ADA EUROPE 2010

HRT-UML and Ada Ravenscar Profile: a Methodological Approach to the Design Of Level B Spacecraft Software

Roberto López, Ana Rodríguez

© GMV, 2010 Property of GMV
All rights reserved

INNOVATING SOLUTIONS

OVERVIEW

- Multinational conglomerate founded in 1984
- Private capital
- Subsidiaries in Spain, Portugal and USA
- Over 1000 employees all over the world
- Roots tied to the Space and Defense industries
- Currently operating in Aeronautics, Space, Defense, Security, Transportation, Healthcare and ITC industries.

MADRID – HEADQUARTERS

VALLADOLID

SEVILLA

BARCELONA

VALENCIA

CANARY ISLANDS

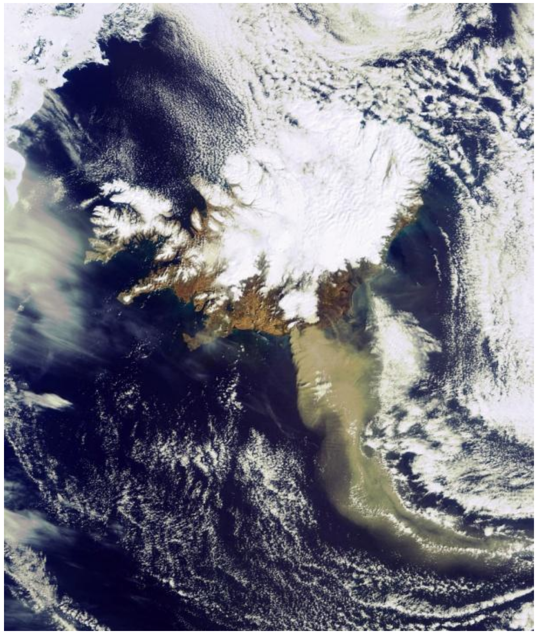
EEUU

PORTUGAL
GMV-SKYSOFT

POLAND
GMV Innovating Solutions Sp.zo.o

BACKGROUND

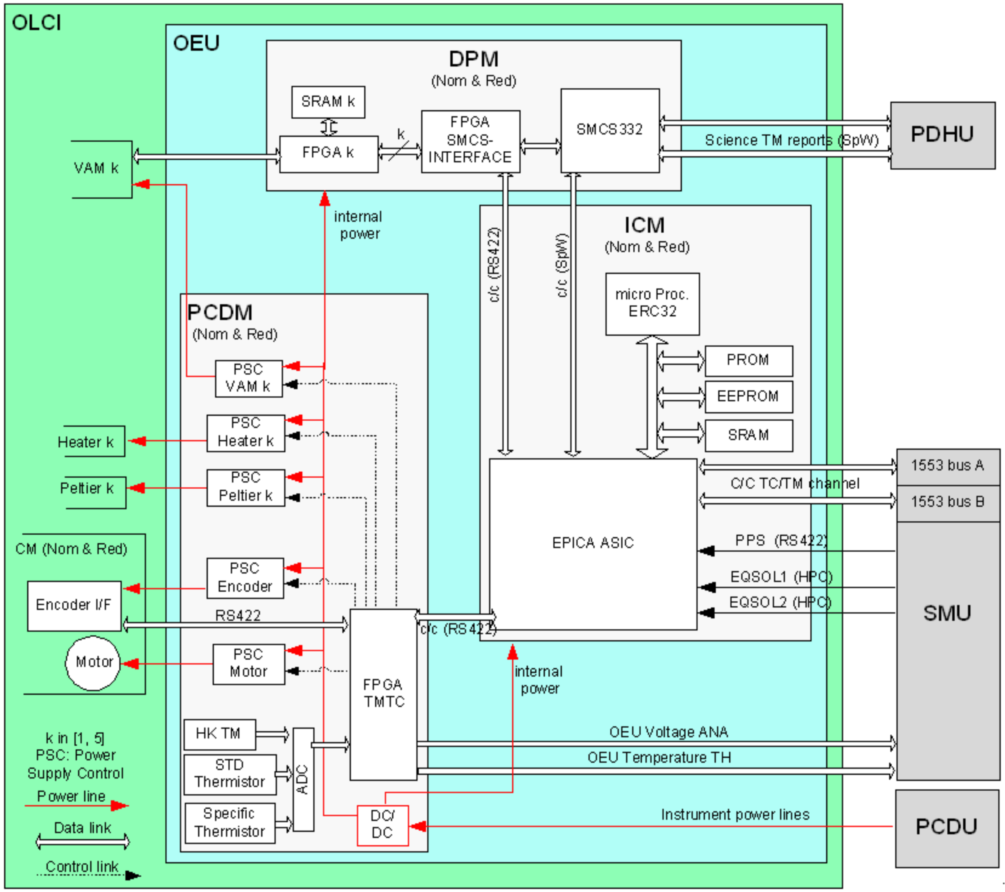
- Sentinel-3 continues the observation missions of ESA's ENVISAT and SPOT Satellites
  - Ocean & Land Colour Instrument (OLCI) continuation of ENVISAT/MERIS mission, launch on March 2002
  - GMV involved in the development of the OBSW for MERIS for ENVISAT
    - ESA PSS05, HOOD Methodology
    - 15K LOCS Ada
    - 144K LOCS Test Software
    - 136 Person/Month Efforts



19 April 2010 by ESA's ENVISAT satellite
The Eyjafjallajoekull Volcano in Iceland
http://www.esa.int/

BACKGROUND (2)

- Under Thales Alenia Space contract, GMV is responsible for the development of the ICM software
- Critical software ECCS-E-40B level B
- ERC32 microprocessor
- SW Development Environment:
  - Ada95,
  - AdaCore High Integrity Ravenscar Run Time for ERC32 (GNAT Pro for ERC32)

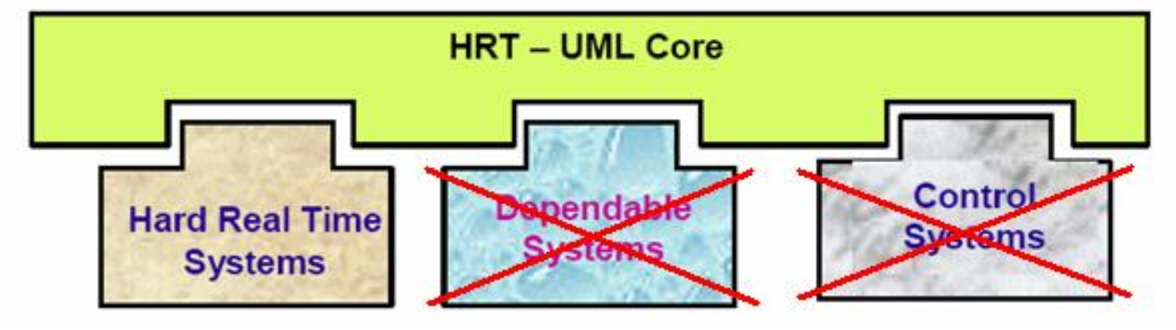


MOTIVATION

- An integrated solution (method + toolset) for the design of Ada95 OBSW systems compatible with ESA’s criticality categories
- Assessment of timing and performance requirements during the whole development lifecycle, as requested by ECSS-E-ST-40C:
  - Technical Budgets:
    - Memory size
    - CPU utilization
  - Schedulability analysis for real-time software
  - Behaviour modelling verification
- Possibility of accurate analysis of real-time behaviour by choice of scheduling/dispatching method together with suitable restrictions on the interactions allowed between tasks

HRT-UML

- The selected analysis and design methodology has been Object-Oriented supported by HRT-UML
  - Initial HRT-UML based on UML 1.4
  - A new evolution of the methodology has been developed in the context of ASSERT project, based on UML 2.0, but it is out of scope of this presentation
  - Customized version of UML expressing the HRT-HOOD methodology
- Extensions for Dependable Systems and Control Systems have not been used



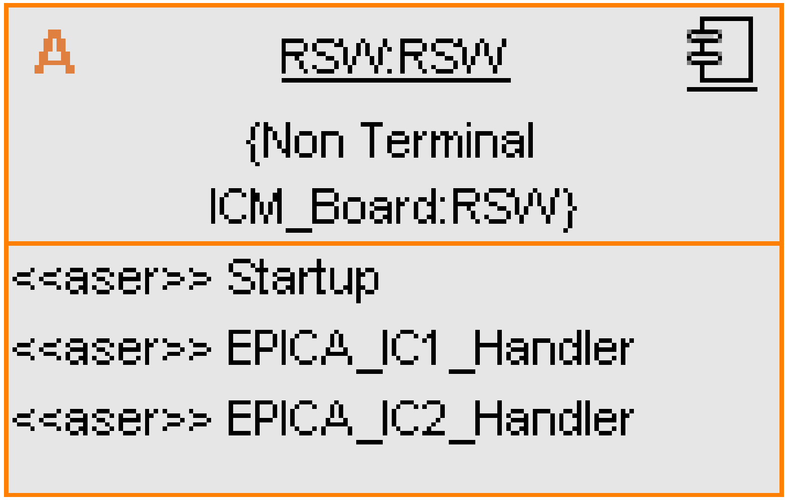
HRT-UML (2)

- HRT-UML main features:
  - Easy decomposition of the software architecture into design objects having internal parts that communicate with each other and with the outside environment
  - Explicit recognition of the typical activities of real-time systems
  - Integration of appropriate scheduling paradigms with the design process
  - Explicit definition of the application timing requirements for each activity
  - Static verification of processor allocation, schedulability and timing analysis
  - Provided toolset by Intecs implements Utilization Test, Response Time Test and Hyperplane Test

RAVENSCAR PROFILE

- A subset of the Ada tasking model defining restrictions to reduce the full tasking model
- Allows potential verification techniques
  - information flow analysis,
  - schedulability analysis,
  - execution order analysis and
  - formal model checking.
- Scheduling model:
  - Pre-emptive fixed priority scheduling
  - Priority Ceiling Protocol to avoid unbounded priority inversions and deadlocks
  - Supports *cyclic* and *sporadic* activities, and the idea of *hard, soft, firm* and *non-critical* tasks

EXAMPLE: SENTINEL 3 OLCI ICM SW



RSW:RSW

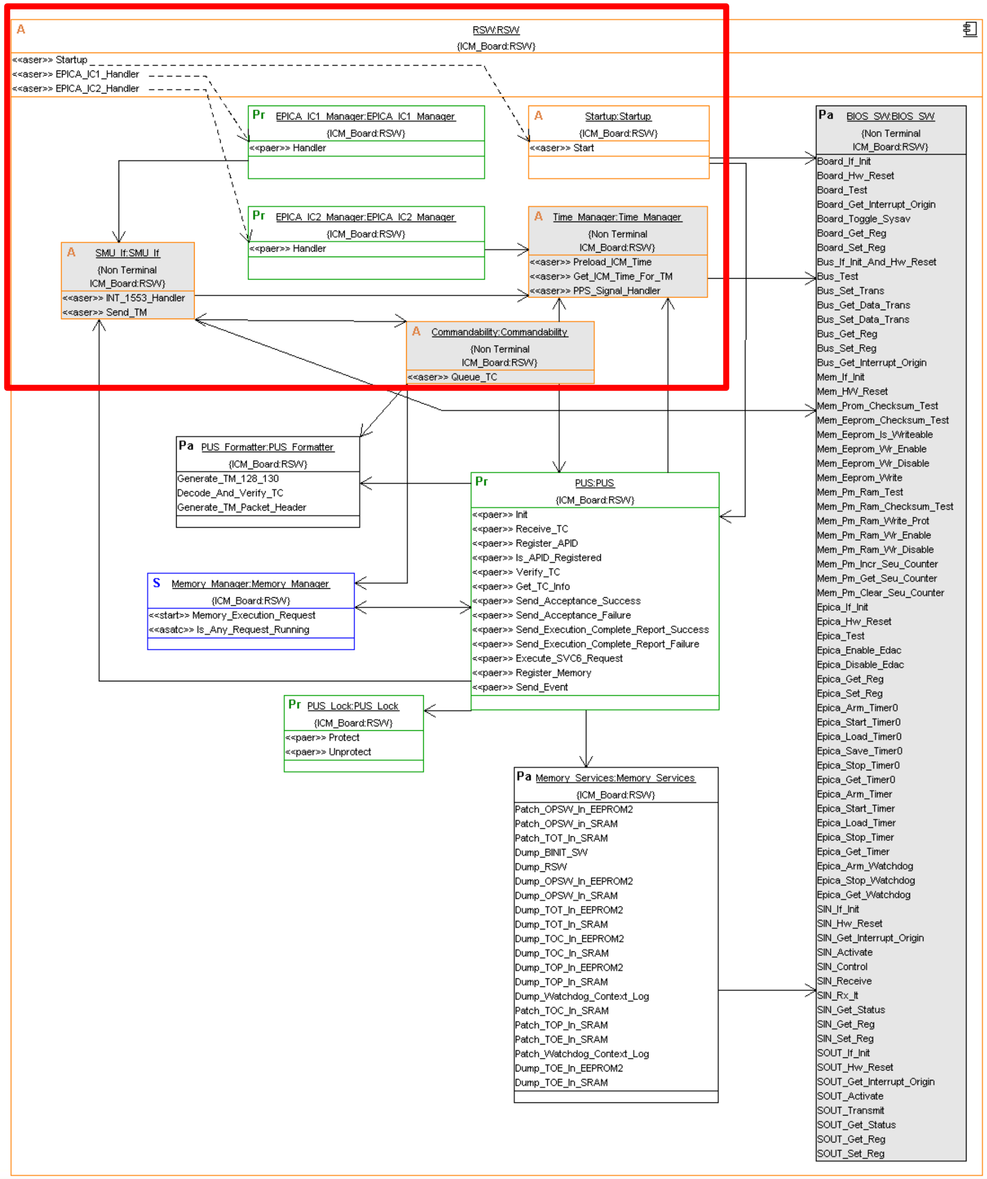
{Non Terminal ICM_Board:RSW}

<<aser>> Startup

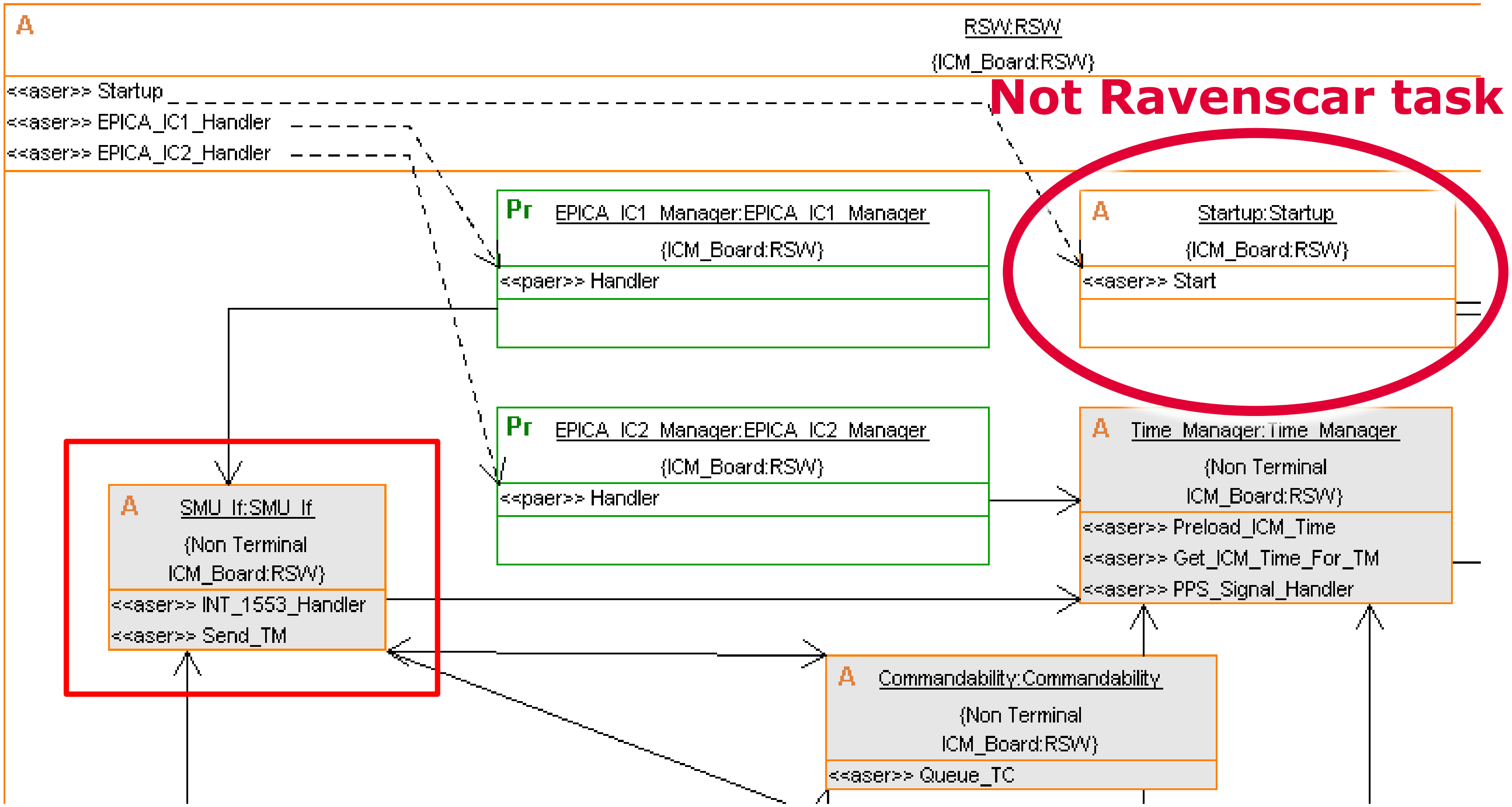
<<aser>> EPICA_IC1_Handler

<<aser>> EPICA_IC2_Handler

EXAMPLE: SENTINEL 3 OLCI ICM SW (2)



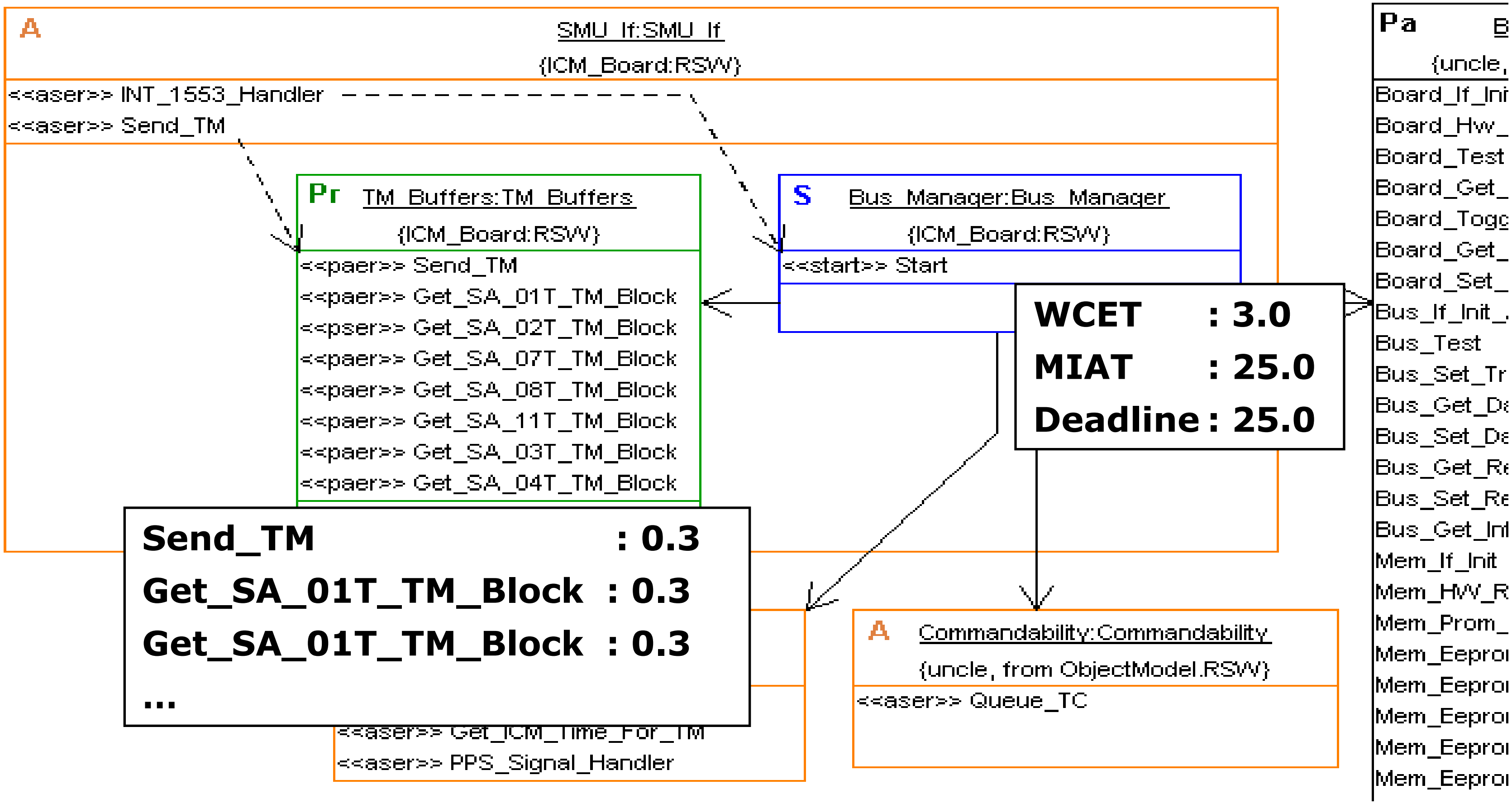
EXAMPLE: SENTINEL 3 OLCI ICM SW (3)


A
RSW:RSW
{ICM_Board:RSW}
<<aser>> Startup
<<aser>> EPICA_IC1_Handler
<<aser>> EPICA_IC2_Handler
Pr
EPICA_IC1_Manager:EPICA_IC1_Manager
{ICM_Board:RSW}
<<paer>> Handler
A
Startup:Startup
{ICM_Board:RSW}
<<aser>> Start
Pr
EPICA_IC2_Manager:EPICA_IC2_Manager
{ICM_Board:RSW}
<<paer>> Handler
A
Time_Manager:Time_Manager
{Non Terminal
ICM_Board:RSW}
<<aser>> Preload_ICM_Time
<<aser>> Get_ICM_Time_For_TM
<<aser>> PPS_Signal_Handler
A
SMU_If:SMU_If
{Non Terminal
ICM_Board:RSW}
<<aser>> INT_1553_Handler
<<aser>> Send_TM
A
Commandability:Commandability
{Non Terminal
ICM_Board:RSW}
<<aser>> Queue_TC

Not Ravenscar task

- It is possible to check Ravenscar compliance at model level
- HRT-UML rules are generally compliant with allowed Ravenscar features (e.g. no task allocators, no dynamic priorities, etc.)

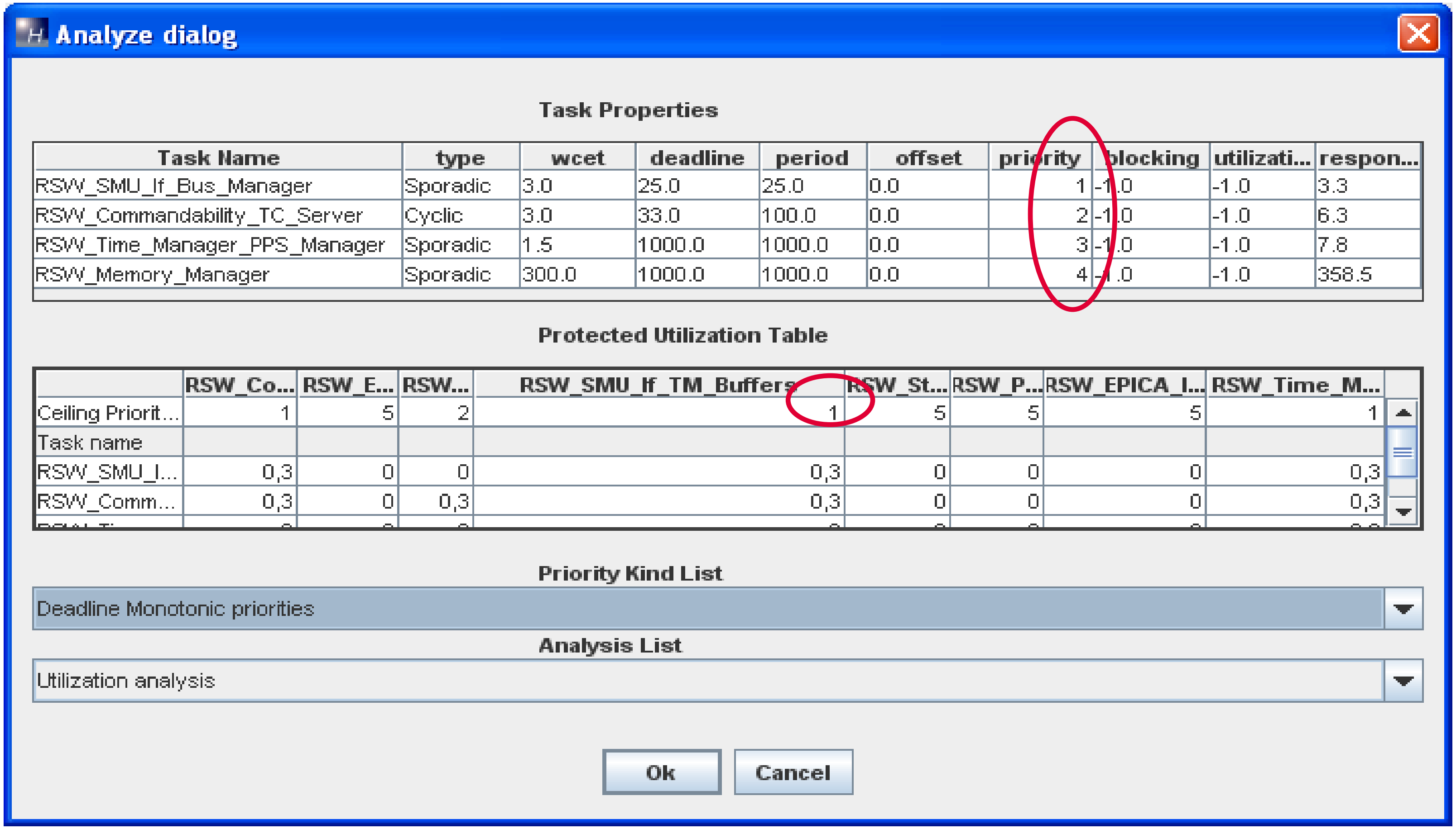
EXAMPLE: SENTINEL 3 OLCI ICM SW (4)



WCET : 3.0
MIAT : 25.0
Deadline : 25.0

Send_TM : 0.3
Get_SA_01T_TM_Block : 0.3
Get_SA_01T_TM_Block : 0.3
…

EXAMPLE: SENTINEL 3 OLCI ICM SW (5)



Task Properties

Task Name	type	wcet	deadline	period	offset	priority	blocking	utilizati...	respon...
RSW_SMU_If_Bus_Manager	Sporadic	3.0	25.0	25.0	0.0	1	-1.0	-1.0	3.3
RSW_Commandability_TC_Server	Cyclic	3.0	33.0	100.0	0.0	2	-1.0	-1.0	6.3
RSW_Time_Manager_PPS_Manager	Sporadic	1.5	1000.0	1000.0	0.0	3	-1.0	-1.0	7.8
RSW_Memory_Manager	Sporadic	300.0	1000.0	1000.0	0.0	4	-1.0	-1.0	358.5

Protected Utilization Table

	RSW_Co...	RSW_E...	RSW...	RSW_SMU_If_TM_Buffers	RSW_St...	RSW_P...	RSW_EPICA_I...	RSW_Time_M...
Ceiling Priorit...	1	5	2	1	5	5	5	1
Task name								
RSW_SMU_I...	0,3	0	0	0,3	0	0	0	0,3
RSW_Comm...	0,3	0	0,3	0,3	0	0	0	0,3

Priority Kind List

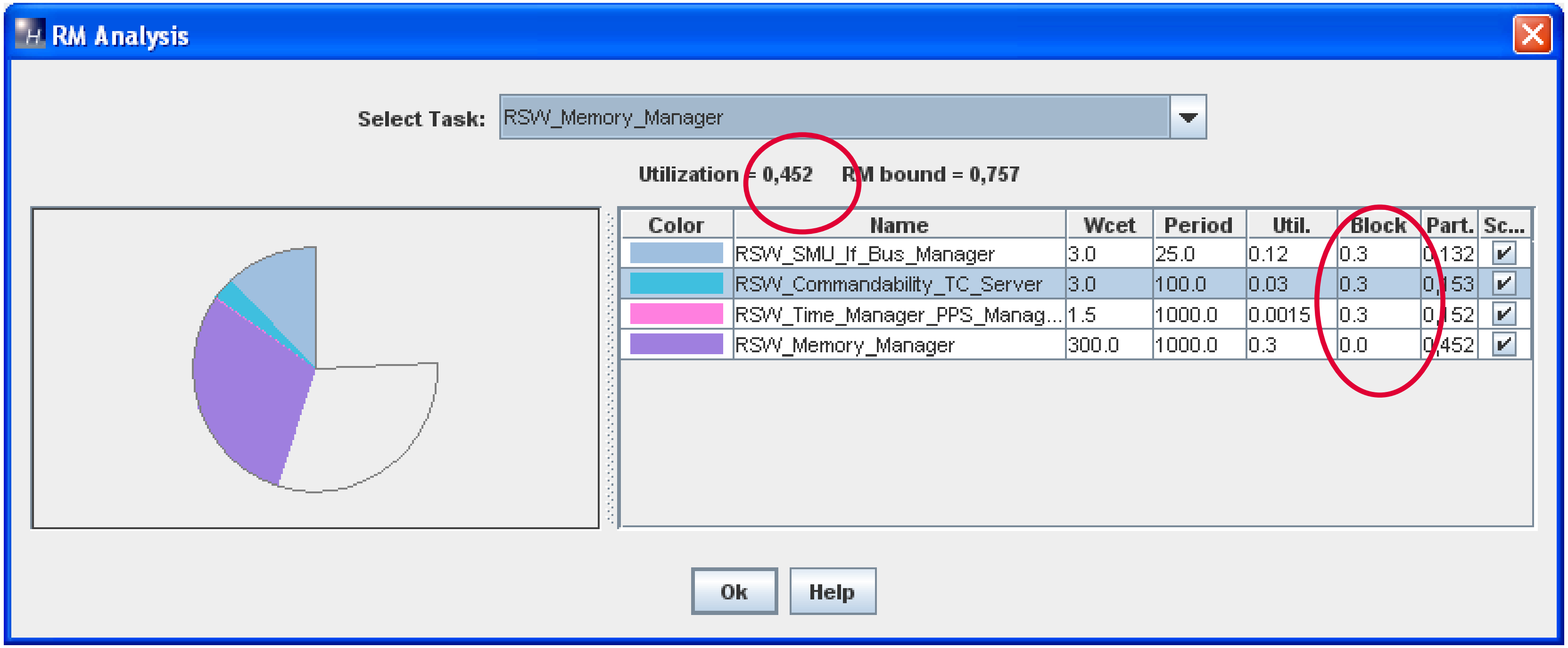
Deadline Monotonic priorities

Analysis List

Utilization analysis

- Priority assignment to tasks and protected objects according to Rate or Deadline Monotonic methods.

EXAMPLE: SENTINEL 3 OLCI ICM SW (6)



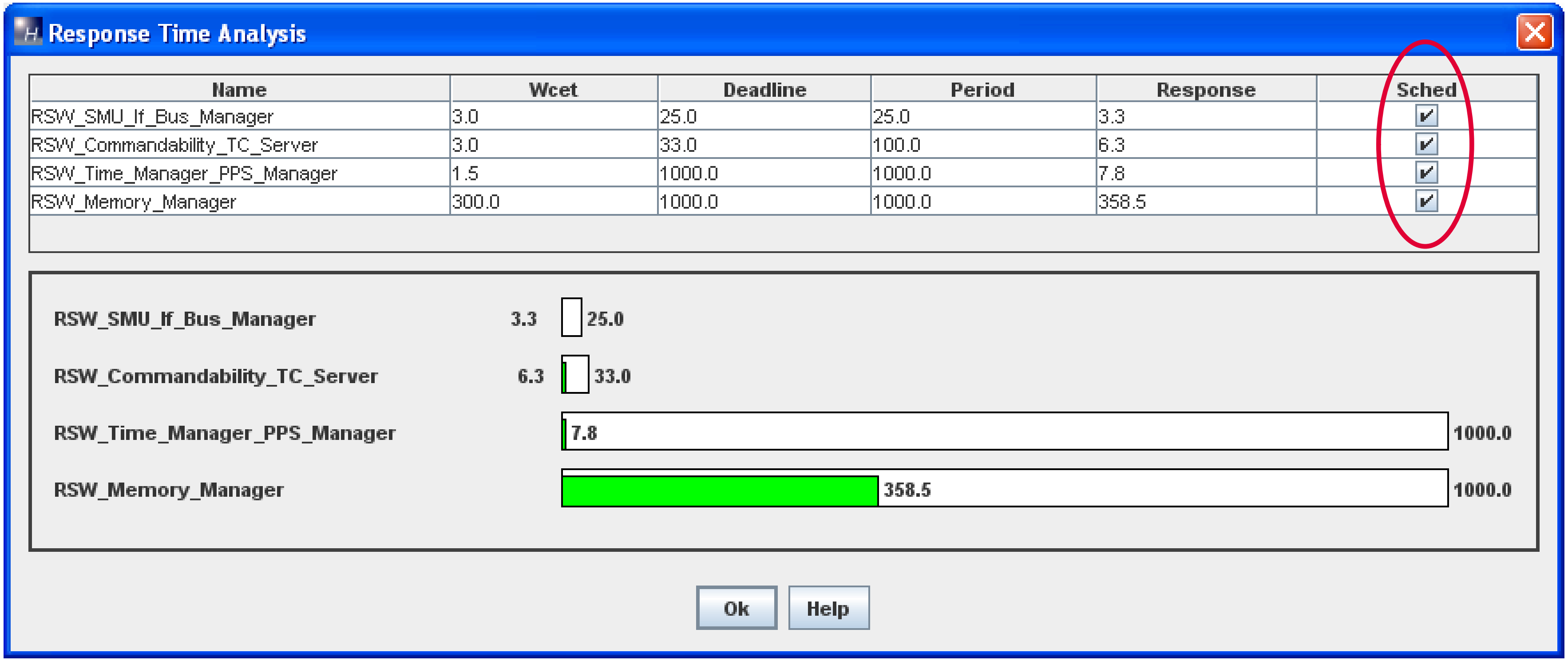
Select Task: RSW_Memory_Manager

Utilization = 0,452   RM bound = 0,757

Color	Name	Wcet	Period	Util.	Block	Part.	Sc...
■	RSW_SMU_If_Bus_Manager	3.0	25.0	0.12	0.3	0,132	☑
■	RSW_Commandability_TC_Server	3.0	100.0	0.03	0.3	0,153	☑
■	RSW_Time_Manager_PPS_Manag...	1.5	1000.0	0.0015	0.3	0,152	☑
■	RSW_Memory_Manager	300.0	1000.0	0.3	0.0	0,452	☑

- Checking CPU utilization

EXAMPLE: SENTINEL 3 OLCI ICM SW (7)

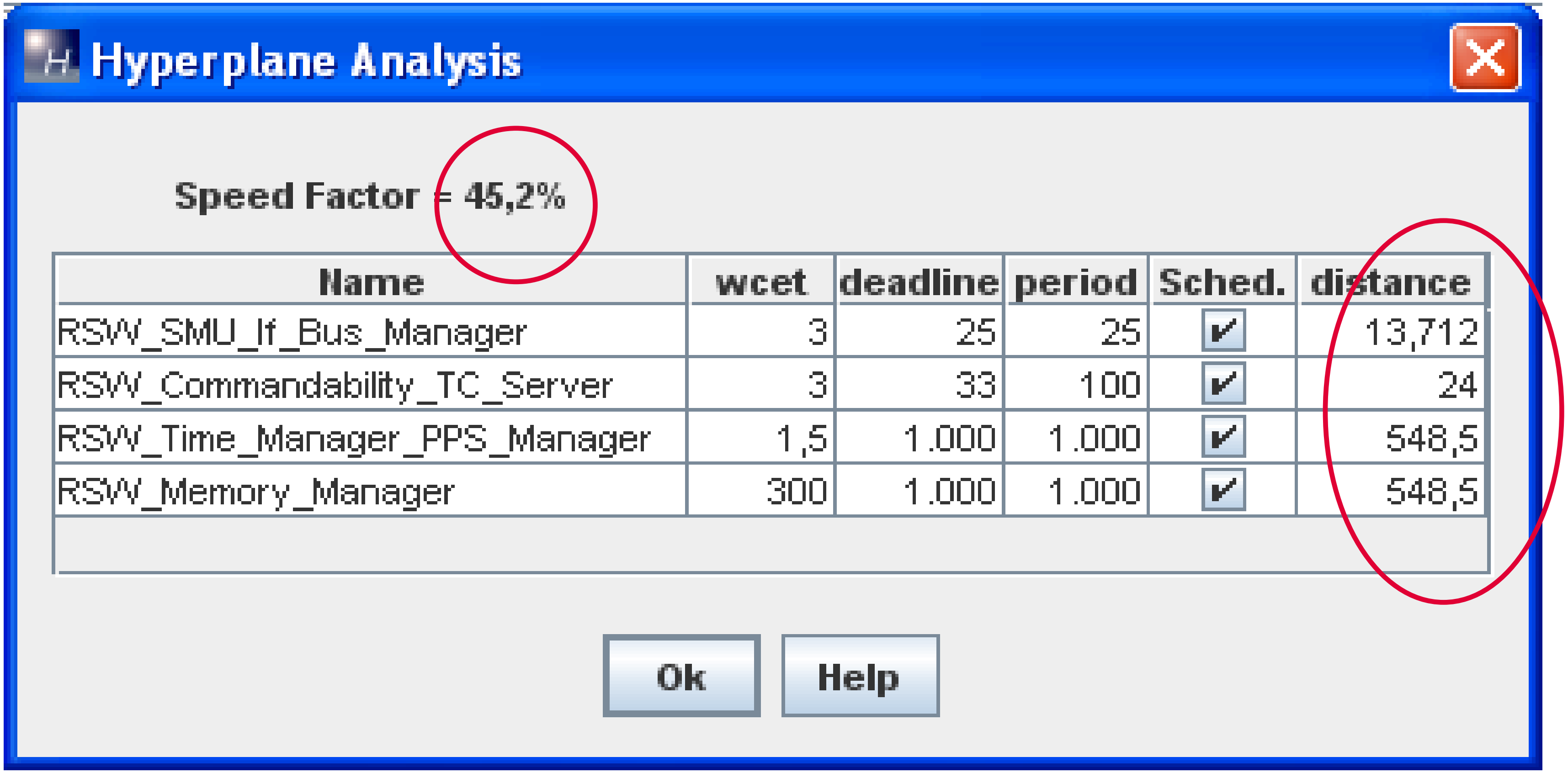


Response Time Analysis

Name	Wcet	Deadline	Period	Response	Sched
RSW_SMU_If_Bus_Manager	3.0	25.0	25.0	3.3	☑
RSW_Commandability_TC_Server	3.0	33.0	100.0	6.3	☑
RSW_Time_Manager_PPS_Manager	1.5	1000.0	1000.0	7.8	☑
RSW_Memory_Manager	300.0	1000.0	1000.0	358.5	☑

- Checking system schedulability according to Response Time Test

EXAMPLE: SENTINEL 3 OLCI ICM SW (8)

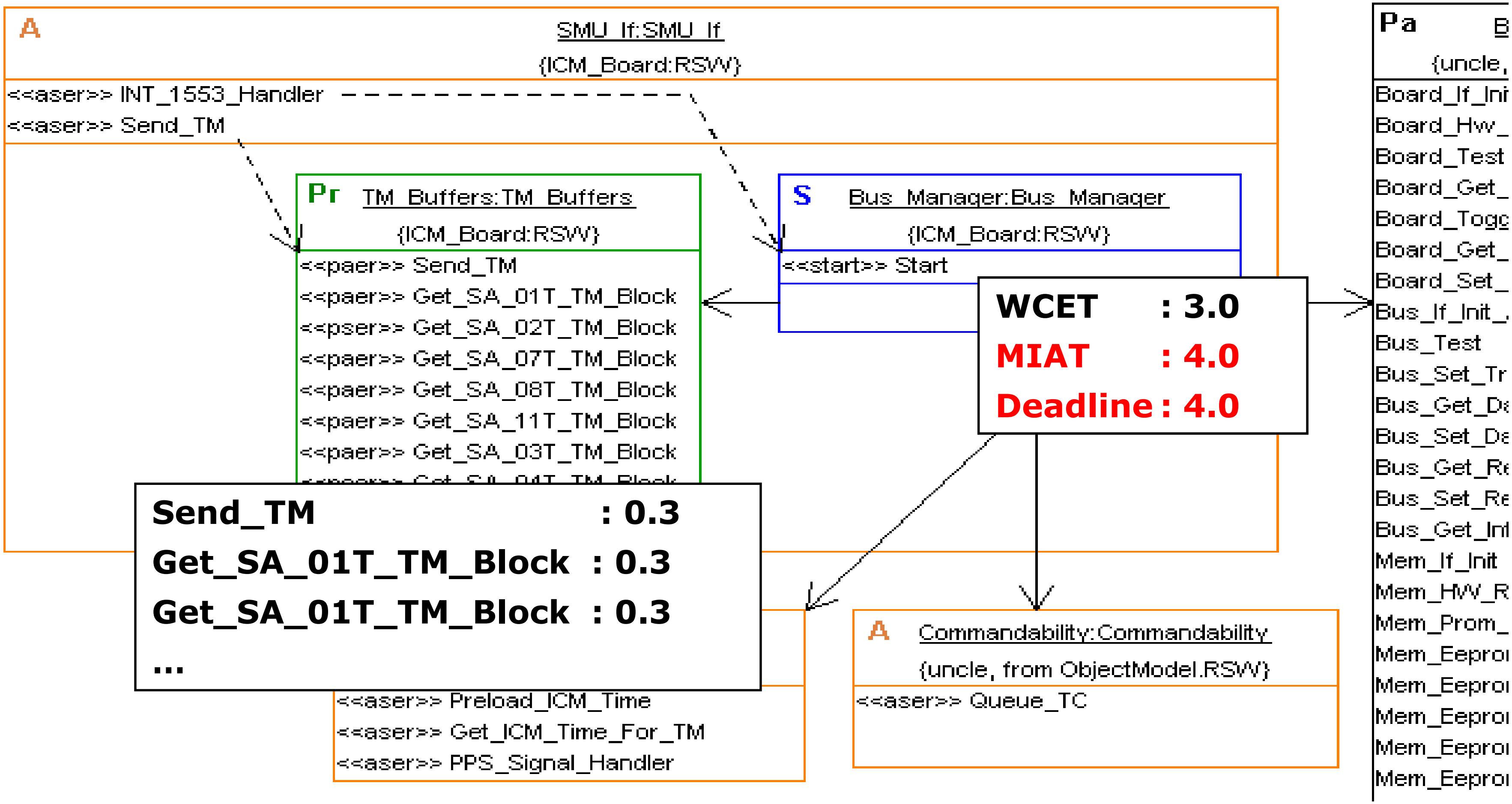


Speed Factor = 45,2%

Name	wcet	deadline	period	Sched.	distance
RSW_SMU_If_Bus_Manager	3	25	25	☑	13,712
RSW_Commandability_TC_Server	3	33	100	☑	24
RSW_Time_Manager_PPS_Manager	1,5	1.000	1.000	☑	548,5
RSW_Memory_Manager	300	1.000	1.000	☑	548,5

- Hyperplane analysis to check system schedulability, distance and speed factor.

EXAMPLE: SENTINEL 3 OLCI ICM SW (9)



A SMU_If:SMU_If
{ICM_Board:RSW}
<<aser>> INT_1553_Handler
<<aser>> Send_TM

Pr TM_Buffers:TM_Buffers
{ICM_Board:RSW}
<<paer>> Send_TM
<<paer>> Get_SA_01T_TM_Block
<<pser>> Get_SA_02T_TM_Block
<<paer>> Get_SA_07T_TM_Block
<<paer>> Get_SA_08T_TM_Block
<<paer>> Get_SA_11T_TM_Block
<<paer>> Get_SA_03T_TM_Block
<<paer>> Get_SA_04T_TM_Block

S Bus_Manager:Bus_Manager
{ICM_Board:RSW}
<<start>> Start

WCET : 3.0
MIAT : 4.0
Deadline : 4.0

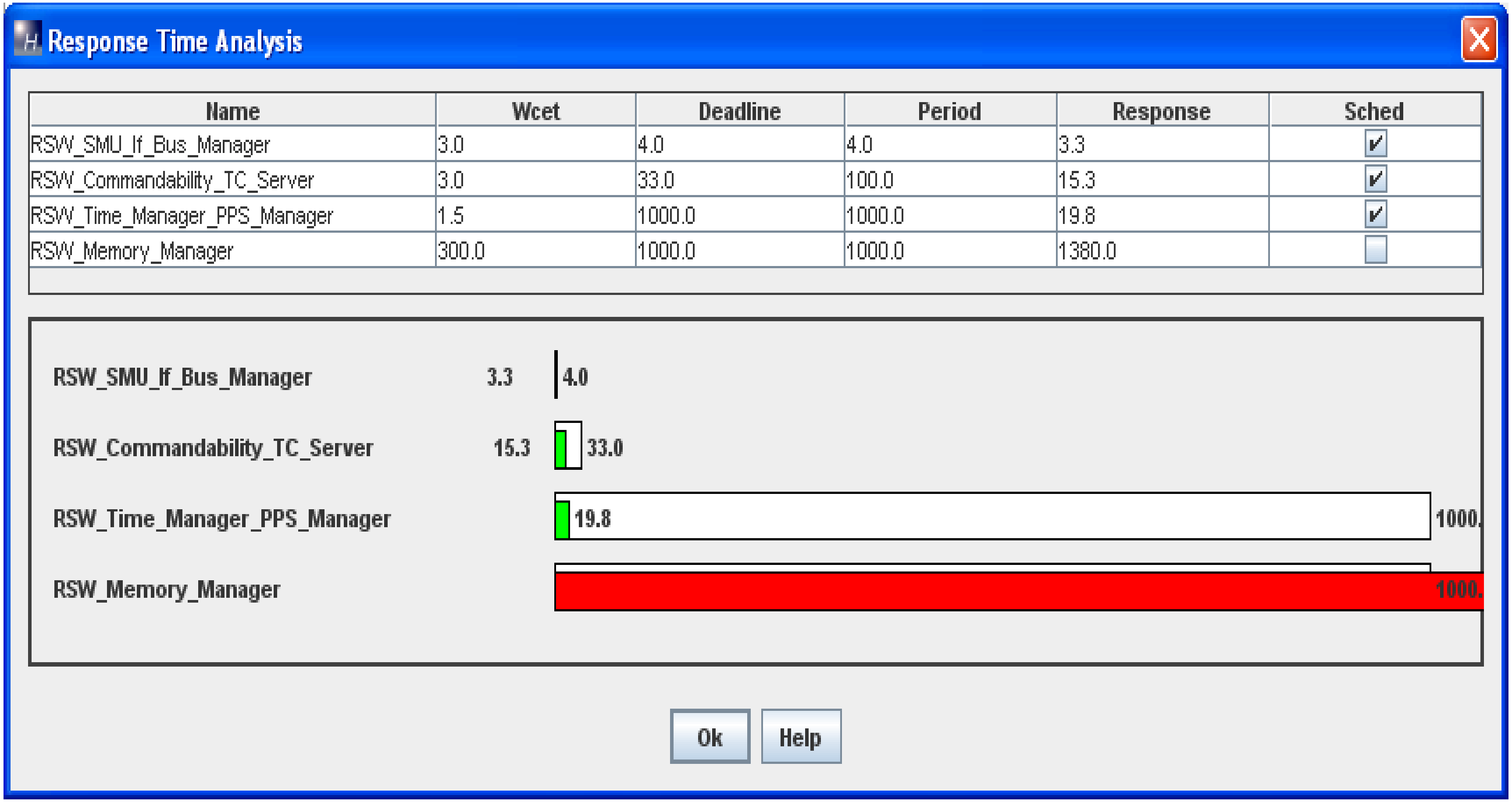
Send_TM : 0.3
Get_SA_01T_TM_Block : 0.3
Get_SA_01T_TM_Block : 0.3
...

<<aser>> Preload_ICM_Time
<<aser>> Get_ICM_Time_For_TM
<<aser>> PPS_Signal_Handler

A Commandability:Commandability
{uncle, from ObjectModel.RSW}
<<aser>> Queue_TC

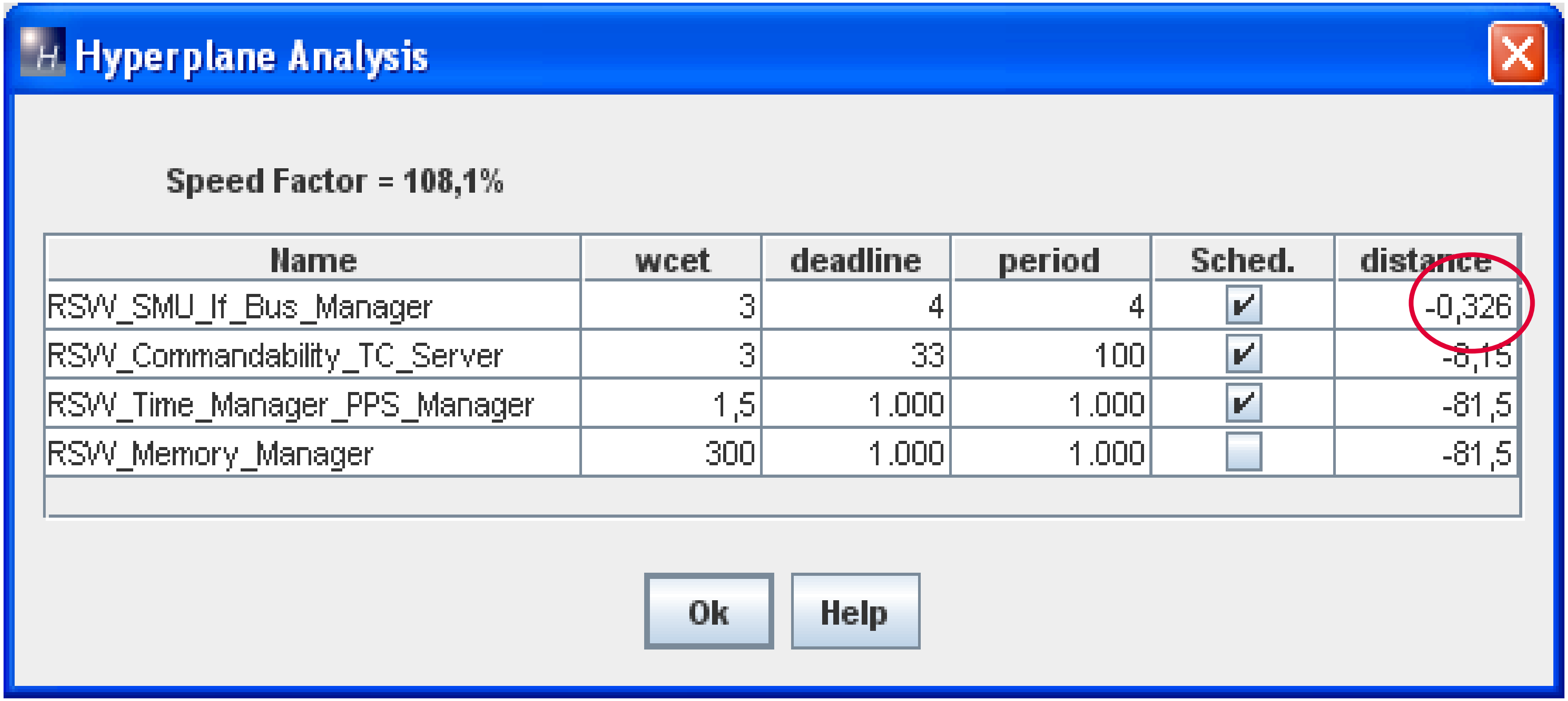
Pa B
{uncle,
Board_If_Ini
Board_Hw_
Board_Test
Board_Get_
Board_Togg
Board_Get_
Board_Set_
Bus_If_Init_
Bus_Test
Bus_Set_Tr
Bus_Get_Da
Bus_Set_Da
Bus_Get_Re
Bus_Set_Re
Bus_Get_Int
Mem_If_Init
Mem_HW_R
Mem_Prom_
Mem_Eeprom
Mem_Eeprom
Mem_Eeprom
Mem_Eeprom
Mem_Eeprom

EXAMPLE: SENTINEL 3 OLCI ICM SW (10)



Name	Wcet	Deadline	Period	Response	Sched
RSW_SMU_If_Bus_Manager	3.0	4.0	4.0	3.3	☑
RSW_Commandability_TC_Server	3.0	33.0	100.0	15.3	☑
RSW_Time_Manager_PPS_Manager	1.5	1000.0	1000.0	19.8	☑
RSW_Memory_Manager	300.0	1000.0	1000.0	1380.0	□

EXAMPLE: SENTINEL 3 OLCI ICM SW (11)

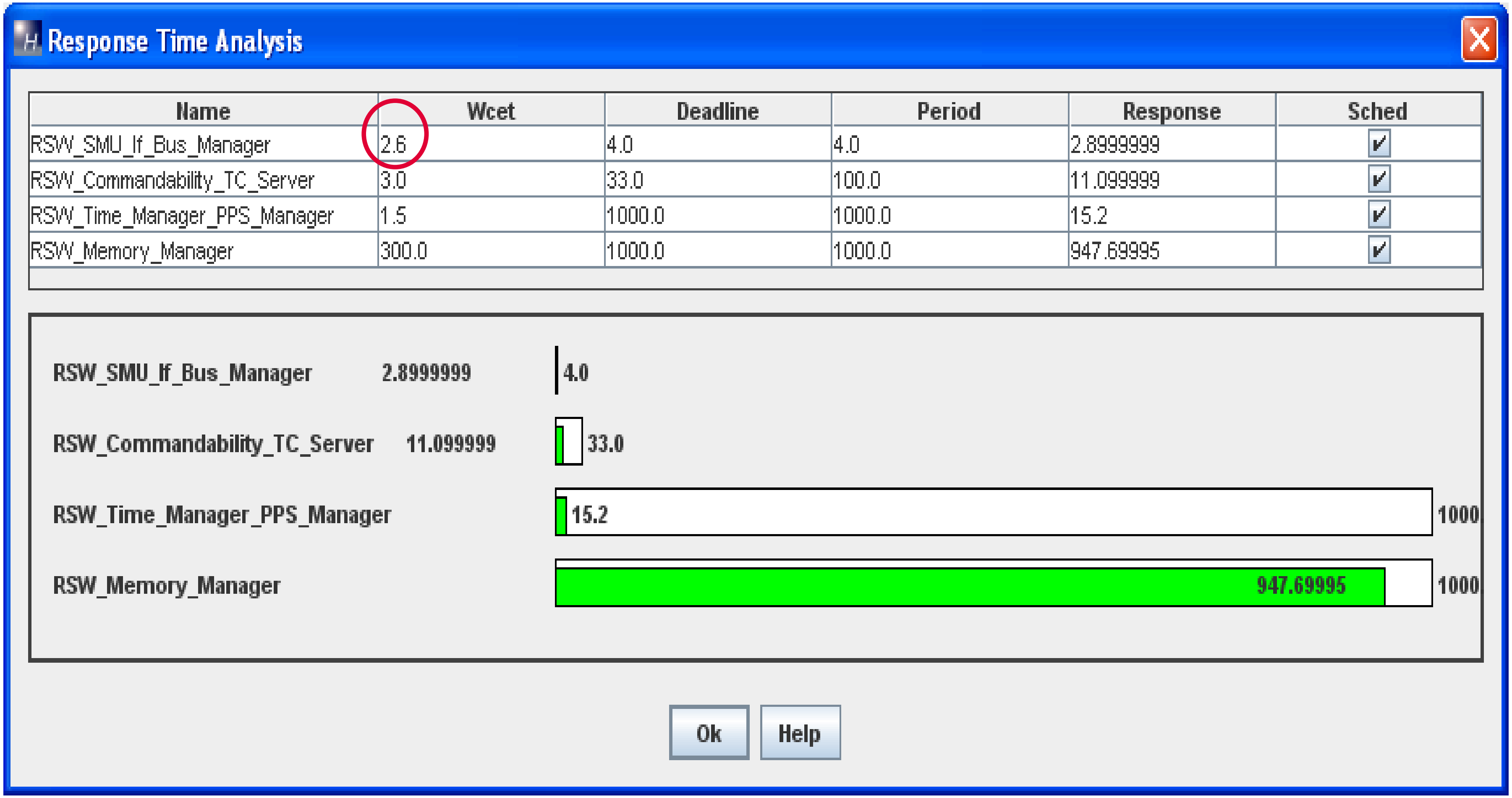


Hyperplane Analysis

Speed Factor = 108,1%

Name	wcet	deadline	period	Sched.	distance
RSW_SMU_If_Bus_Manager	3	4	4	☑	-0,326
RSW_Commandability_TC_Server	3	33	100	☑	-8,15
RSW_Time_Manager_PPS_Manager	1,5	1.000	1.000	☑	-81,5
RSW_Memory_Manager	300	1.000	1.000	□	-81,5

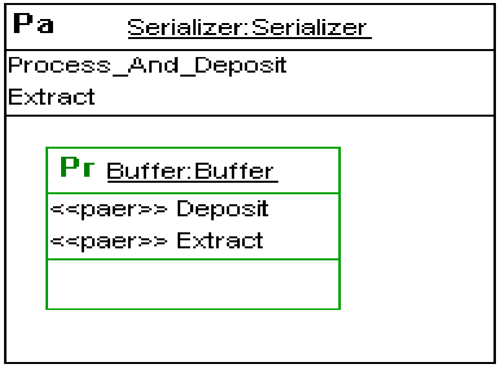
EXAMPLE: SENTINEL 3 OLCI ICM SW(12)



Name	Wcet	Deadline	Period	Response	Sched
RSW_SMU_If_Bus_Manager	2.6	4.0	4.0	2.8999999	☑
RSW_Commandability_TC_Server	3.0	33.0	100.0	11.099999	☑
RSW_Time_Manager_PPS_Manager	1.5	1000.0	1000.0	15.2	☑
RSW_Memory_Manager	300.0	1000.0	1000.0	947.69995	☑

DIFFICULTIES

- HRT-UML model restrictions to comply Ravenscar Profile seem to be more restrictive than the profile itself.
- HRT-UML classes and Data types
  - Objects are instances of Classes
  - Attributes are instances of Data Types. Arguments of operations are also based on Data Types
  - Data Type and UML class concept are the same: Not completely necessary to have Data Types, could be replaced by Passive classes



  - Implications from schedulability point of view and HRT-UML consistency needs to investigated

RAVENSCAR RUN-TIME KERNEL

- Apart from specific Ravenscar kernels on ERC32:
  - GNAT Pro (Adacore): Being qualified for ECCS-E40-B level-B.
  - ERC32 Ada (XGC)
  - ObjectAda RAVEN (Aonix)
  - GNAT/ORK
- Same approach can be achieved using ESA's RTEMS:
  - Possible to reproduce Ravenscar restrictions on top on RTEMS
  - RTEMS provides the same Ravenscar scheduling model: pre-emptive fixed priority scheduling and priority ceiling protocol when accessing to shared sections
  - RTEMS Ada API is not used due to:
    - OAR has stopped to support ADA for RTEMS
    - The only qualified API (under level B qualification process, performed by RTEMS Center) is the RTEMS Classic API
  - Solution based on using GNAT Pro for ERC32 compiler using Zero-Foot-Print run time system on top RTEMS Classic API

CONCLUSIONS

- Covered expectations
  - An integrated solution (method + toolset) for the design of embedded real-time systems √
  - Assessment of timing and performance requirements during the whole development lifecycle, as requested by ECSS-E-ST-40C
    - Technical Budgets :
      - Memory size, **X (manually done)**
      - CPU utilization and √
    - Schedulability analysis for real-time software √
    - Behaviour modelling verification **Partially**
  - Possibility of accurate analysis of real-time behaviour by choice of scheduling/dispatching method together with suitable restrictions on the interactions allowed between tasks √

Thank you

Ana Isabel Rodríguez
airodriguez@gmv.com
www.gmv.com

INNOVATING SOLUTIONS