Open Systems Ada Technology (OSAT) Program

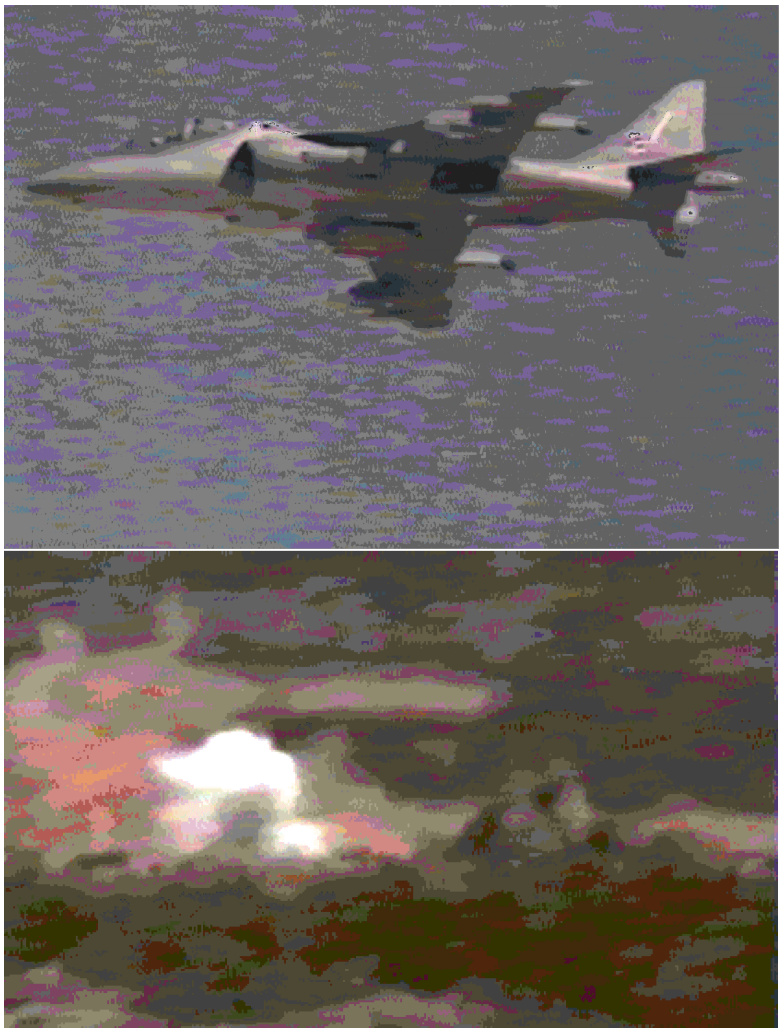
Don Winter

Boeing - Phantom Works

Open Systems Ada Technology Program

OSAT

- Dual emphasis - Ada95 and POSIX
  - Proved mixed language support attributes of software architecture (Ada95, C, C++)
  - First flight application of Ada95
  - Utilized POSIX features of VxWorks, collected metrics
  - First live demo of Common OFP 30-step ballistics integrator
  - First flight of Computing Devices International (CDInt) PowerPC mission computer
  - Accuracy was <u>not</u> an explicit test objective, but scored 6/6 hits
  - Pilot feedback very favorable



Project Objectives

OSAT

- Convert the mission computer of an AV-8B to a COTS, open standards-based platform
  - PowerPC 604 Processor
  - Wind River VxWorks POSIX-compliant RTOS
  - Boeing's Common Operational Flight Program (COFP)
  - Ada95 (AV-8B compatible) F-15 Ballistics Algorithm
- Develop / demonstrate an HOL OFP
  - Basic Navigation, Communications and HUD display functions (C++ from Common OFP)
  - A / G Ballistics and Stores Management functions (new Ada95)
    - Continuously Computed Impact Point (CCIP) calculation
    - Release of Mk-76 Bombs

Objectives - Continued

- Compare the observed CEP from this demonstration with the AV-8B Fleet OFP CEP
- Integrate the Data Fusion Integrity Process (DFIP) Algorithm into the AV-8B OFP
  - Test DFIP functionality in the AV-8B Flight Simulator
  - Report results in Final Report
- Collect and report lessons learned:
  - POSIX
  - Ada95, mixed language OFPs
  - DFIP

Project Participants

- Sponsors
  - Ada Joint Program Office: Demo flight application of Ada95
  - Open Systems Joint Technology Force: Demo COTS, POSIX
  - JSF Program Office: Avionics risk reduction
  - Wright Laboratory: Demo of DFIP, reuse adapter
- Contractors
  - Boeing / McDonnell: System analysis, development and test
  - Computing Devices International: COTS MC, support S / W
  - Green Hills Software: Ada95 / C++ Development Tools
  - Wind River: VxWorks Real Time Operating System
- Project Management and Technical Evaluation
  - NAWC-WD, China Lake: Aircraft integration and flight test

Flight Test Results

OSAT

- Flight Test Data
  - Weapons Delivery Flight (20 March 1997, Baker Range)
    - 6 X MK-76 all South-North runs
    - 3 X MK76 @ 10 Degree Dive
      - First @ xx Feet
      - Second @ xx Feet
      - Third @ xx Feet
    - 3 X MK-76 @ 45 Degree Dive
      - First @ xx Feet
      - Second @ xx Feet
      - Third @ xx Feet
- Data Evaluation
  - Based on limited number of releases, bomb impacts were as good as current fleet AV-8B Night Attack software

Open Systems Components

OSAT

- Computing Devices International Mission Computer
  - Single card has PowerPC 604e Processor, program memory, two 1553 channels, Ethernet, RS232, and discrete I/O
  - Sun Laptop used as support computer - OFP compile, reload
- Baseline C/C++ MC OFP
  - Microsoft Visual C++ Desktop Development
- C++ Executive, POSIX-compliant
- Green Hills MULTI Ada95 and C++ Tool Set
  - Mixed language OFP linking, loading, and debugging
- Wind River VxWorks RTOS

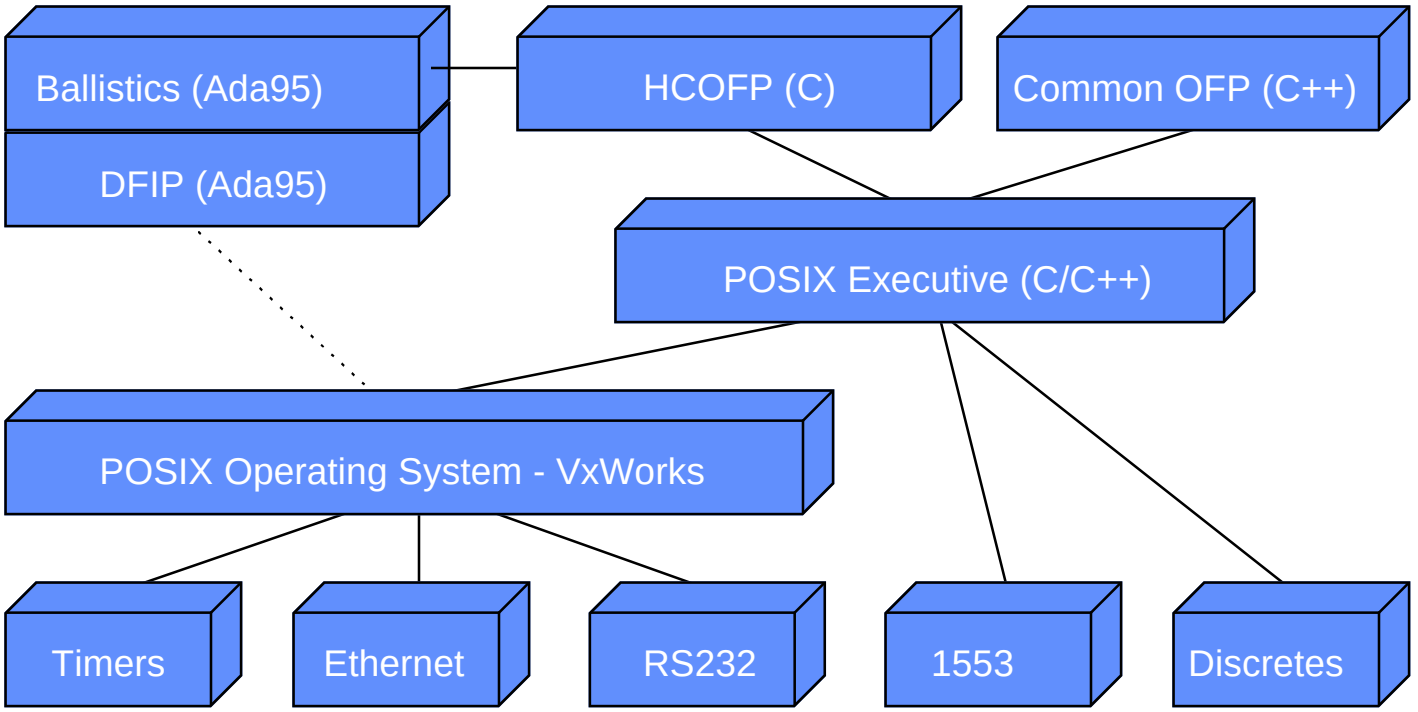
Gaining experience with commercial tools and POSIX API contributes to the maturation process of open systems avionics

OFP Configuration

- Rehosted “C” OFP (Common OFP) from AV-8B Flight Simulator
  - AV-8B Night Attack functionality
- VxWorks RTOS With POSIX
- C++ Executive utilizing VxWorks POSIX calls
- COFP C++ Navigation components
  - Same as used in F-15 and F/A-18 flight demonstrations
- AV-8B C++ Communication components
- Re-engineered F-15 Ada95 Ballistic Integrator
- Ada95 DFIP Algorithm

OFP Components

OSAT



POSIX Usage

- Message queues for communication between interrupt service routines and rate group tasks
- Semaphores in bus controller services to protect simultaneous access of scheduled I/O chain linked list
- Timers and synchronous real-time signals in tasks to perform scheduling of I/O
- Retained VxWorks native specific calls
  - Tasking
  - Interrupts
  - System set-up

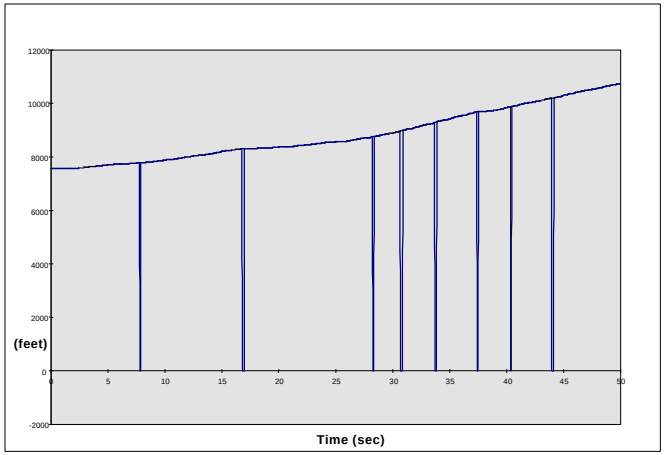
POSIX Lessons Learned

- Execution times of POSIX and VxWorks features are similar
- POSIX features were easy to employ and intermingle with native features within VxWorks
- VxWorks POSIX is not complete; it doesn't support POSIX threads
- For future projects, recommend that POSIX options be used wherever possible
  - Utilize any individual native OS calls where needed for additional functionality or increased efficiency.

Data Fusion Integrity Process

OSAT

- Wright Lab's / TASC DFIP algorithm provides detection, limiting and recovery from intermittent data errors
- Ada95 DFIP filter was applied to four Ballistics input data channels and the Weapon Range output
  - Filter can be used to stabilize CCIP solution
- Typical results when applied to Weapon Range output, given input data drop-outs:



DFIP Evaluation

- Performance was tested in AV-8B Flight Simulator
  - Short-duration (induced) data drop-outs were managed
  - Longer-term drop-outs and highly dynamic valid data would require a compromise design
  - Matrix style filters are expensive with respect to memory and execution time
    - Execution time for five channels was approximately 1 MSec
- Current algorithm requires further refinement to add value to Boeing's ballistics applications
  - Short-term drop-outs not seen in simulators, rarely in flight
  - Other protection methods are already in place in fleet OFPs

Ballistic Algorithm Design

OSAT

China Lake's Runge-Kutta Ballistic Integrator → Boeing's F-15 Precision Improvements Conversion to Ada83 → OSAT's Conversion to Ada95 On Fast COTS CPU, POSIX RTOS → Risk Reduction for Current/Future Programs

- Re-engineered F-15 Runge-Kutta Ballistics Algorithm (BA) from Ada83 to Object Oriented Ada95
  - Employed Rational Rose design tool and OO methodology
  - Higher performance processor allowed improvements to the accuracy of the Ballistics solution over AV-8B
    - Position differential equation solved
      - Velocity differential equation solved
      - Throughput available to run 30 steps rather than 10, 3D rather than 2D
      - Trajectory completed in one frame at 20 Hz
      - Step size picked every trajectory step
    - Last step adjusted to complete trajectory at target elevation

Ada95 Feature Usage

- Tagged types - including extension of tagged type
- Abstract types and functions
- Aliased types
- Access-to-constant types
- Reused legacy Ada83 generics for vector operators

Ada 95 Annex Feature Usage

OSAT

- Annex A - Predefined Language Environment (Numerics)
  - package Ada.Numerics.Long_Elementary_Functions
- Annex B - Interfaces to Other Languages
  - pragma Import and Export
  - package Interfaces.C
- Annex C - Systems Programming
  - pragma Preelaborate
  - Machine Code Insertion - used in Timing builds only
- Ada not the Main Program
  - Ada95 components were called from a C++ main program

OO/Ada95 Evaluation

OSAT

- Encountered very smooth language transition for experienced Ada83 engineers
  - New object oriented features are a natural extension to the language
- Learning OO design methodology can be difficult, especially for structured top-down programmers
- Good training leads to success - Designers attended AJPO's *Transitioning to Ada95* course (Ada95 for Ada83 Programmers & Embedded / Real-Time Programming)

OO/Ada95 Evaluation - Continued

- Features for mixed language support were easy to implement
  - Interfacing to C software was simple with the new Ada95 features
  - Interfacing to C++ was more difficult since C++ is not standardized, and so no package Interfaces.Cpp exists yet.
- Ada95 is very portable
  - OSAT OFP Ada95 components were run on Sun Workstation (Rational), PC / Pentium Workstation (Object Ada), Motorola PowerPC Card (Green Hills), DY-4 PowerPC Card, and the CDInt MC PowerPC Card
  - Conversion of Ada83 software to use Ada95 compiler (without re-engineering) was relatively simple
  - Changes were isolated to low-level design areas such as processor-dependent data formats

Conclusions

OSAT demonstrated Ada95 and OO methodology in a flight-worthy avionics application

The demonstration included an application of POSIX with a COTS real-time operating systems

Multi-language OFP components were combined and reused, demonstrating the capability of COTS tools, OO architecture and wrappers (adapters)

The DFIP Algorithm was implemented and evaluated in a flight-worthy application

A commercial processor and board support package was flown in an avionics Mission Computer

The performance of the prototype MC and software in flight test was equivalent to AV-8B fleet performance