ARIANE 6 FLIGHT SOFTWARE DESIGNED FOR A SIMPLER VALIDATION

Ada-Europe 2018
Lisbon, Portugal

Presented by Philippe Gast
20thJune 2018
A world leader in access to space, serving institutional and commercial customers and supporting Europe’s strategic independence

The flight software is developed by Ariane Group
Ariane 6 maiden flight 2020
THE SOFTWARE OF A LAUNCHER IS RESPONSIBLE FOR

Mission description

Engine ignition

Flight control

Stage release

Attitude control

…
A FLIGHT SOFTWARE IS MAINLY COMPOSED OF

- **Launchers sequential**

**GNC**

- **Sensors**
- **Navigation**
- **Guidance**
- **Control**
- **Actuators**

**Control algorithms**

- Acquisition of measurement
- Where am I?
- Where shall I go?
- Compute the commands
- Send commands to actuators
THE ARIANE 6 FLIGHT SOFTWARE DESIGN DRIVERS

The objective is to define a software real time design which is correct by construction and fully deterministic

• To reduce the tests combinatorial,
• To be robust to system changes,
• To be consistent with incremental development approach,
• To permit software debug in non-real time environment

The solution

• A method to define the functional architecture of the system: the Functional Unit approach
• A synchronous design for the Flight Software synchronised with avionics,
• A dedicated light validation test facility to support debug
THE FUNCTIONAL UNIT APPROACH

Control

- Lower Ctrl
- Upper Ctrl
- Stabilizing
- Nominal
- Ctrl lower stage
- Stabilize

FU commands
- Upper Ctrl
- Lower Ctrl
- Ctrl upper stage

FU command
- Internal command

Mission event
- Launcher sequence
- End of thrust
- Pre-stop
- Stop
- Pressurization

Lower module
- On
- Pre-stop
- Stop

Upper module
- Off
- Pre-On
- On
- Pressurization
Flight Software design rules

- Flight Software implementation strongly similar to system Functional Definition
- Functional software processing implemented in a limited number of cyclic tasks
- No acyclic tasks
- High consistency/low coupling limiting interface between Functional Unit
EXTENDED RMS (RATE MONOTONIC SCHEDULING)

- Harmonic task 1
- Harmonic task 2

Time triggered communication per rendezvous

Real-time architecture manually designed
- List of threads, periods, deadlines, offsets
- Deployment of functional blocks on threads
- Mission Launcher Manager executed step by step at highest frequency
A DEDICATED REALTIME SEQUENCER

Basic Cyclic Task
- period = Minor Cycle
- activation shift (2)
- deadline shift (1)

Harmonic Cyclic Task 1
- activation shift (7)
- deadline shift (13)
- Maximum execution time (3)

Harmonic Cyclic Task 2
- period = Major Cycle
- activation shift (21)
- deadline shift (21)
- Maximum execution time (6)

Middleware Synchronization Event
- activation rendezvous processing
- deadline rendezvous processing

No shared data

Middleware Synchronization Event
- activation rendezvous processing
- deadline rendezvous processing
Processor’s redundancy concept

- Same on-board software running in each processor in hot redundancy
- No data exchange between the two processors
- Each processors receives same data
- Processor fail silent

Avionics Communication Bus

- The Communication System is in charge of exchanging functional and telemetry data flows between avionic equipment of the launcher during both ground and flight phases. It is based on Time Triggered Ethernet (TT Ethernet).
I/O SYNCHRONISATION : LAST IN NEXT OUT

On target

Cluster cycle interrupt from TTE

Reactivity of task T1

Reactivity of task T2

Priority

BG

T2

T1

TTE Bus (In)

TTE Bus (Out)
FLIGHT SOFTWARE VALIDATION

2 types of Software Validation Facilities

Hardware Software Validation Facility (HSVF)
- With real on-board computer and avionics Interface (Time Trigger Ethernet)
- Real-time representativeness / full functional representativeness
- To be used for Ariane 6 flight software formal validation
- Expansive (only one item)

Emulated Software Validation Facility (ESVF)
- With on-board processor emulator executed on host
- NOT real-time representativeness / full functional representativeness (*independently of real-time aspects*)
- Cheap (no dedicated Hardware)
- Used for flight software validation preparation/debug

Test preparation is similar between HSVF and ESVF; test execution is faster on ESVF (no real-time)
HARDWARE SOFTWARE VALIDATION FACILITY (HSVF)

Hardware Software Validation Facility

- On-board Software under test (Cross compiled)
  - Applicative Software
  - Middleware (in line with target hardware)

On-board Software execution facility:
- Real hardware target

Equipment Simulator execution facility:
- Host (Linux Environment)
  - Configuration
  - Output Data
  - Equipment models (open loop / closed loop)
  - Equipment Simulator Input Data

Spy

Probe Output for analysis

Output for analysis

User
EMULATOR SOFTWARE VALIDATION FACILITY (ESVF)

User

Emulator Software Validation Facility

On-board Software under test (Cross compiled)
Applicative Software
Output Data

Middleware (in line with GNATemulator/GNATBus)

On-board Software execution facility:
ARM CPU simulator (GNATemulator/GNATBus)

On-board Software / Equipment Simulator Interface (functional data)

ESVF Gateway

Equipment Simulator execution facility
Configuration
Output Data

Equipment models (open loop / closed loop)
SVF Equipment Simulator

Equipement Simulator Input Data

Coverage tool
Debug and Analysis capabilities
Output for analysis
Socket Spy
Exchange Protocol (Socket)

User

This document and its contents are property of Arianegroup. It shall not be communicated to any third party without the owner’s written consent. ArianeGroup SAS - All rights reserved.
REAL-TIME SEQUENCER ON ESVF

- MainTask
- Basic Cyclic Task
- Harmonic Cyclic Task 1
- Harmonic Cyclic Task 2

- Simulated Synchronization Event
- activation rendezvous processing
- deadline rendezvous processing
- activation rendezvous processing
- deadline rendezvous processing
INPUT/OUTPUT SYNCHRONISATION ON ESVF

Cycle N
Cycle N+1
Cycle N+2
Cycle N+3
Cycle N+4

Basic Cyclic task
Harmonic task
Simulated Cluster cycle interrupt

Socket I/F
System Simulator
USAGE HSVF / ESVF: PREPARE TEST PROCEDURE ON ESVF BEFORE EXECUTION ON HSVF

Test Execution Parameters:
- Ground scenario
- Equipment scenario
- Product under test
- Test Duration
- ...

HSVF
- On board computer
- executable HSVF
- HSVF Simulator

ESVF
- GNAT EMU
- executable ESVF
- ESVF Simulator

Switch

Gateway

SPY.PCAP (TTE/TFTP)
- Data payload

SPY.PCAP (UDP/TCP)
- Data payload

Data payload

Analysis tools

Test Analysis Parameters:
- TestCases
- ...

Same TestCases Results

Same TestCases Results

TestCases Results
USAGE HSVF / ESVF: REPLAY FUNCTIONALITY ➔ DEBUG REAL RUN ON ESVF

Test Execution Parameters:
- Ground scenario
- Equipment scenario
- Product under test
- Test Duration
- ...

Error detected on target (Software validation or Qualification)

Full debug capability

SPY.PCAP (TTE/TFTP)
Data payload

SPY.PCAP (UDP/TCP)
Data payload
CONCLUSION

Flight software fully synchronous design consistent with System Functionnal breakdown method (Functional Units approach)

• Lean transition from System to Software

• Easier to debug flight software or test: Emulator Software Validation Facility provides full debug capability with behavioural flight representativeness

• Emulator Software Validation Facility is available on software development platform

Such approach for launcher is an adapted reuse from orbital vehicle (Automated Transfer Vehicle). It can then be applied to various types of Software Intensive Systems.
Any questions?
philippe.gast@ariane.group