

Automated Testing of Aircraft Controller Modules

Dr. Oliver Meyer

Verified Systems International GmbH, Bremen, Germany

5th ICS Test, Düsseldorf, April, 2004

Overview

- ▶ IMA Controllers and their Characteristics
- ▶ Test System
- ▶ Test Approach and Objectives
- ▶ Testing with Different Configurations
- ▶ Data Flow Based Testing
- ▶ Conclusion

Integrated Modular Avionics

- ▶ Current Situation
 - Different Controllers for Different Applications
 - Different Interfaces, Operating Systems, etc.
 - ➡ High Expenses for Validation, Verification, Qualification

- ▶ Integrated Modular Avionics (IMA)
 - Unified Hardware Platform for Avionics Controllers
 - Provides a Set of Standard IO Interfaces
 - Highly Configurable
 - Shared Hardware for Multiple Applications

IMA Module Characteristics

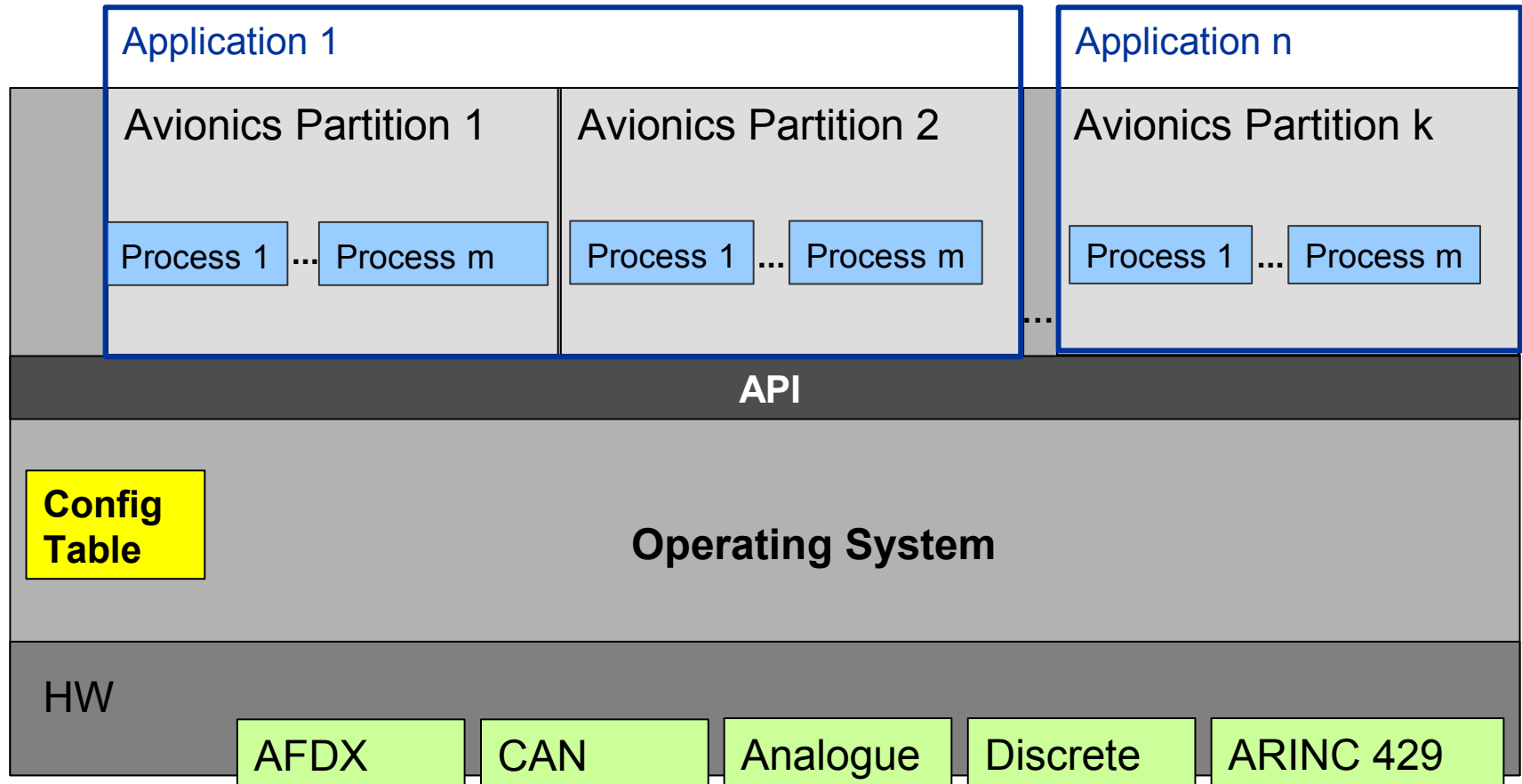
- ▶ IMA Module
 - Large Variety of Different I/O types
 - Discretes, Analogues, CAN, ...
 - AFDX (Avionics Full Duplex), ARINC-429
 - Built-In Test Equipment (BITE)
 - IMA Module Configuration Definition (ICD)
- ▶ ARINC 653 Based Operating System
 - Hard Real-Time Capability
 - Spatial and Temporal Partitioning
 - Unified Access to Different I/O Types
 - Standard API Interfaces for C and Ada

Configurability

- ▶ Large Number of Potential Applications Requires Flexibility

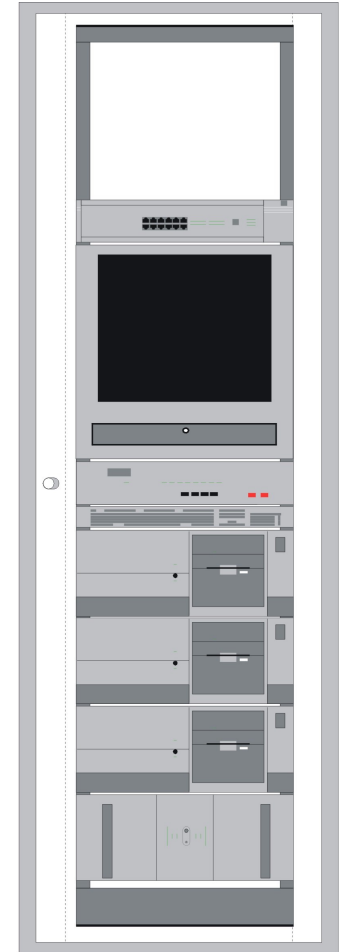
- ▶ ICD allows to define
 - I/O Allocation, Routing, Conversion, Bandwidth-Limitations, ...
 - I/O Data Formats
 - Scheduling of Partitions
 - Memory Allocation and Limitations
 - Health Monitoring Reactions
 - General Module Characteristics (Caching, RAM sizes, ...)

IMA Module Overview



Test System – Linux Cluster

- ▶ Hardware:
 - Cluster Based on Three Multi-Processor PCs
 - Scalable Cluster Size
 - Myrinet for Connecting the Cluster Nodes
- ▶ Operating System:
 - Linux Kernel with Patch for CPU Reservation
 - Hard Real-Time Communication using Myrinet Ringbuffers
 - Time Synchronisation between the Cluster Nodes
 - Roundtrip Time 40kByte: < 250 μ s



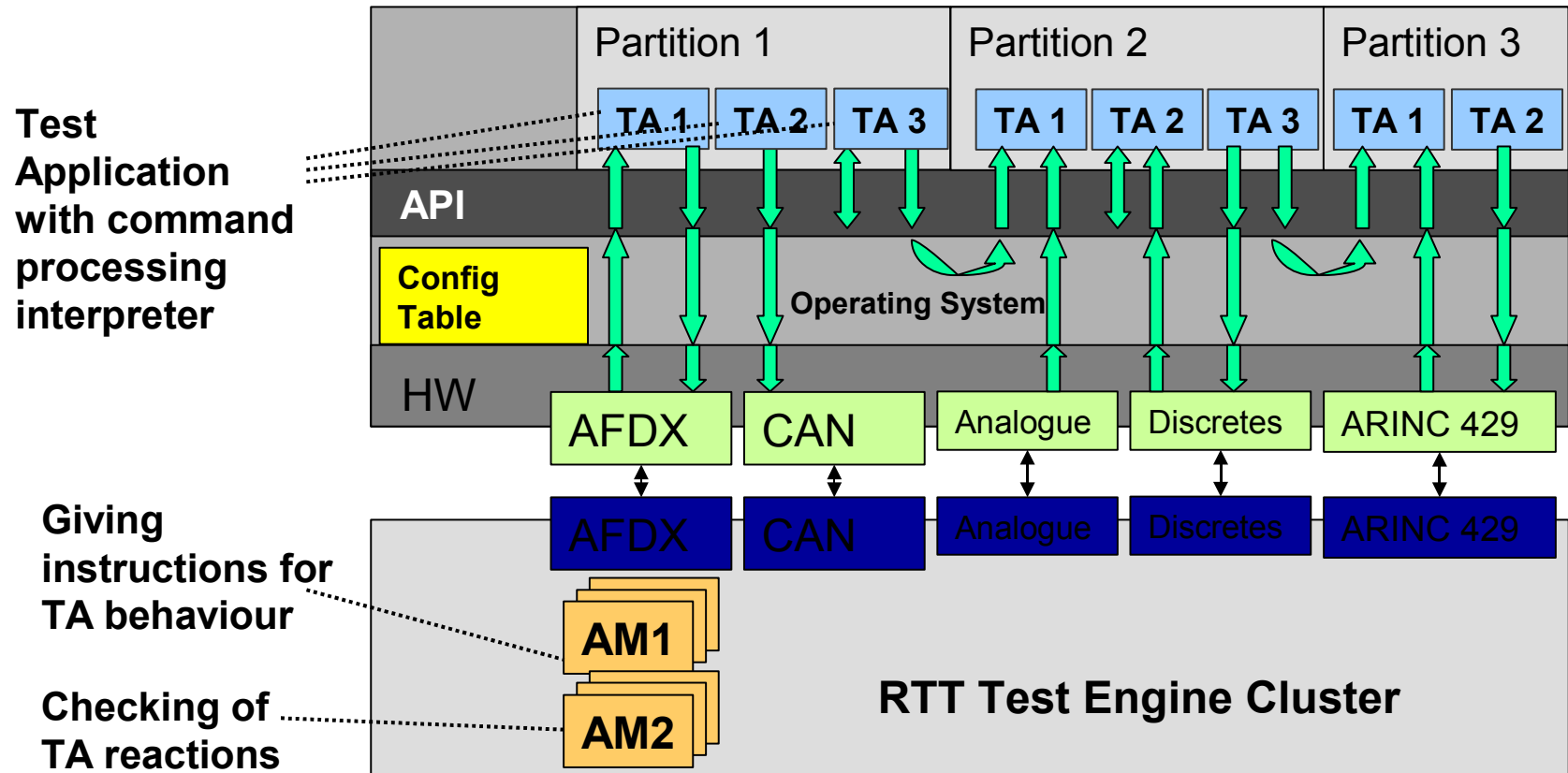
IMA Module Testing: Strategy

- ▶ **Verification Step 1:**
Demonstrate correct behaviour for arbitrary configurations compatible with the underlying CPU and I/O hardware
- ▶ **Verification Step 2:**
Demonstrate correct behaviour for an application specific module configuration
- ▶ **Verification Step 3:**
Demonstrate correct behaviour for network of configured IMA modules
- ▶ **Verification Step 4:**
Demonstrate the correct behaviour of the integrated application software

Test Approach (1)

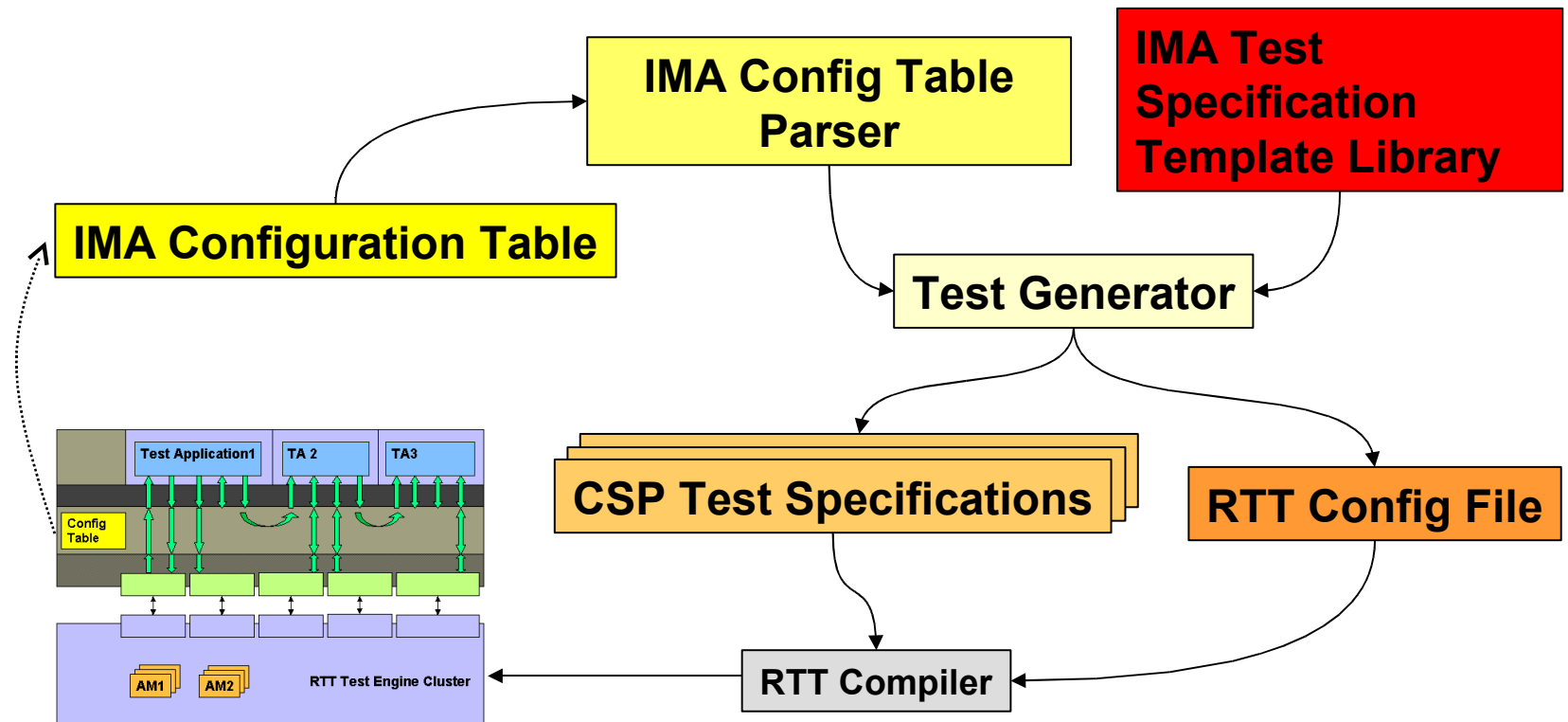
- ▶ Test Applications (TA)
 - Command Process Interpreter
 - Pre-programmed Behaviours ("Scenarios"), e.g.
 - Illegal access of resources for robustness tests
 - Burst loads on external interfaces (e.g., CAN, AFDX) or inter-application interfaces
 - Test Applications are Generic (Large Behavioural Variations)
- ▶ Test Specification Template Library
 - Can be Instantiated with Different IMA Configurations
 - Test Specifications Send Commands to the TAs and Check their Correct Behaviour

Test Approach (2)



Test Approach (3)

► Automatic Instantiation of Test Specifications



Test Objectives

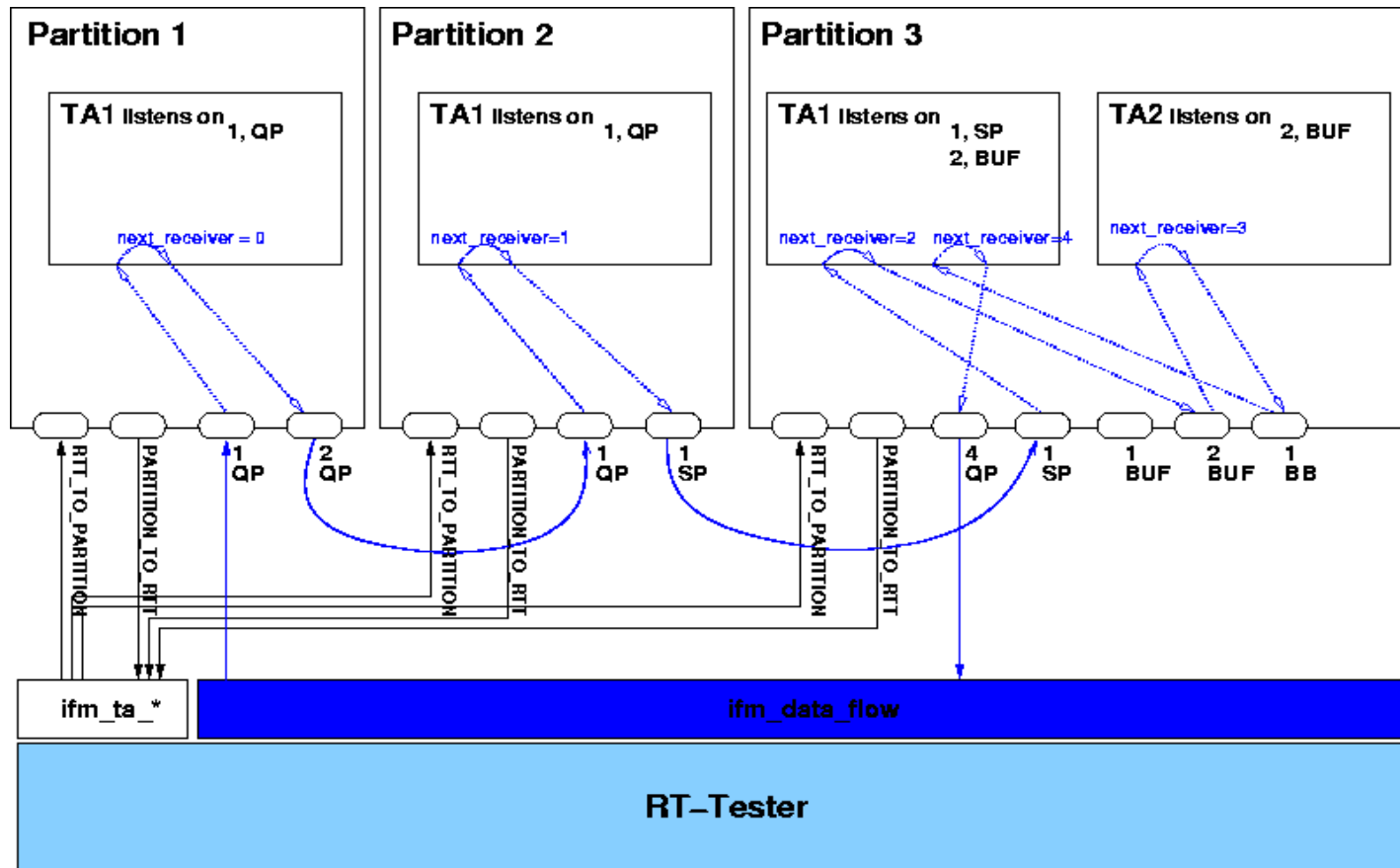
- ▶ Normal Behaviour Testing, e.g.
 - OS Services (i.e., API Calls)
 - Transitions of Operational Modes
 - Access Configured Internal and External Interfaces
 - Different Configurations to Exercise Wide Range of IMA Capabilities
 - Checks Data Loading

- ▶ Robustness Testing, e.g.
 - Illegal API Calls and API Calls with Illegal Parameter Values
 - Checks whether Partitioning is Ensured by
 - read/write operations to illegal memory areas
 - maximum/illegal CPU allocation by stress tests
 - illegal access to internal and external interfaces
 - Checks that Illegal Configurations cannot be Loaded
 - Checks the Handling of Power Interrupts

Configuration Generator

- ▶ Automatic Generation of Legal & Illegal ICDs
 - Generation of “Artificial” Configurations to Exercise the Widest Possible Range of IMA Capabilities
 - Based on Rules for Modifying Existing Configurations
 - CLONE
 - DELETE
 - MODIFY
 - CONNECT
 - AUTOCONNECT
 - One Rule can Generate a Set of IMA Configuration Tables
 - `MODIFY PARTITION 1 QP * PORT_MAX_MESSAGE_SIZE={1..8192} (3)`
 - `MODIFY PARTITION [1,2] QP * PORT_MAX_MESSAGE_SIZE={1..512} <3>`

Data Flow Based Testing



Dataflow Message

- Size
- Seq. ID
- CRC
- Next Recv.

List of

- Receivers
- Ports
- Timestamps

Data Flow Test Properties

- ▶ Data Flow Scenarios for Checking
 - Behaviour in Case of High CPU and I/O Load
 - Inter-Partition Communication
(Using Different I/O Types in Parallel)
 - Intra-Partition Communication
(Using Different Communication Mechanisms)
 - Timing Properties
 - Scheduling Properties (Process and Partition Level)
 - Partitioning Properties (Robustness Tests)

Conclusion / Lessons Learned

- ▶ Standard-PC Cluster Based Real-Time Testing Approach
- ▶ Test Template Library for
 - Checking Behaviour for Large Number of Configurations (ICD Parser / Generator Based Approach)
 - Automated Regression Testing in case of HW / SW Upgrades
 - Re-Use of Tests for Different Module Types
- ▶ Data Flow Test Concept for Various Test Objectives
- ▶ Test Automation Level Mainly Depends on Interfacing
- ▶ Automated Test Analysis/Documentation is Crucial