The evolving ARINC 653 standard and it’s application to IMA

Alex Wilson
Senior Program Manager
Wind River
November 13th 2007
Agenda

• IMA and ARINC 653

• DO-297

• Certification of IMA under DO-297

• Conclusions
Why Integrated Modular Avionics?

- Allows for consolidation and portability of applications
  - Lower program lifecycle costs
- Improved software re-use
  - Reduce impact for re-using components
- Improve modularity
  - Reduce impact for application changes
- Improve portability
  - Reduce upgrade costs
  - A standard platform provides integrator with choices of vendors
- Flexibility and fault tolerance
  - Results in improved dispatch reliability
- Reduce the number of LRU’s
  - Lower maintenance costs
  - Reduce space, weight and power
- Support Multiple DO-178B Safety Levels on a single microprocessor

Honeywell claims that IMA design can save 350 pounds of weight on a narrow-body jet: equivalent to two adults.
ARINC 653 specification

- 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 OSes and applications are typically certified per DO-178B; different partitions can be certified to different DO-178B “levels”.

Federated System

Integrated Modular Avionics (IMA)

- Cockpit Displays
- Air Data Computer
- Flight Management System
- ARINC 429
- Air Data
- FMS
- Displays
- MMU-Partitioning Operating System
**Workbench Development Suite**
- Eclipse Framework
- Support for multiple OSes
  - VxWorks 653, VxWorks 6
  - Linux, VxWorks MILS
- Editor, compiler, debugger
  - C, C++, Ada*
  - On-chip debug support for Module OS and Application Partition
- Analysis tools
  - System Viewer
  - Source code analyzer
* Partner products

**DO-178B Certification Tool Suite – Cuts Cert Time, Cost**
- XML Configuration Suite
  - DO-178B Level A qualified development tool
  - Schema submitted to ARINC 653 committee
- DO-178B qualified verification tools
  - Agent for Certification Environment (ACE)
  - Port monitor
  - CPU monitor
  - Memory monitor
  - Host shell command tool

**Integrated Partner Support**
- Certifiable ARINC 664 Stack
- CORBA
- Certifiable OpenGL
- ARINC 615A Data Loader
- AFDX

**VxWorks 653**
- Time and space partitioning
  - Slack time scheduling option
  - Meets SC-200 IMA requirements
  - Up to 16 unique schedules
- ARINC 653 Supplement 2, Part 1 compliance
  - Integrated Health Management
  - Module/Partition cold/warm restart
  - ARINC SAP Ports (Part 2)
- Multiple partition OS with support for:
  - ARINC 653 API
  - VxWorks 5.5 API subset
  - POSIX subset
  - Customer legacy OS possible
- DO-178B, Level A UDP/IPv4 Network stack (optional)
- DO-178B Level A cert evidence
VxWorks 653 – designed for performance

- **VxWorks 653 implements a two-level “OS” model**
  - "Virtual machine" approach as described in DOT/FAA/AR-99/58, *Partitioning in Avionics Architectures: Requirements, Mechanisms and Assurance* authored by John Rushby
  - Corresponds to the concept of a virtual machine as described in DO-178B, section 6.4.1
  - Gives especially high scheduling performance, with the ability to run dozens of partitions with minimal RTOS partition switch overhead even at high clock rates
  - Scales from a single partition system to a maximum of 255 partitions without performance degradation seen with other implementations
VxWorks 653 Architecture

- ARINC Application
- POSIX Application
- VxWorks Application
- Ada Application

- ARINC API
- POSIX API
- VxWorks API
- Ada API

- Partition OS
- Partition OS
- Partition OS
- Partition OS

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

Board Support Package (BSP)

Hardware Board
VxWorks 653
ARINC 653 Time and Space Scheduling

Partition #1
Partition OS

Partition #2
Partition OS

MMU Partition #1 Time Allocation

MMU Partition #2 Time Allocation

Time
VxWorks 653
Priority Preemptive Scheduling Intrapartition

Partition OS

<table>
<thead>
<tr>
<th>Partition #1 Time Allocation</th>
<th>Partition #2 Time Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exec Time</td>
<td>Exec Time</td>
</tr>
<tr>
<td>Idle Time</td>
<td>Idle Time</td>
</tr>
</tbody>
</table>

Time
The ARINC 653 standard

- **ARINC 653 Specification First Published** <Jan 1997>
- **ARINC 653 Supplement 1** <Oct 2003>
  - Provided refinement and clarification to the 1997 standard
- **ARINC 653 Part 1 (Required Services) Supplement 2** <Mar 2006>
  - ARINC 653 partition management
  - Cold start and warm start definition
  - Application software error handling
  - ARINC 653 compliance
  - Ada and C language bindings
- **Added ARINC 653 Part 2** <Jan 2007>
  - Extended Services, including File System, Logbook, Service Access points...
- **Added ARINC 653 Part 3** <Oct 2006>
  - Conformity Test Specification
- **On-going work** <Next Meeting at Wind River in Alameda, California Nov 13-15 2007>
  - Part 1 Required Services – Supplement 3 <Various updates including HM and XML>
  - Part 2 Extended Services – Supplement 1 <Various updates including FS and Name Service>
  - Part 3 Conformity Tests – Supplement 1 <To include Part 2 Testing>
  - Part 4 Embedded Profiles <Proposal to develop subsets of overall standard>
So what is RTCA DO-297 /EUROCAE ED-124?

“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
Certification of IMA system

From DO-297:

“Six tasks define the incremental acceptance of IMA systems in the certification process:”

- Task 1: Module acceptance
- Task 2: Application software or hardware acceptance
- Task 3: IMA system acceptance
- Task 4: Aircraft integration of IMA system – including Validation and Verification (V&V)
- Task 5: Change of modules or applications
- Task 6: Reuse of modules or applications

Key implementation and certification challenges:-

- **How to change** application or configuration entities without affecting the entire system?
  - Without requiring re-testing or re-certification of other independent entities

- **How to reuse** applications from one IMA project on the next IMA project?
  - Without having to re-write and re-test the entire application
Certification stakeholders

Certification Applicant
- Responsible for demonstrating compliance to applicable aviation regulations
- Seeking Type Certificate (TC), Amended TC, Supplemental TC (STC) or Amended STC

System Integrator
- Integrating the “platform” and “applications” to produce “IMA System”
- System Configuration, Resource allocation, IMA V&V

Platform Provider
- Provide processing hardware and software resources (including the core software)
- Specify interfaces, shared resources, configuration tables
- Platform V&V

Application Developer
- Develops “Hosted” applications and verifies on “platform”
- Specifies external interfaces and resource requirements of application

Key implementation and certification challenges:-

How to keep supplier roles separate during configuration and build?
Typical federated system architecture

Application / Tasks

Linked With

RTOS

Include File Configuration

Board Support Package (BSP)

Hardware Board

Kernel Mode
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
Independent Build, Link, and Load

A VxWorks 653 system consists of at least four pieces:

• A Module OS (MOS) (Partition Scheduler)
• Configuration data (XML)
• At least one Partition OS (POS)
• At least one application

IBLL enables Independence of software modules

• Independent Build
  – Don't need the entire source to build one piece
  – No more "system" project that builds everything
• Independent Link
  – Don't need OS binaries to link an application
• Independent Load
  – Binaries can be loaded/updated (flashed) separately
Replaceable Software Units

Without Wind River

Configuration data

C compiler or other unqualified tool

Certify all together

App 1  App 2  App 3  App 4

Configuration data from unqualified tool

Other ARINC 653 operating systems

Configuration data (partitions, ports, etc.) in C, text, XML, created by unqualified tool—must test and certify entire system as a whole, even for minor configuration change

Higher initial development time, certification cost, cost of change

With Wind River

XML configuration data

Qualified XML compiler

Certify separately

App 1  App 2  App 3  App 4

Binary configuration data

VxWorks 653

XML-based configuration data managed by DO-178B qualified XML → binary compiler

Test, certify, and recertify applications independently and asynchronously

Result: Lower development time, initial cert cost, and cost of change
Why evolve the Supplement 1 XML schema

• The ARINC Supplement 1 XML schema is not suitable for large-scale complex real-world systems
  – It matured relatively independently of the crucial role definitions in DO-297
  – It is not sufficiently flexible for commercial airplane products

• The XML for VxWorks 653 has matured over 4 years by satisfying the requirements of 5 Boeing airplane programs
  – Including meeting the extended challenge for the 787 of working with multiple suppliers, sometimes competitors, for the full set of applications
  – One of the original authors of the Supplement 1 schema, said that “… you are starting to identify and think about problems that no other OS vendor is aware of yet. You are leading in this area…”

• Wind River, in conjunction with Verocel (lead) and the 787 IMA Supplier, is helping to contribute this knowledge back to the airplane developer community through its work on ARINC 653 Supplement 3
Example: HM Table reference

Each table must be unique!

Tables can be reused!
Example: Supplement 1 Schedule Representation

- A change to a *partition* schedule affects the entire *module* schedule!
- Hard to identify the overall schedule and schedule conflicts
Supplement 3 proposed schedule representation

Partition A

Partition B

Partition C

Partition D

Partition-Ref

Partition-Ref

Partition-Ref

Partition-Ref

Window 1

Duration

Window 2

Duration

Window 3

Duration

Window N

Duration

Major Frame

Schedule 1
Applying the DO-297 stakeholder concept

- **Separate** and **organize** configuration data and build activities per IMA roles:
  - **System Integrator** (SI),
  - **Platform Provider** (PP) and
  - **Application Developers** (AP)

- Each **role** has its **own** configuration data and set of activities
- Each activity is **independent** of every other
XML Table Generator for Review of Configuration Data for Credit

- **Platform Provider**
  - XML Tables
  - XML Config File

- **System Integrator**
  - XML Tables
  - XML Config File

- **Application Developers**
  - XML Tables
  - XML Config File
  - FMA
  - Nav

- **Display**
  - XML Config File

- **XML Table Generator**
  - DO-178B Qualified Verification Tool

- **Reviewers, DERs and Certification Authorities**
  - Platform Data
  - Schedule Tables
  - HM Table
  - Application FMA
  - Application Nav
  - Application Display

**XML Business Rules**
Typical ARINC 653 XML Compilation

- Unconstrained XML Input
- The configuration files for a single platform can be large (50,000 lines of XML or more)
- Translation to intermediate language
- Very large C data file
- Translation to binaries
- Load binaries onto target
XML Data Testing

• Every translation must be traced!
  – Configuration requirements to XML configuration data
  – XML configuration data to C code
  – C code to binaries

• All tools must be proven to be reliable and consistent

• The entire process must be proven as reliable and repeatable

• Tests must be written for every XML configuration
  – How can one edit and test a large data file reliably?
VxWorks 653 XML compilation

- 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
Wind River’s XML configuration solution

A DO-178B Qualified Development Tool Suite using XML for Configuration of ARINC 653 Systems

- Updated XML schema with heritage in ARINC 653 Supplement 1
  - Improves Supplement 1 design, now proposed for ARINC 653 Supplement 3

- XML File Checker performs many consistency checks to verify consistency of configuration, qualified as a DO-178B verification tool

- XML Compiler qualified to DO-178B Level A under FAA 8110.49 Chapter 9 as a development tool
  - No further test of binary configuration data or qualification required

- XML Table Generator translates XML to human-readable tables organized by role, qualified as a DO-178B verification tool

Result: Build, debug, test, re-test, and certify each independent application independently, incrementally, and asynchronously
Benefits

• Clearly defines *responsibility* and *ownership* of configuration data
• Enables each configuration entity to be submitted *independently*
• *Incremental* changes can be introduced *without* impacting the entire program
• Preserves *confidentiality* between parties since configuration data sharing is not required (except with System Integrator)
• Establishes the notion of *contracts* between roles
• *Minimizes* “cost of change”
• Creates *manageable* configuration data set
Conclusion

• ARINC 653 Standard is being evolved and augmented as it is used on real projects such as the Boeing 787 Dreamliner

• IMA global best practices have emerged into new standards
  – DO-297/ED-124 and ARINC 653 Supplement 3

• IMA systems are extremely complex and must be carefully managed

• Configuration and development processes are key factors for successful certification

• Special emphasis should be put on both areas from the start of a program

• Both areas require careful design
Questions?

Alex Wilson
Senior Program Manager
alex.wilson@windriver.com
http://www.windriver.com