil training community.

In March 2001, the joint Airlines Electronic Engineering Committee/Flight Simulator Engineering and Maintenance Conference (AEEC/FSEMC) Simulator Subcommittee met to review a strawman for AEEC Project Paper 610B. The intent is to simplify 610A by deleting unused functions and removing specific communication protocol details. Version B will also address alternate implementations (Rehost and Retarget), new avionics applications (e.g., Electronic Checklists), and lessons learned. The goal is to release the document by year-end.

ARINC 610 is a “guidance” document. No specific implementations are mandated. It does not require that the simulator functions be embedded within the avionics equipment software. The document states “This guidance report attempts to provide the appropriate level of understanding to allow these issues to be balanced one against the other and help the designers identify the *best possible compromise providing a maximum of functionality while maintaining the ownership costs at a minimum*”.

While there is a trend towards the use of civil CNI systems in military aircraft, training methods and priorities differ substantially. And, of course, civil aircraft are not equipped with “mission” avionics. Although the guidance contained in ARINC 610 is generally applicable to similar avionics systems, it specifically targets civil training methods. A similar document, written specifically to address military simulation and training needs, would serve to promote concurrency. The document would:

- Provide consensus opinions. Involving end-users in the guidance generation process, outside the budget and schedule constraints of a single program, would enhance the prospects of achieving concurrency.
- Provide general guidance for selecting an approach. Each situation is unique, and may call for a different solution. For example, one would probably not invest a vast amount of money to simulate (versus stimulate) a system that is due to be replaced in a couple years, even though most other factors may favor the simulation approach.
- Provide objective performance measures. Once the reader is “guided” to a particular approach, the document should include specific recommendations. One of the objectives of 610B is to simplify 610A in order to obtain broader acceptance, and answer the question “what does compliance to 610 mean?” The C-130 Avionics Modernization Program (AMP), for example, requires 610A compliance. What does that mean? Is the same simulation approach required for each equipment type? Should all 32 simulator functions be supported, even if some have no impact on the operation of a particular black box? When will this determination be made, and how will it be costed? Can it be achieved within the delivery schedule? Have the simulator manufacturers weighed in on the consequences? Are they participating in the training analysis? Are they in the loop on the aircraft avionics design? *(Although some of these follow-on questions carry a negative connotation, simulator manufacturers were thrilled to see 610 as an AMP requirement. The ability to achieve lifecycle concurrency will be greatly enhanced.)*

Communications and Education

An organization similar in purpose and structure to the Flight Simulator Engineering & Maintenance Conference (FSEMC) would provide a communications medium for military concurrency issues. The FSEMC’s mission is to provide a *neutral* forum for the discussion

and resolution of flight simulation, engineering, and maintenance issues. While there is some military training participation, the group primarily serves civil aviation needs. They meet annually, and have a highly successful track record of reducing costs and solving common engineering problems. They are a relatively new organization, meeting for the 7th time this year. Costs are borne by the individual participants.

The USAF sponsored GATM Users’ Conference is also a good model for a concurrency forum. It has broad participation, and is primarily intended to “educate”.

Just as ARINC facilitates the FSEMC, a sponsor is needed to bring military users and suppliers together to discuss avionics concurrency issues. Following is a list of potential sponsors:

- Interservice/Industry Training, Simulation and Education Conference
- Institute for Simulation and Training
- National Center for Simulation
- Aeronautical Systems Center Training Systems Product Group
- Naval Air Warfare Center Training Systems Division
- U.S. Army Simulation, Training and Instrumentation Command

Once the organization takes shape, generating a concurrency guidance document would be one of the primary objectives.

CONCLUSION

Concurrency is a significant military simulation and training issue. Concurrent development brings concurrency into the present tense, in that frequent black box OFP changes are inevitable in the short term. In addition to the new aircraft currently being developed, avionics modernization programs and GATM driven system upgrades make concurrent development a real challenge. Achieving concurrency requires the participation and understanding of all parties involved in the development and use of simulators. It requires a long-term view, which is often lost in the cost and schedule constraints of a simulator development program.

Guidance, communications and education are essential to achieving concurrency. Hopefully, the C-130J case study has contributed to the education process by highlighting some of the challenges involved with concurrent development.