ARINC 653

An Avionics Standard for Safe, Partitioned Systems
Agenda

• Aerospace Trends
• IMA vs. Federated
• ARINC 653
  – Main concepts
  – Safety facilities
  – Example ARINC 653 Implementation
• Configuration and Certification of an ARINC 653 System
• Q&A
Aerospace Trends
Main Aerospace & Defense Trends

**Aerospace**

- More functionality – smarter avionics, more passenger systems, more payload
- “All electric” aircraft (more computer-based systems)
- Global procurement/partnerships
- Safe and secure
- Pressure on development cost, schedule
- Pressure on operational cost (personnel, training, spares)

**Defense**

- More functionality – more lethality/survivability, integrated battlefield, more arms and armor
- Cyber warfare (more computer-based systems)
- Coalitions/interoperation
- Secure and safe
- Pressure on development cost, schedule
- Pressure on operational cost (personnel, training, spares)
System Implications

More functions, “systems of systems,” more connectivity in less space, weight, and power (SWaP), reduced cabling

- Hardware consolidation (multiple applications on fewer processors)
- Software “pressure”: larger volume of software comingleed on fewer processors
- New challenges to safe and secure
What is a Certified System?

- The FAA certifies *aircraft, engines and propellers*
- Components are certified only as part of an airplane or engine
- Safety Case flows down from “aircraft” to “systems” to “software”

“aircraft”

“system”

“software”
Federated vs. IMA

- Engine Controls
- Engine Monitoring
- Fire Control
- Flight Management
- Inertial Reference System
- Displays
- Navigation Computer
- Mission Computer
- Flight Controls
- Weapons Control
- Stores Management
- Targeting Computer
- Radar
- Sensor systems
- FLIR
- EO/OP
Federated vs. IMA

Flight Controls
Flight Management
Inertial Reference System

Displays
Navigation Computer
Mission Computer

Engine Controls
Engine Monitoring
Fire Control

Weapons Control
Stores Management
Targeting Computer

Radar
Sensor systems
FLIR
EO/OP
Federated vs. IMA

Flight Controls
Flight Management
Inertial Reference System

Displays
Navigation Computer
Mission Computer

Engine Controls
Engine Monitoring
Fire Control

Weapons Control
Stores Management
Targeting Computer

Radar
Sensor systems
FLIR
EO/OP
Federated vs. IMA?

**Federated**

**PROs**
- Traditional methodology (well understood)
- Relative “ease” of design and certification
- Supply chain geared for this

**CONs**
- SWaP – Each function is separate LRU
- Poor S/W re-use
- Poor portability
- Poor modularity
- Tier 1 at mercy of Primes ($$ for Tier 1)

**IMA**

**PROs**
- SWaP (multiple functions on single LRU)
- Excellent S/W re-use
- Excellent portability
- Excellent modularity
- Changes require *minimal* re-certification $$$$

**CONs**
- “Modern” methodology (777, A380, 787…)
- Poorly understood
- Complexity of design and certification
- Supply chain not setup for IMA projects

Changes require *complete* re-certification $$$$
Federated vs. IMA – “Reality”

- These will co-exist for some time
  - Some functions still preferred on single LRU – Flight Controls for instance
- ARINC 653 Standard evolving to include “federated” LRUs
  - ARINC 653 Part 4

Boeing KC767A Tanker Aircraft
[Certified Apr 2008]

IMA
- Avionics & Flight Computer
- Aerial Refueling Control Computer

Federated
- Multi Purpose Control Unit
- Hose Deploy
The ARINC 653 Standard

Multiple Partitions
Multiple Criticality Levels
The ARINC 653 standard

• **Supplement 1 <Jan 1997> - AEEC, Boeing**
  – Health Management, APeX services
  – Time and Space Partitioning
• **Supplement 2 <Mar 2006>**
  – Part 1 <Mar 2006> – Required Services, including changes to:
    • ARINC 653 partition management
    • Cold start and warm start definition
    • Application software error handling
    • ARINC 653 compliance
    • Ada and C language bindings
  – Part 2 <Jan 2007> - Extended Services, including File System, Logbook, Service Access points...
  – Part 3 <Oct 2006> - Conformity Test Specification
• **Supplement 3 <under consideration>**
  – Part 1 – Required Services
    • Health Monitor - raise application error
    • Sampling port services refresh period
    • Queuing port services
    • Ada language bindings
    • XML schema update
    • Other items to be identified
IMA and ARINC 653

- ARINC 653 is a specification for an application executive used for integrating avionics systems on modern aircraft.
- It is an API of 51 routines: time and space (memory) partitioning, health monitoring (error detection and reporting), communications via “ports”.
- ARINC 653 OS and applications are typically certified per DO-178B; different partitions can be certified to different DO-178B “levels”.

![Federated System](image)

![Integrated Modular Avionics (IMA)](image)
ARINC 653 APEX APplication EXecutive Application Programming Interface

• The ARINC 653 APEX API provides the following services:

  Process Management  Buffer Management
  Time Management  Blackboard Management
  Partition Management  Semaphore Management
  Sampling Port Management  Event Management
  Queuing Port Management  Error Management

• An API for C and Ada is defined
ARINC 653 Advantages

• Portability
  – The APEX (APplication/EXecutive) interface facilitates portability of software applications.

• Reusability
  – The APEX interface allows the production of reusable application code for IMA systems.

• Modularity
  – By removing hardware and software dependencies, the APEX interface reduces the impact on application software from modifications to the overall system.

• Integration of Software of Multiple Criticalities
  – Each application uses a virtual target (DO-178B, Section 6.4.1)
  – Supports DO-178B Level A- E on the same processor
Hierarchical Health Management

- HM Framework supports ARINC 653 model
  - Process level - controlled by the Application Developer
  - Partition level - controlled by the System Integrator
  - Module level - controlled by the Platform Provider

- Support for Cold and Warm restarts provided
  - Partition level
  - Module level

- Partition and Module Health Management is configured completely with XML
ARINC 653

The main areas where ARINC 653 is used are where there is the need of:

1. Integrating different systems into one CPU environment
2. ARINC 653 time and space partitioned systems
3. Multiple (hostile) vendors using the same processor
4. Safety-critical control systems
5. Integrated platforms with multiple OSs

+ Reduces weight, power, wiring, remote computing units

− Increases certification complexity and diligence
Example of an ARINC 653 OS
ARINC 653 RTOS Architecture

ARINC Application Executive
(with ARINC 653 ports and time/space scheduler)

Board Support Package (BSP)

Hardware Board

User Mode

ARINC API
Partition OS

POSIX API
Partition OS

RTOS API
Partition OS

Ada API
Partition OS

Kernel Mode

ARINC Application

POSIX Application

RTOS Application

Ada Application
ARINC 653 Scheduler
Priority-Preemptive Inside Partitions

Partition OS

Partition OS

Partition OS

Partition OS

T1
T2
T3

T1
T2
T3

T1
T2
T3
T4

T1
T2
T3

Partition #1

Partition #2

Partition #3

Partition #4

Time
## Typical ARINC 653 RTOS Features

### The VxWorks 653 Example

<table>
<thead>
<tr>
<th><strong>Strong Partitioning:</strong></th>
<th>• Time and Space Partitioning</th>
</tr>
</thead>
</table>
| **Multiple APIs though Multiple Partition Operating Systems (MPOS):** | • APEX (ARINC 653) – Ada and C  
• POSIX Subset – C  
• Multilanguage – Ada, C, and C++  
• Legacy possible with COIL |
| **Error Management:** | • Health management  
• Cold / Warm Restarts: 2 secs / 100 millisecs typical  
• Temporal Violation Detection (TVD) |
| **Certification Audit:** | • DO-178B Level A Certification Evidence available on hyper-linked DVD – a complete package |
| **Multiple Certification Levels on one system:** | • Robust Partitioning meets **DO-297** IMA requirements |
| **Priority Preemptive Scheduling (PPS):** | • Slack time scheduling – interrupt driver threads run in idle time of selected partitions |
| **Certification Tools – Practical IMA** | • Configuration created from XML by qualified development tool  
• Qualified “flying” monitors: CPU time, memory, ports  
• Agent for Certification Environment (ACE): debug, comm |
ARINC 653 XML-Based Configuration
Typical ARINC 653 Architecture

- IMA System Integrator
- Platform Provider
- Application Developers

User Mode vs. Kernel Mode

ARINC 653 Space/Time Scheduler
Board Support Package (BSP)
Hardware Board
So What Is RTCA DO-297?

“Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations”

• **Purpose:**
  “..provides guidance for IMA developers, integrators, applicants, and those involved in the approval and continued airworthiness of IMA systems. It provides specific guidance for the assurance of IMA systems as differentiated from traditional federated avionics”

• Results of joint US/EU Study **RTCA SC-200** and **EUROCAE WG-60**

• Defines roles and responsibilities – Certification applicant, Systems Integrator, Platform Provider, Application Developer

• References RTCA DO-178B (EUROCAE ED-12B) and ARINC 653
XML-Based Configuration

• Consistent with DO-297
• Separates control of concerns among *Platform Provider, System Integrator, and Application Developers* for configuration-based ‘plug-n-play’
  – XML schema and document divided into files for each role
  – Application XML document is a *contract* between the Application Developer and the System Integrator
  – Configuration data in an application XML document can be kept private between the application team or company and the System Integrator
Certification Considerations

How XML Can Ease the Certification of an ARINC 653 System
Expensive Industry Problem: 
Certifying ARINC 653 Configuration Data

• To certify an ARINC 653 system to DO-178B:
  – Write human-readable requirements
  – Write and run tests to prove the requirements are met

• Three ways to **certify the configuration data** (partitions, ports, health monitoring, ...) :
  – Test the entire system as a whole – all applications together – not feasible with more than 2-3 applications! Cost of change too high during initial cert and on later changes!
  – Write tests for the configuration data, and update with each change – also very expensive!
  – **Use a DO-178B qualified development tool to guarantee binary configuration data is correctly translated from requirements**
XML to Binary Compiler
DO-178B Qualified as A Development Tool

- Constrained XML input, checked and verified
- Discrete XML configuration files for each application, supplier, and integrator per DO-297
- DO-178B tool qualification eliminates the need for testing output
- No intermediate language to trace or add errors
Experience Gained in IMA Systems

• IMA systems are extremely complex:
  – Large number of applications: 10+
  – Large application: 2,000,000+ lines of code, 4-8 MBytes
  – Large configuration data: 40,000+ configuration entries

• Complexity must be managed to be successful
  – Roles and responsibilities have to be defined
  – Role activities have to be decoupled

• Development cycles are shorter and shorter

• Cost of Change must be very low
  – Introducing a change should have a low impact even during the certification cycle

• Solution: Configuration & Build Partitioning