Secure, Real-Time CORBA
Requirements for Military Avionics

Presented to OMG/NSA Workshop
April 1997

Roberta Gotfried
(310) 334-7655
rgotfried@msmail4.hac.com

Dennis Finn
(310) 334-1043
dfinn@msmail4.hac.com
Outline

• Characteristics of Military Avionics Processing Environments
• Software Architecture Issues in Military Avionics Systems
• Real-Time Requirements
  – RT CORBA Functional Requirements
  – Real-Time Features of Avionics Operating Systems, POSIX and Ada95
  – Which Real-Time Requirements Implemented in the Application, OMG’s OMA, OS, Hardware?
• Evolution of Avionics Processing Architectures
• Security Requirements
  – Information Security is a Recognized Requirement in Airborne Systems
  – Security Features of F-22 & Future Military Avionics Systems
  – Which Security Requirements Implemented in the Application, OMG’s OMA, OS, Hardware?
• Technical Risk Reduction Plan for CORBA in Military Avionics
Characteristics of Military Avionics Processing Environments

• Real-Time: Periodic & Aperiodic Events; Hard Real-Time; Resource Management - QoS
• Processing: Serial & Parallel; Signal & Data
• Parallel Processing: Cache Coherent Shared Mem versus Message Passing Distributed Mem (e.g., Mercury)
• I/O: Multiple Buses; Not Typically TCP/IP; Streaming Data
• Adaptive Behavior: Increase or Decrease Processing Load in Response to Dynamic Environment (e.g., sensor resolution, EW, Fire Control, Radar Modes, ...)
• Security: Military & Intelligence Threats; Multi-Level; International
• Mission Critical: Lives Depend on Correct Operation (BIT, Fault Management, System Integrity)
• Embedded: Remote Operations; Field Replaceable Modules; Size, Weight and Power: 2X Increase => 10X $ Increase
Example Avionics Processing Architecture

This architecture is taken from the Joint Advanced Strike Technology Program Avionics Architecture Definition, Version 1.0 dated 9 August 1994.
Software Architecture: Issues in Military Avionics Systems

• Evolution (Evolvability)

• Increased Situational Awareness
  – Increased Survivability and Lethality

• Aircraft LifeCycle Cost
  – Development
  – Maintenance
  – Upgrades (technology, function, cost reduction)

• Scalability at Runtime

CORBA represents part of a solution to address many of these challenges.
System Evolvability
20 - 30 Year LifeCycle

• Why Upgrade: Parts Obsolescence; Changes in Functionality & Performance
• Cost-Effective Upgrades
  – Reengineer Legacy S/W, OO, Reuse, COTS
  – Revalidation strategies for cost, reliability, correctness (flight test)
Increased Situational Awareness
(Survivability & Lethality)
Decreasing Aircraft Life Cycle Costs

- API Standards Increase Portability
- OO Software Architectures Increase System Modularity
- CORBA Increases Portability of Objects & Interoperability Between Objects
- Increased Potential for Reuse and for Use of COTS Components Lowers Development and Incremental Upgrade Costs
- Software: Jovial, Ada83, other --> Jovial, Ada95, COTS, Legacy Reuse, other
- Increased Use of COTS Standards: Portability, Interoperability, Scalability
- Increased Use of COTS Hardware & Software Components
- Fewer Hardware Module Types
Run-Time Scalability
Real-Time CORBA
Required In Military Avionics

• All Real-Time SIG (ORBOS) Activities Necessary in Military Avionics
  – Fault Tolerance WG
  – Flexible Bindings WG
  – Embedded ORB WG
  – Multiple Protocols WG (low latency transport, RT IOP, UDP GIOP, ...)
  – Time Services WG
  – End-to-end Timeliness Predictability WG
  – Scheduling WG
  – Run Time Performance Metrics WG (Metrics SIG - initial RFI real-time market)

• Real-Time Parallel Processing for CORBA Needed in Military Avionics
  – Parallel ORB Supporting SPMD Applications on MIMD Parallel Processor
  – No OMG SIG/WG on Parallel Processing Platform
  – Tandem Has Parallel ORB for Fault Tolerance on Proprietary Non-Stop Processor
  – MPI DeFacto Standard in HPCC Community - RT MPI as RT SIG RFI Response
  – DARPA HPC++
Real-Time OS + CORBA + Security in Military Avionics

• JSF, DISA (AJPO), and USAF Wright Lab funded Hughes to evaluate and determine the suitability of the POSIX and AOS APIs, and Ada 95 features for real-time embedded software
  – Areas of Interest: availability, performance, security, and supportability tradeoffs
  – Delta Document Comparing RT POSIX (IEEE 1003.5b/D5), AOS, Ada 95
    • 165 page Delta Doc on OMG Server: orbos/97-03-02, orbos/97-03-03
• Examining CORBA + Security Implications for AOS/POSIX/Ada95 in Military Avionics
## SAE Requirements

<table>
<thead>
<tr>
<th>Category</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Synchronization</strong></td>
<td><img src="image" alt="Synchronization" /></td>
<td>Timekeeping and task coordination</td>
</tr>
<tr>
<td><strong>Data Security</strong></td>
<td><img src="image" alt="Data Security" /></td>
<td>Information protection and authentication</td>
</tr>
<tr>
<td><strong>Timer Services</strong></td>
<td><img src="image" alt="Timer Services" /></td>
<td>Timing functions and delays management</td>
</tr>
<tr>
<td><strong>Special Devices</strong></td>
<td><img src="image" alt="Special Devices" /></td>
<td>Device management and configuration</td>
</tr>
<tr>
<td><strong>Non-Operational Support</strong></td>
<td><img src="image" alt="Non-Operational Support" /></td>
<td>Bootup/Initialization/Shutdown</td>
</tr>
<tr>
<td><strong>Program Support</strong></td>
<td><img src="image" alt="Program Support" /></td>
<td>Program management and execution</td>
</tr>
<tr>
<td><strong>Memory Management</strong></td>
<td><img src="image" alt="Memory Management" /></td>
<td>Memory allocation and manipulation</td>
</tr>
<tr>
<td><strong>File Management</strong></td>
<td><img src="image" alt="File Management" /></td>
<td>File handling, creation, and deletion</td>
</tr>
<tr>
<td><strong>Data Conversion</strong></td>
<td><img src="image" alt="Data Conversion" /></td>
<td>Data transformation and encoding</td>
</tr>
<tr>
<td><strong>Built-In Test</strong></td>
<td><img src="image" alt="Built-In Test" /></td>
<td>Diagnostic and test capabilities</td>
</tr>
<tr>
<td><strong>Bootup/Initialization/Shutdown</strong></td>
<td><img src="image" alt="Bootup/Initialization/Shutdown" /></td>
<td>System startup and shutdown operations</td>
</tr>
<tr>
<td><strong>Task Control</strong></td>
<td><img src="image" alt="Task Control" /></td>
<td>Task scheduling and prioritization</td>
</tr>
<tr>
<td><strong>Communication</strong></td>
<td><img src="image" alt="Communication" /></td>
<td>Data transmission and network protocols</td>
</tr>
<tr>
<td><strong>Input / Output</strong></td>
<td><img src="image" alt="Input / Output" /></td>
<td>Input/output handling and device control</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td><img src="image" alt="Configuration" /></td>
<td>System configuration and settings</td>
</tr>
<tr>
<td><strong>Instrumentation</strong></td>
<td><img src="image" alt="Instrumentation" /></td>
<td>Measurement, testing, and monitoring</td>
</tr>
<tr>
<td><strong>Reinitialization</strong></td>
<td><img src="image" alt="Reinitialization" /></td>
<td>System reset, recovery, and restart</td>
</tr>
</tbody>
</table>
Real-Time POSIX Should Address

**Requirements:**
- Program Support
- Data Security
- Memory Management
- Input Output
- Data Conversion
- Fault Management
- Non-Operational Support

**Number of Requirements:**
- 108 Total Requirements

**Findings:**
- Significant POSIX Deficiencies were Found in:
  - Program Support
  - Data Security
  - Memory Management
  - Input Output
  - Data Conversion
  - Fault Management
  - Non-Operational Support

**Recommendation:**
- Present The Missing Requirements to The Real-Time Working Group.
- Get a Consensus on The Needed Requirements.
- Implement The Agreed-on Requirements.
- Migrate Any Requirements That have not Been Agreed-on to Category 4.
- Recommend The Implementation of Ada Bindings of Any Relevant Requirements.
The Trend in APIs

**Past**
- Real-Time Functionality Lacking in OS, POSIX, and Ada
- Considerable Overlap in OS, POSIX, and Ada

**Future**
- High Order Functionality in Ada
- General OS Functionality in POSIX
- Hardware Specific Functionality in RTOS

**Present**
- Ada + POSIX
Evolution of Avionics Processing Architectures

Federated System Properties:
- Single Application Within Each Physical Boundary
- Single Applications Developer Per Unit
- Debugging Scope Is Limited to Application

Integrated Avionics Properties:
- Multiple Applications Sharing Many Common Resources
- Multiple Applications Developers
- Multiple Applications Debugging

Data Security Importance:
- Protect Classified Information From Leaking

Data Security Approach:
1. Each Unit At Application High
2. "Natural" Red/Black Separations

Software for:
- Radar
- Comm/Nav
- IRST
- Mission Mgmt
- EC
- Pilot Interface

Data Security Importance:
- Prevent Illicit Interactions Between Applications

Data Security Approach:
1. "Built-In" Robust Hardware and Software Separation Mechanisms: Trusted Computing Base (TCB)
2. Assurance Through Trust Engineering Disciplines
Air Vehicle Interfaces Extend Beyond the Operational Environment

The IWSDB Data Dictionary logically integrates Data Repositories which may be physically located anywhere on the network.

Manage

- Schedule and Performance Monitoring
- Cost Accounting
- Configuration Management
- CDRL Tracking

Test and Evaluate

- Test and Evaluate
- Instrumentation
- Diagnostics
- Mission Support
- Data Processing

Operate and Support

- Operate and Support
- Integrity Monitoring
- Flight & Environ. Data
- BIT, IOBD Results
- Diagnostics
- Existing Systems
- Mission Support
- Training

Design

- Design
- Requirements Definition
- Design Modeling and Analysis

Manufacture

- Manufacture
- NC Equip. Fabrication
- Resource Planning and Control

Develop Logistics

- Develop Logistics
- LSA
- Training Development

Analyze Problems

- Analyze Problems
- Design Analysis
- Integrity and R&M Analysis
- Manufacturing Analysis
- Logistics Analysis

Copyright © 1997 by Hughes Aircraft Co. Permission to make digital or hard copies of part or all of this work is granted without fee provided the copies are not made or distributed for profit or commercial advantage and that the copies bear this notice. Hughes Aircraft Co. makes no warranty of any kind with regard to this material and shall not be liable for errors contained herein.
Information Security is a Recognized Requirement in Airborne Systems

Off-Board Information
- National Assets
  - COMINT
  - ELINT
  - IMINT
- Threat Assets
  - HUMINT
  - Surveillance Information

Multi Level Security
- SAP/SAR
- SCI
- Codeword
- NOFORN
- NATO

Top Secret
Secret
Confidential

On-Board Information
- Mission Plan
- Threat/Target Information
- Aircraft Capabilities and Technology
- Databases
- Electronic Keys

Example Security Threats in Airborne Systems
- Insider Threat (developers, maintainers)
- Disclosure
- Eavesdropping
- Penetration
- Traffic Analysis
- Masquerading (Spoofing, Malicious Logic)
- Emissions Attack
- Reverse Engineering (Tech/Alg)
- Penetration (Maintenance)
- Falsification
- Obstruction (Overload)

Applications
- F-22
- Joint Strike Fighter
- Upgrades to Existing
  - RECCE
  - JSTARS
  - E2C
  - F15
  - Comanche
- Data Fusion
- Sensor Fusion
- Situation Awareness
- RealTime Intell
- Integrated Avionics
- Off-Board Sensors
- SATCOM
Technical Risk Reduction Plan
for CORBA in Military Avionics

• Real-Time, Secure CORBA
  – Performance Assessment of COTS ORBs (execution time & memory usage)
  – Real-Time, Trusted ORB Supporting MLS Using Standard RTOS API (e.g., AOS)
• Increased Experience Using CORBA With Ada95 on Real-Time, Embedded COTS Processor (e.g., OIS/Iona Orbix/Ada on PPC)
• Profiles of COTS ORBs - Use Only The Necessary Functionality
• Extensible ORBs (e.g., I/O)
• Parallel, Real-Time, Secure CORBA Applications
  – DeFacto Parallel Processing API Standards (i.e., MPI, Embedded MPI, Real-Time MPI) for Scalability
  – Real-Time, Secure OS Experience in COTS Parallel Processors (e.g., DARPA PROSE for Intel TeraFlops)
  – Secure, RT CORBA for SPMD Applications on COTS Embedded Parallel Processors (e.g., Mercury, CSPI, Sky)
• Demonstrate Scalable, Real-Time, Secure Military Application Software Using CORBA on Embedded Processors
Summary

• CORBA Provides Same Benefits to Commercial and Military Systems
  – Standard APIs Increase Application Portability
  – Heterogeneous Languages, COTS Components, Reuse
  – Interoperability Between Distributed Objects

• Military Avionics Systems Require Solutions That Address Combinations of
  – Security + Real-Time + Embedded + Fault Tolerance + Scalability

• CORBA Needs to Provide
  – Flexibility in Security Policy and Models
  – Well-Defined and Acceptable Levels of Assurance in ORBs
  – Security Architecture That Clearly Defines OS/ORB Roles