EDISOFT

DEFENCE & AEROSPACE TECHNOLOGIES

A THALES Group Company

EDISOFT

DEFENCE & AEROSPACE TECHNOLOGIES

RTEMS LEON Upgrade

TEC-ED & TEC-SW Final Presentation Days 2014 May

Noordwijk, Wednesday, 21st May 2014

A THALES Group Company

DEFENCE & AEROSPACE TECHNOLOGIES

Agenda

RTEMS LEON Upgrade

INTRODUCTION AND OBJECTIVES

Real-Time Operating System for Multiprocessor Systems (RTEMS)

Community: www.rtems.org

RTEMS CENTRE: http://rtemscentre.edisoft.pt

RTEMS LEON Upgrade

ESA Contract Number 4000103825

General Support Technology Programme (GSTP)

Start: 9th February 2012

End: 31st March 2014

Background Projects

RTEMS Validation and Testing - Saab Space AB

- Validation in ERC32
- Subset of RTEMS Managers
- Parts of the Kernel out of the study
- Phase 1 – Code assessment – Manual Inspection and Collection of metrics
- Phase 2 – Tests Specification

Software Safety and Dependability Evaluations - Critical Software

- Validation in ERC32
- Robustness and Stress Testing of RTEMS API

Background Projects

RTEMS CENTRE – Maintenance and Support CENTRE for RTEMS operating system - EDISOFT

- Acquire Technical Expertise in RTEMS
- Development of Support Tools for RTEMS (Timeline Tool and RTEMS and Application Configuration Tools)
- Development of Support Platform for RTEMS CENTRE (http://rtemscentre.edisoft.pt), including Problem Reporting Tool

RTEMS Improvement - EDISOFT

- Facilitate the qualification of RTEMS for Space Missions, Galileo Software Standards for Development Assurance Level-B
- Validation in ERC32, LEON2 and LEON3
- 100% Statement Coverage
- 100% Decision Coverage

DEFENCE & AEROSPACE TECHNOLOGIES

RTEMS LEON Upgrade Objectives

Implementation of Requirements identified in the RTEMS Improvement Project Software Criticality Analysis

Update the GNU Compiler Collections (GCC) and Binutils (Assembler and Linker) toolchain for RTEMS compilation

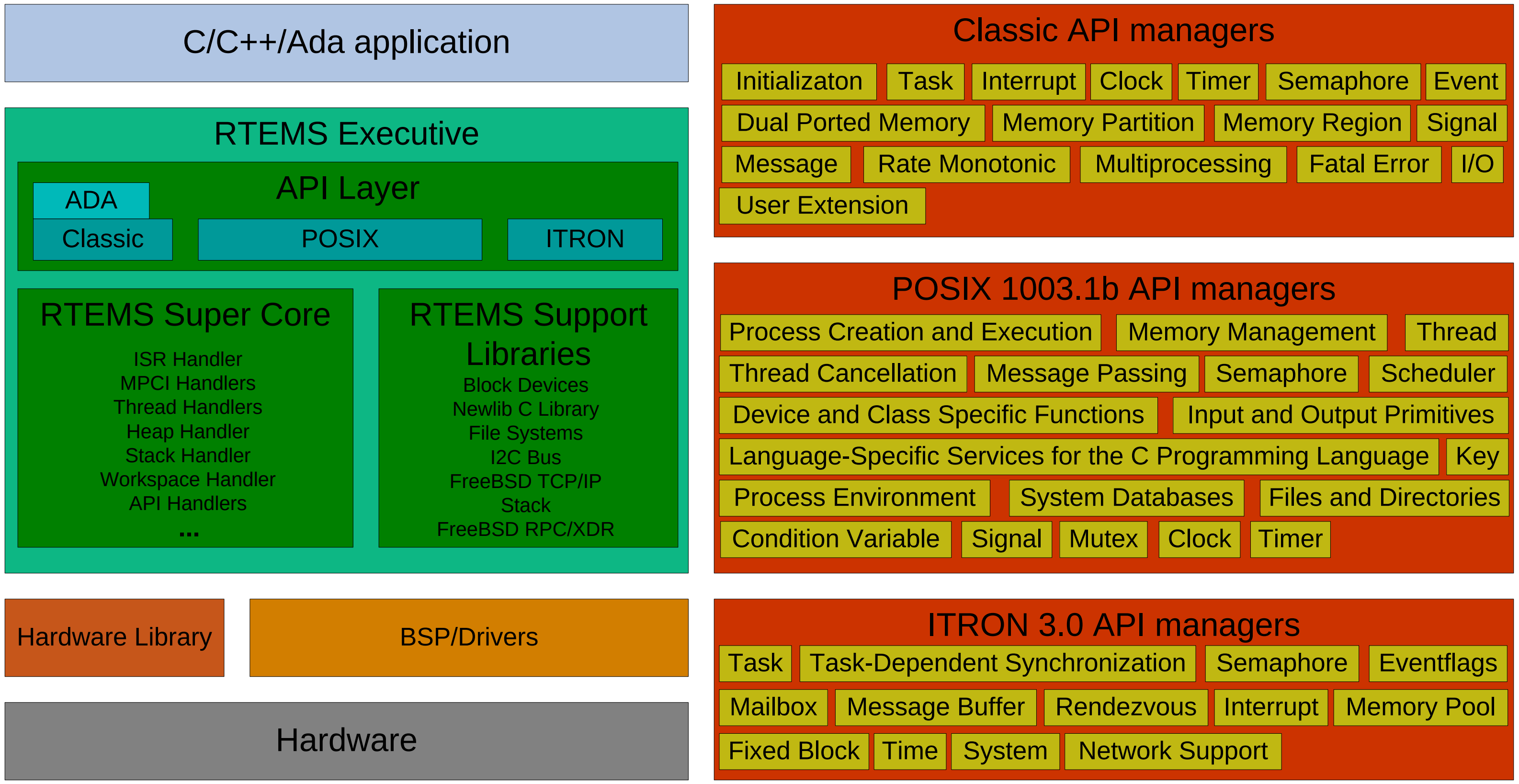
Technical Support for Integrated Modular Avionics (IMA) Projects (Xtratum, AIR and PikeOS)

DEFENCE & AEROSPACE TECHNOLOGIES

RTEMS LEON Upgrade

OVERVIEW

A THALES Group Company



C/C++/Ada application
RTEMS Executive
API Layer
ADA
Classic
POSIX
ITRON
RTEMS Super Core
ISR Handler
MPCI Handlers
Thread Handlers
Heap Handler
Stack Handler
Workspace Handler
API Handlers
...
RTEMS Support Libraries
Block Devices
Newlib C Library
File Systems
I2C Bus
FreeBSD TCP/IP Stack
FreeBSD RPC/XDR
Hardware Library
BSP/Drivers
Hardware
Classic API managers
Initializaton
Task
Interrupt
Clock
Timer
Semaphore
Event
Dual Ported Memory
Memory Partition
Memory Region
Signal
Message
Rate Monotonic
Multiprocessing
Fatal Error
I/O
User Extension
POSIX 1003.1b API managers
Process Creation and Execution
Memory Management
Thread
Thread Cancellation
Message Passing
Semaphore
Scheduler
Device and Class Specific Functions
Input and Output Primitives
Language-Specific Services for the C Programming Language
Key
Process Environment
System Databases
Files and Directories
Condition Variable
Signal
Mutex
Clock
Timer
ITRON 3.0 API managers
Task
Task-Dependent Synchronization
Semaphore
Eventflags
Mailbox
Message Buffer
Rendezvous
Interrupt
Memory Pool
Fixed Block
Time
System
Network Support

RTEMS Manager	RTEMS Primitive
Initialization	All primitives
Task	rtems_task_create
	rtems_task_ident
	rtems_task_start
	rtems_task_restart
	rtems_task_delete
	rtems_task_suspend
	rtems_task_resume
	rtems_task_is_suspended
	rtems_task_set_priority
	rtems_task_mode
	rtems_task_get_note
	rtems_task_set_note
	rtems_task_wake_after
	rtems_task_wake_when
	rtems_task_variable_add
	rtems_task_variable_get
	rtems_task_variable_delete
Event	All primitives

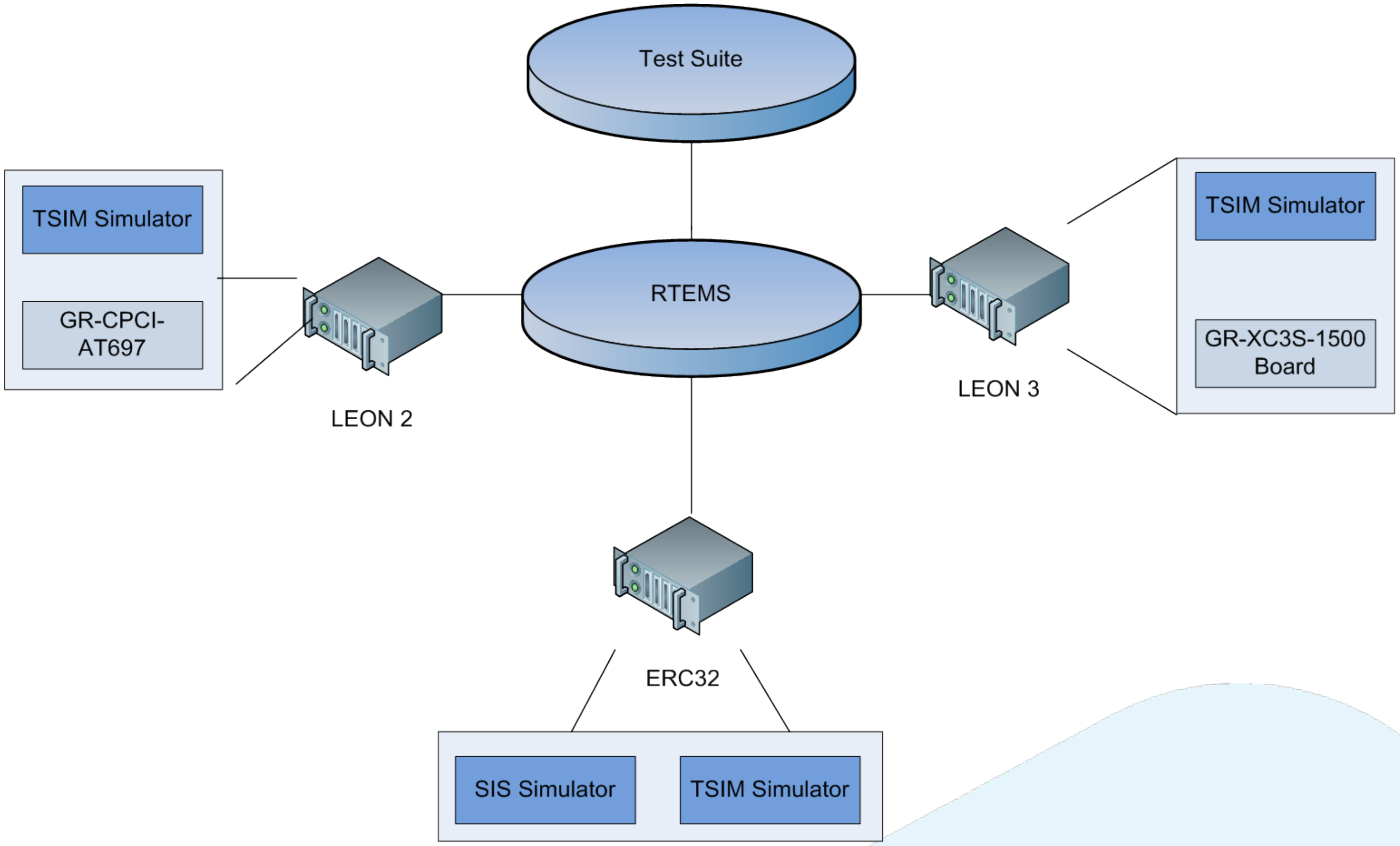
RTEMS Manager	RTEMS Primitives
Timer	All primitives
Semaphore	All primitives
Message Queue	All primitives
I/O	rtems_io_initialize
	rtems_io_open
	rtems_io_close
	rtems_io_read
	rtems_io_write
	rtems_io_control
Fatal Error	All primitives
Interrupt	All primitives
Clock	All primitives
User Extensions	All primitives
Rate Monotonic	rtems_rate_monotonic_create
	rtems_rate_monotonic_ident
	rtems_rate_monotonic_cancel
	rtems_rate_monotonic_delete
	rtems_rate_monotonic_period
	rtems_rate_monotonic_get_status

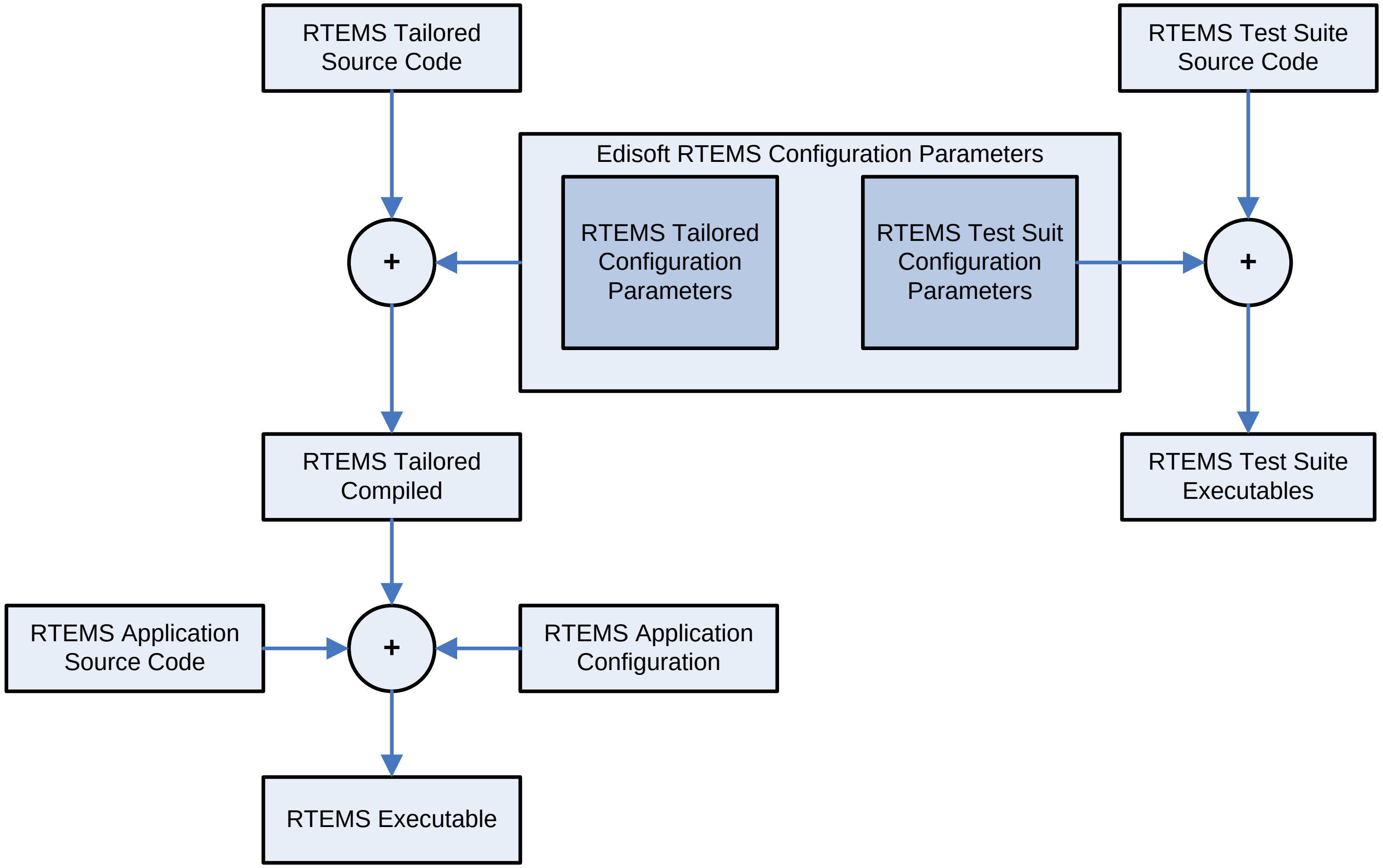
EDISOFT
DEFENCE & AEROSPACE TECHNOLOGIES

RTEMS Managers Removed
Stack Bounds Checker
CPU Usage Statistics
Barrier
Signal
Partition
Region
Dual-Ported Memory
Multiprocessing

RTEMS Manager	RTEMS Primitive Removed
Task	rtems_iterate_over_all_threads
Rate Monotonic	rtems_rate_monotonic_get_statistics
	rtems_rate_monotonic_reset_statistics
	rtems_rate_monotonic_reset_all_statistics
	rtems_rate_monotonic_report_statistics
I/O	rtems_io_register_driver
	rtems_io_unregister_driver
	rtems_io_register_name
	rtems_io_lookup_name

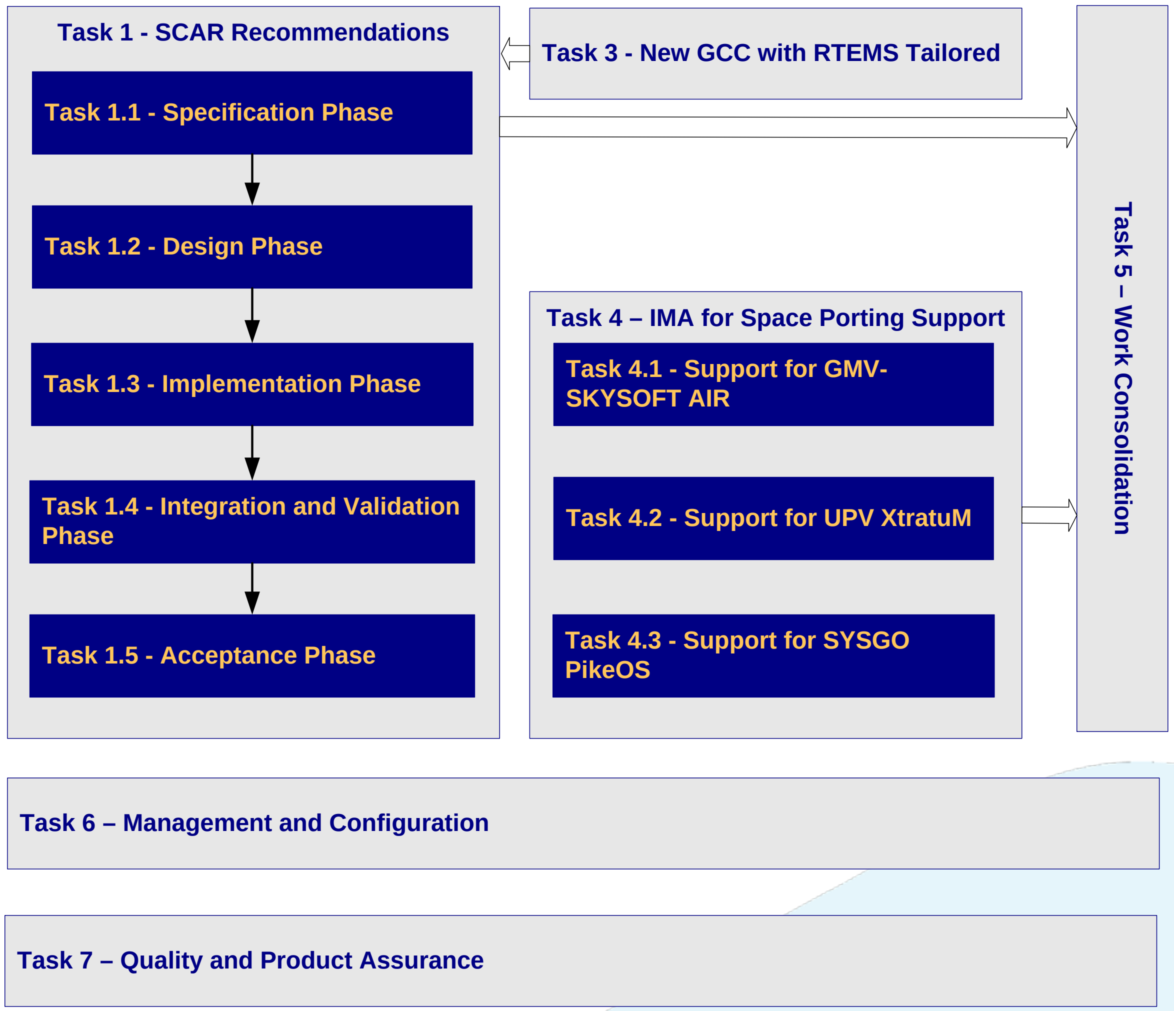
DEFENCE & AEROSPACE TECHNOLOGIES



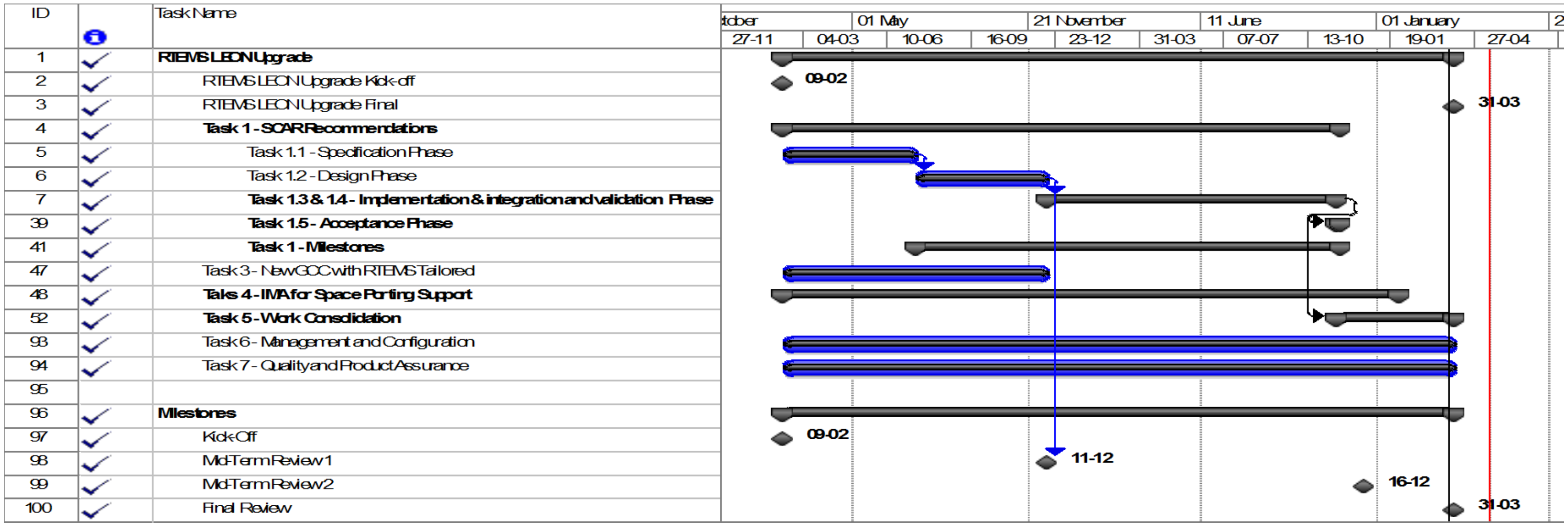

RTEMS Tailored Source Code
RTEMS Test Suite Source Code
Edisoft RTEMS Configuration Parameters
RTEMS Tailored Configuration Parameters
RTEMS Test Suit Configuration Parameters
+
+
RTEMS Tailored Compiled
RTEMS Test Suite Executables
RTEMS Application Source Code
+
RTEMS Application Configuration
RTEMS Executable

EDISOFT
DEFENCE & AEROSPACE TECHNOLOGIES

RTEMS LEON Upgrade – Study Logic


Task 1 - SCAR Recommendations
Task 1.1 - Specification Phase
Task 1.2 - Design Phase
Task 1.3 - Implementation Phase
Task 1.4 - Integration and Validation Phase
Task 1.5 - Acceptance Phase
Task 3 - New GCC with RTEMS Tailored
Task 4 – IMA for Space Porting Support
Task 4.1 - Support for GMV-SKYSOFT AIR
Task 4.2 - Support for UPV XtratuM
Task 4.3 - Support for SYSGO PikeOS
Task 5 – Work Consolidation
Task 6 – Management and Configuration
Task 7 – Quality and Product Assurance

DEFENCE & AEROSPACE TECHNOLOGIES



ID		Task Name
1	✓	**RTEMS LEON Upgrade**
2	✓	RTEMS LEON Upgrade Kick-off
3	✓	RTEMS LEON Upgrade Final
4	✓	**Task 1 - SCAR Recommendations**
5	✓	Task 1.1 - Specification Phase
6	✓	Task 1.2 - Design Phase
7	✓	**Task 1.3 & 1.4 - Implementation & integration and validation Phase**
39	✓	**Task 1.5 - Acceptance Phase**
41	✓	**Task 1 - Milestones**
47	✓	Task 3 - New GCC with RTEMS Tailored
48	✓	**Taks 4 - IMA for Space Porting Support**
52	✓	**Task 5 - Work Consolidation**
93	✓	Task 6 - Management and Configuration
94	✓	Task 7 - Quality and Product Assurance
95		
96	✓	**Milestones**
97	✓	Kick-Off
98	✓	Mid-Term Review 1
99	✓	Mid-Term Review 2
100	✓	Final Review

A THALES Group Company

Deliverable	**Reference**
RLU Software Requirements Document	09060301-006.SRD
RLU Software Design Document	09060301-014.SDD
RLU Validation Test Specification	09060301-022.VTS
RLU Unit Test Plan	09060301-020.SUP
RLU Integration Test Plan	09060301-018.SIP
RLU User Manual Design Notes	09060301-008.UMDN
RTEMS Tailored	09060301-039.SFW
TestSuite	09060301-028.testsuite
RLU Validation, Unit and Integration Test Report	09060301-026.GTR
RLU Software Budget Report	09060301-012.SBR
RLU Software Acceptance Test Plan	09060301-031.SATP
RLU Procured Software Justification File	09060301-024.PSJF
RLU Verification Report	09060301-010.RIVR
OAR Testsuite	09060301-052.OARtestsuite
RLU Software Development Plan	09060301-040.SDP
RLU Configuration Management Plan	09060301-045.SCMP
Review Plan	09060301-041.RP
Progress Report	09060301-046-YYYYMMDD.PR
Final Report	09060301-042.FR
RLU Software Configuration File	09060301-016.RICF
RLU SOC with GSWS	09060301-046.SOC
RLU Software Product Assurance Plan	09060301-043.SPAP
RLU Software Product Assurance Report	09060301-044.SPAR

DEFENCE & AEROSPACE TECHNOLOGIES

Deliverable	Reference
RTEMS Improvement Software Requirements Document	09060101-006.SRD
RTEMS Improvement Software Design Document	09060101-014.SDD
RTEMS Improvement Validation Test Specification	09060101-022.VTS
RTEMS Improvement Unit Test Plan	09060101-020.SUP
RTEMS Improvement Integration Test Plan	09060101-018.SIP
RTEMS Improvement User Manual Design Notes	09060101-008.UMDN
RTEMS Improvement Validation, Unit and Integration Test Report	09060101-026.GTR
RTEMS Improvement Software Budget Report	09060101-012.SBR
RTEMS Improvement Software Acceptance Test Plan	09060101-031.SATP
RTEMS Improvement Procured Software Justification File	09060101-024.PSJF
RTEMS Improvement Verification Report	09060101-010.RIVR

DEFENCE & AEROSPACE TECHNOLOGIES

RTEMS LEON Upgrade

SOFTWARE CRITICALITY ANALYSIS RECOMMENDATIONS

A THALES Group Company

EDISOFT
DEFENCE & AEROSPACE TECHNOLOGIES

Software Criticality Analysis Requirements

SW-FMECA 2, 3 and 18 (System-wide Error Report and Storage)

SW-FMECA 5, 6 and 7 (Rate Monotonic Deadline definition)

SW-FMECA 8 (Removal of Dynamic Memory Allocation from RTEMS Initialization)

SW-FMECA 17 (Stack Bounds Check)

SW-FMECA 19, 20, 21, 22, 23, 24 and 25 (Improvement of Semaphores with priority inheritance and ceiling and Interrupt Mask and Unmask)

SW-FMECA 2, 3 and 18 Requirements

22 New Requirements

RTEMS shall make available a **system-wide error reporting function (usable by either the System or the User Application)**

RTEMS shall record the **fatal** and **non-fatal errors**

RTEMS shall only be able to report on **100 fatal errors** and **200 non-fatal errors**, kept in a **ring-buffer**

SW-FMECA 2, 3 and 18 Requirements

The Internal Error shall report and record in the **Super Core Internal Error Handler** the**:**

- Source of the Error
- Name of the detector (application or RTEMS)
- Error type
- File and line where the error was detected
- Time of occurrence

Fatal errors shall be of the type**:**

- API
- Super API
- Super Core
- Hardware
- Device Driver

SW-FMECA 5, 6 and 7 Requirements

24 New Requirements

The RTEMS **Rate Monotonic Manager** shall make available **a deadline verification mechanism (defined and reactivated by the User Application),** coupled to a rate monotonic task's execution period

The RTEMS **Rate Monotonic Manager** shall allow obtaining the **current state of a deadline**

The **Application** shall specify the **deadline expiration handler during the creation of a rate monotonic object. The handler** shall be **invoked** when the **rate monotonic deadline is expired**

RTEMS shall be able to **calculate and report the execution time (maximum and minimum)** of the rate monotonic object

SW-FMECA 8 (Removal Dynamic Memory Allocation in Initialization) Requirements

Requirements Removal

- RI-SR-FUNC-16090 - Workspace Allocation/Deallocation
- RI-SR-FUNC-16100 - Heap Allocation/Deallocation
- RI-SR-FUNC-18030 - Extra stack configuration
- RI-SR-FUNC-01110 - Task variables

SW-FMECA 17 Requirements

4 New Requirements

RTEMS shall initialize a **task’s stack header and footer** (represented by 2 unsigned 32-bit integers) to values **0xAAAAAAAA and 0x77777777**

During a **task context switch,** if RTEMS verifies that the current task stack’s header/footer has been changed from its initial value, **it shall issue an internal fatal error with value INTERNAL_ERROR_TASK_STACK_OVERFLOW/UNDERFLOW**

SW-FMECA 19, 20, 21, 22, 23, 24 and 25 Requirements

13 New Requirements

RTEMS shall be able to **mask/unmask** a **specific interrupt**

RTEMS shall allow the user to **verify if a specific interrupt is masked or unmasked**

RTEMS **shall not allow** that a task that owns **semaphores having priority inheritance or priority ceiling** protocols to be **suspended**

RTEMS **shall not allow** a task that owns a **semaphore with priority inheritance/ceiling protocol to be blocked on any call, other than the obtain of a semaphore with priority inheritance protocol**

EDISOFT

DEFENCE & AEROSPACE TECHNOLOGIES

SW-FMECA 19, 20, 21, 22, 23, 24 and 25 Requirements

RTEMS **shall not allow** that a task owning a semaphore with priority inheritance protocol or priority ceiling protocol **to change its priority (except by the defined automatic inheritance protocol selected)**

RTEMS **shall not allow** that a task holding semaphores with priority inheritance protocol or priority ceiling protocol **change its mode to non-preemptable**

RTEMS **shall not allow a task in non-preemptive mode to obtain any semaphores with priority inheritance or priority ceiling protocol**

RTEMS shall not allow a task to own at the same time semaphores with different priority protocols

DEFENCE & AEROSPACE TECHNOLOGIES

A THALES Group Company

DEFENCE & AEROSPACE TECHNOLOGIES

SW-FMECA 2, 3 and 18 Architecture, Design and Implementation

New Components/Files

- cpukit/rtems/include/rtems/rtems/interr.h – with the definition of error manager types and the user application interfaces **rtems_error_report** (to report an error), rtems_error_get_latest_non_fatal_by_offset (to get a non-fatal error) and rtems_error_get_latest_fatal_by_offset (to get a fatal error)
- cpukit/rtems/src/**interrgetlatestfatalbyoffset.c** – implementation of **rtems_error_get_latest_fatal_by_offset** (to get a fatal error)
- cpukit/rtems/src/**interrgetlatestnonfatalbyoffset.c** – implementation of **rtems_error_get_latest_non_fatal_by_offse**t (to get a non-fatal error)

- cpukit/score/src/**interrcontrolinitialize.c** – for the initialization of the error control
- cpukit/score/src/**interrreport.c** – with the implementation of Error Report handler
- cpukit/score/src/interrmessagegetindex.c – implementation of _Error_Message_Get_Index to collect the messages from the ring buffer

136 Files modified

EDISOFT
DEFENCE & AEROSPACE TECHNOLOGIES

SW-FMECA 5, 6 and 7 Architecture, Design and Implementation

New Components/Files

- cpukit/rtems/src/ratemondeadline.c – implementation of **rtems_rate_monotonic_deadline** to be used by the application to insert a deadline.
- cpukit/rtems/src/ratemongetdeadlinestate.c – implementation of **rtems_rate_monotonic_get_deadline_state** to be used by the application to collect the deadline state

- cpukit/rtems/src/**ratemondeadlineinsert.c** – implementation of _Rate_monotonic_Deadline_Insert to insert a deadline of a periodic task in the SuperCore
- cpukit/rtems/src/**ratemondeadlineremove.c** – implementation of _Rate_monotonic_Deadline_Remove to remove the deadline definition of a periodic task from the SuperCore
- cpukit/rtems/src**/ratemondeadlinetickle.c** – implementation of _Rate_monotonic_Deadline_Tickle to perform and check the deadline state in every clock tick of RTEMS

78 Files modified

SW-FMECA 8 Architecture, Design and Implementation

Components/Files Removed

- cpukit/rtems/src**/taskvariable**get.c
- cpukit/rtems/src/taskvariableadd.c
- cpukit/rtems/src/taskvariabledelete.c

- cpukit/score/include/rtems/score/wkspace.h
- cpukit/score/include/rtems/score/heap.h
- cpukit/score/src/heapallocate.c
- cpukit/score/src/**heap**.c
- cpukit/score/src/heapfree.c
- cpukit/score/src/**wkspace**.c
- cpukit/score/inline/rtems/score/heap.inl
- cpukit/score/inline/rtems/score/wkspace.inl

195 Files modified

SW-FMECA 17 Architecture, Design and Implementation

Modified Components/Files

- cpukit/rtems/src/taskdelete.c – no major update in this file. Just included a modification to STATES_BEING_DELETED state
- cpukit/score/include/rtems/score/interr.h – with the definition of INTERNAL_ERROR_TASK_STACK_UNDERFLOW and INTERNAL_ERROR_TASK_STACK_OVERFLOW
- cpukit/score/include/rtems/score/states.h – with the definition of STATES_BEING_DELETED
- cpukit/score/include/rtems/score/stack.h – with the definition of variables (including watermarks) used in the stack checking mechanism

- cpukit/score/include/rtems/score/thread.h – definition of stack header and footer
- cpukit/score/src/**threaddispatch.c** – updates in the thread dispatch to check the stack bounds
- cpukit/score/src/threadstackfree.c – update to include the initial location of the stack, including the footer;
- cpukit/score/src/**threadinitialize.c** – update to the initialization of the stack, including the header and the footer
- cpukit/score/src/**threadstackallocate.c** - updates to allocate the stack, including the header and the footer
- cpukit/score/inline/rtems/score/stack.inl – implementation of the _Stack_Initialize_Header_and_Footer to initialize the header and footer of the stack

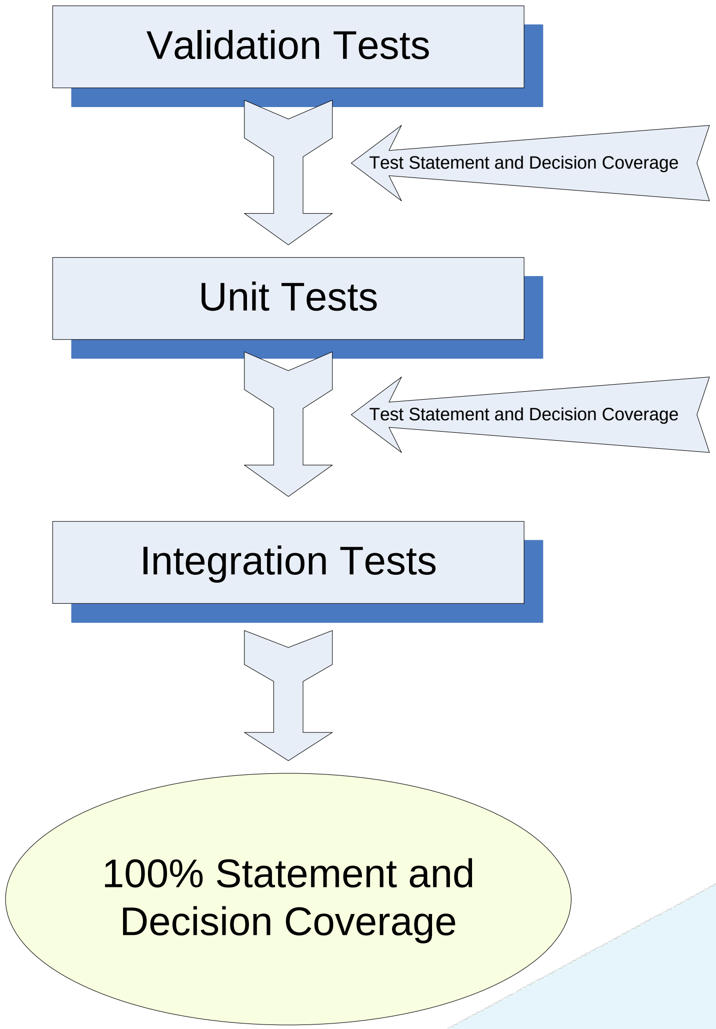
EDISOFT
DEFENCE & AEROSPACE TECHNOLOGIES

SW-FMECA 19, 20 ,21, 22, 23, 24, 25 Architecture, Design and Implementation

Major Modified Components/Files

- cpukit/rtems/src/**tasksetpriority.c** – implementation of denial of set priority when a task has a semaphore with priority inheritance and ceiling
- cpukit/rtems/src/**tasksuspend.c** - implementation of denial of suspending a task when a task has a semaphore with priority inheritance and ceiling
- cpukit/rtems/src/**taskmode.c** - implementation of denial of set task mode to non-preemptive when a task has a semaphore with priority inheritance and ceiling
- cpukit/rtems/src/**semcreate.c** - implementation of denial of creating a semaphore with different priority schemes

- cpukit/score/inline/rtems/score/**coremutex.inl** – implementation of _CORE_mutex_Is_ceiling_or_inherit function to analyse if a mutex is a priority ceiling or inheritance and updates in the _CORE_mutex_Seize due to semaphore release.


Validation Tests
Test Statement and Decision Coverage
Unit Tests
Test Statement and Decision Coverage
Integration Tests
100% Statement and Decision Coverage

Validation Test Specification

18 new validation tests

197 validation tests modified

13 validation tests substantially modified

Software Unit Plan

6 new unit tests

114 unit tests modified

2 unit tests substantially modified

Software Integration Plan

0 new integration tests

46 integration tests modified

1 unit test substantially modified

Generic Test Report – Validation

Review	Total number of planned tests	Total number of executed tests	Total number of successful tests	Total number of suspended tests	Total number of failed tests
RLU-FR	5364	5364	5364	0	0

Generic Test Report - Unit

Review	Total number of planned tests	Total number of executed tests	Total number of successful tests	Total number of suspended tests	Total number of failed tests
RLU-FR	2784	2784	2784	0	0

Generic Test Report – Integration

Review	Total number of planned tests	Total number of executed tests	Total number of successful tests	Total number of suspended tests	Total number of failed tests
RLU-FR	762	762	762	0	0

Generic Test Report

All Tests Passed

The statement coverage achieved 100%

The decision coverage achieved 100%

Software Budget Report

“Although it was detected an improvement in some RTEMS directives, **SWFMECA-8** introduces a significant **increase in memory occupancy** of applications and limits the **tasks stack to 8Kbytes**. The modifications made to RTEMS in SW-FMECA-8 introduced a **significant loss of product history**. The tasks are also obliged to use **CPU_HARDWARE_FP**. It was **not recommended** the introduction of SWFMECA-8 modifications in the RTEMS Improvement trunk.”

RTEMS Tailored & Testsuite

- **09060101-039-13.SFW**
  - SFW_SWFMECA_2_3_18 (Error Manager)
  - SFW_SWFMECA_5_6_7 (Rate Monotonic Deadline)
  - SFW_SWFMECA_17 (Stack Bounds Checker)
  - SFW_SWFMECA_19_…_25 (Interrupts/Semaphores)

- **09060101-028-14.testsuite**
  - SFW_SWFMECA_2_3_18 (Error Manager)
  - SFW_SWFMECA_5_6_7 (Rate Monotonic Deadline)
  - SFW_SWFMECA_17 (Stack Bounds Checker)
  - SFW_SWFMECA_19_…_25 (Interrupts/Semaphores)

Software Budget Report

"It can be verified that in a total of 392 of the measurements, **181 times RTEMS 4.8.0 was faster than RTEMS Tailored 13 and 211 times RTEMS Tailored 13 was faster than RTEMS 4.8.0.**

- RTEMS Tailored is faster in the **interrupts, context switch, IO, Task, Event, Rate Monotonic and Message Queue** operations;
- RTEMS 4.8.0 is faster in **Clock and Semaphore** operations

Software Product Assurance Report

Concerning **Functionality, the code is complete and correct** for all targets

The metrics related to **maintainability of the code (RTEMS and Test suite) are not fully compliant with the thresholds defined by GSWS.** However it should be **highlighted that only a small fraction (~<3%) of RTEMS has lower maintainability values and it was considered that the risk to improve these modules outcomes the benefits**

The metrics for **Requirements Stability, code comment frequency** and RIDs status demonstrate that the **documentation quality is good**

The **code can be considered reliable** as the values for structural coverage meet established targets

Software Product Assurance Report

It has been demonstrated that the **code is safe.**

The results of the milestone tracking demonstrate that the system engineering effectiveness process can be improved.

RTEMS LEON Upgrade

NEW GCC WITH RTEMS TAILORED

New GCC with RTEMS Tailored

“Based on the measurements and conclusions in the **CPU Occupancy, Timing Report and Memory Report** for the different toolchain, optimizations and hard-float flag, it was **recommend** the usage of RTEMS Improvement toolchain (**GCC 4.2.1** and **Binutils 2.18**) in the **development of RTEMS LEON Upgrade project.”**

RTEMS LEON Upgrade

CONCLUSIONS AND FUTURE WORK

Conclusions

New Features Introduced in RTEMS Improvement

- System-wide error manager/handler
- Rate Monotonic with Deadline
- Stack Bounds Checker
- Improvement of Semaphores with priority inheritance and ceiling and Interrupt Mask and Unmask

RTEMS build toolchain was Maintained

Lessons Learned

- It is essential to have **independent teams** for the realization of the **project** and for the **missions** support. Not having the independent teams have caused delays in the execution of the RTEMS LEON Upgrade project since team elements were shifted to the support.

DEFENCE & AEROSPACE TECHNOLOGIES

Conclusions

RTEMS Improvement Space Missions

- Galileo - FOC
- SmallGEO
- **MTG**
- **Solar Orbiter**
- Sentinel-2
- **Intermediate experimental Vehicle (IXV)**
- Earthcare

EDISOFT

DEFENCE & AEROSPACE TECHNOLOGIES

Future Work

Objective 1 – Maintain the support standards

- Study and **improve the delivery process** to cope with customers' demand to reduce releases time.

Objective 2 – Product Improvement to cope with new space missions requirements

- Develop and facilitate the qualification for **new device drivers**
- Integrate **new support platforms** in the RTEMS product
- **Development of tools** to support the validation and verification activities of the RTEMS space missions.

EDISOFT

DEFENCE & AEROSPACE TECHNOLOGIES

OBRIGADO / THANK YOU

Tel: +351 212 945 900
Fax: +351 212 945 999
info@rtemscentre.edisoft.pt

Rua Calvet Magalhães, 245
2770-153 Paço de Arcos · Portugal
www.edisoft.pt

A THALES Group Company