MARTE/AADL support

Demonstration

Madeleine FAUGERE Ph.D,
Laurent RIOUX Ph.D
THALES

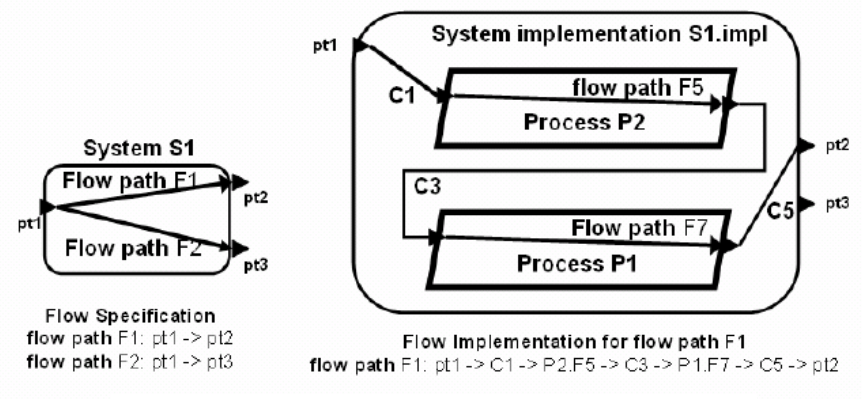
Copyright THALES – Copyright CEA – Copyright INRIA

AADL Architecture Analysis Description Language

- Architecture Description Language dedicated to RTES
- International standard at SAE (AS5506, 2004)
- Adapted for many critical computer system domain
  - Automotive, space, robotics, industrial control, medical, avionics, …
- Allow specification, analysis and automated integration of real-time performance critical
  - Timing,
  - Safety,
  - Schedulability,
  - Fault tolerant,
  - Security,
  - Distributed computing systems,….

Copyright THALES – Copyright CEA – Copyright INRIA

What does AADL look like ?



A graphical representation

```
  compute12: data port compute1.outed ->> compute2.ined;
  compute23: data port compute2.outed -> compute3.ined;
  actuateconn: data port compute3.outed ->> actuate.ined;
  bus access db -> sense.devbus;
  bus access db -> actuate.devbus;
flows
  etelatency: end to end flow sense.flow1 -> senseconn -> com-
pute1.flow1
            -> compute12 -> compute2.flow1 -> compute23 -> com-
pute3.flow1
            -> actuateconn -> actuate.flow1 { latency => 153 ms;};
end application.twosamplesteps;
```

A textual representation

Copyright THALES – Copyright CEA – Copyright INRIA

AADL useful for….

AADL Elements

Software components

Hardware component

Allocation

Port and connections

Flows

Modes

Properties (extensible language)

Non functional aspects through properties specification

- Temporal, safety, reliability,…

Architecture and design

- Components and interfaces, connections
- Data and control flows
- Run time architecture

Analysis and verification aspects

- End-to-end latency
- Age of signal data and jitter
- …

Copyright THALES – Copyright CEA – Copyright INRIA

MARTE shall be generic

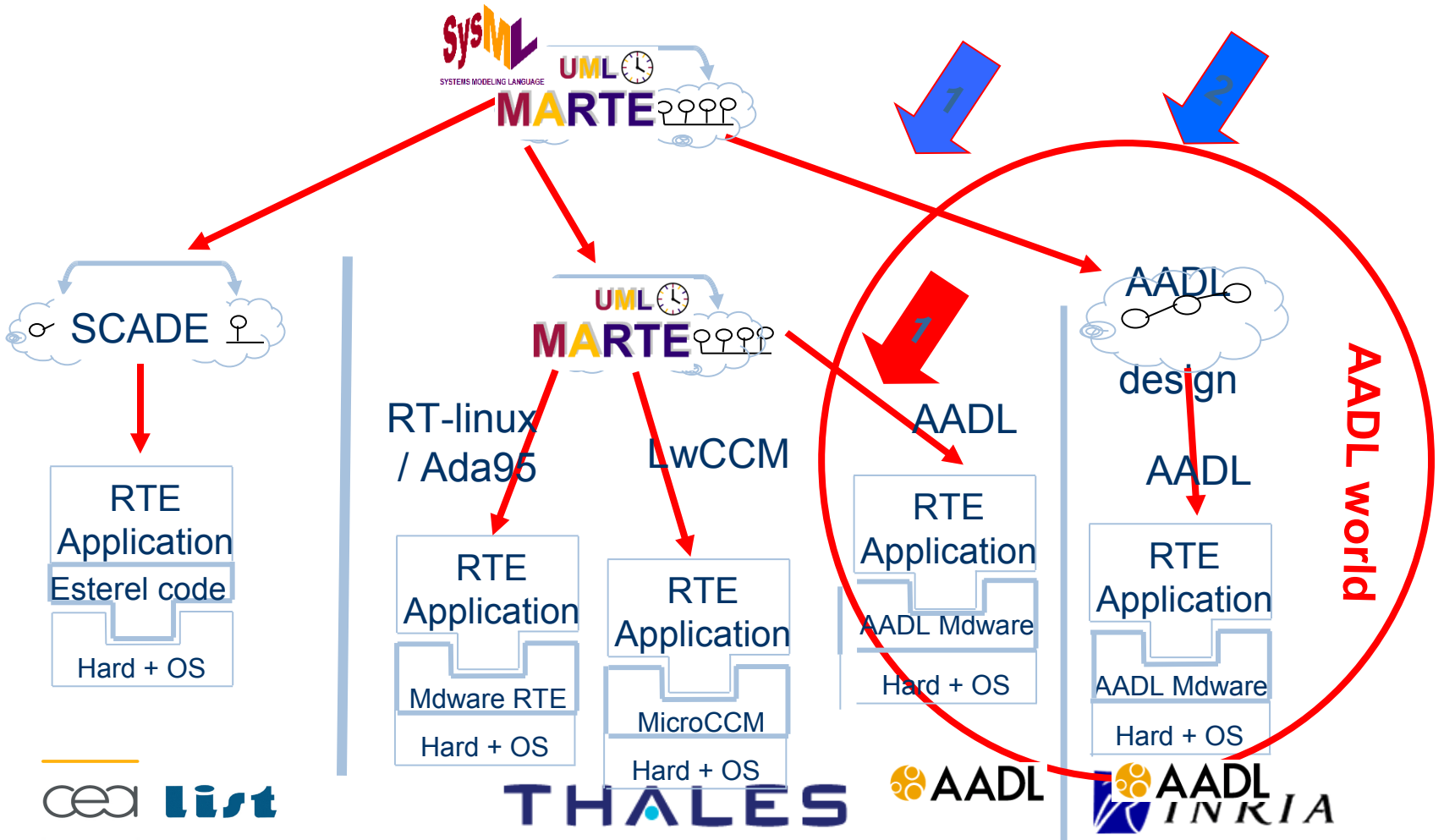
- "The UML profile for MARTE addresses modeling and analysis of real-time and embedded systems, including their software and hardware aspects" :
  - Provides support for non-functional property modeling
  - Adds rich time and resource models to UML
  - Defines concepts for software and hardware platform modeling
  - Defines concepts for allocation of applications on platforms
  - Provides support for quantitative analysis (e.g. scheduling, performance)

Copyright THALES – Copyright CEA – Copyright INRIA

AADL guidelines sense of being

Allow UML/MARTE users to model AADL applications

- Consolidate MARTE on the “Unified” aspects
- Let end-users choose RT language and verification/validation tools at different development cycle stages



Copyright THALES – Copyright CEA – Copyright INRIA

MARTE-AADL concepts mapping

- First AADL – MARTE alignment based on AADL (v1) constructs and features and MARTE artifacts.

AADL concepts	MARTE concepts
Software components	memoryPartition, swSchedulableRessource,.
Hardware components	hwProcessor, hwMer
Binding	Allocated
AADL features	MARTE flowPorts, UML required/provided interfaces…
Subcomponents	UML Parts
Port Connections	UML delegation/assembly
Flow specification	UML object flows
Modes	UML state machines
Properties	Not yet documented

- Next step: Deeper map AADL component semantics, AADL properties and implicit AADL platform semantics on MARTE concepts (MARTE concepts properties and NFP, VSL language)

Copyright THALES – Copyright CEA – Copyright INRIA

MARTE-AADL code generator

- MARTE to AADL code generator is already available
- Bridge between MARTE/AADL and Cheddar (Scheduling analysis) already tested

```
process implementation Nav_Control_Process.PowerPC_G4
subcomponents
        T_GPS_Reader : thread GPS_Sampling_Thread.PowerPC_G4 in modes (GPS_UP_AP_UP, GPS_UP_AP_DOWN);
        T_AP_Compute : thread Autopilot_Compute_Thread.PowerPC_G4 in modes (GPS_UP_AP_UP);  T_AP_Params : thread
Autopilot_Modify_Parameters_Thread.PowerPC_G4;
        D_AP_Destination : data Nav_Types::Position.Simple;
        D_AP_Airspeed : data Nav_Types::Integer;
        D_AP_Altitude : data Nav_Types::Integer;
connections
        data port GPS_Position_Input -> T_GPS_Reader.Position_Input in modes (GPS_UP_AP_UP, GPS_UP_AP_DOWN);
        data port T_GPS_Reader.Position_Output -> T_AP_Compute.Position_Input in modes (GPS_UP_AP_UP, GPS_UP_AP_DOWN);
        data port T_GPS_Reader.Position_Output -> T_AP_Compute.Position_Input in modes (GPS_UP_AP_UP);
        T_AP_Compute.Delta_Roll_Output -> Delta_Roll_Output in modes (GPS_UP_AP_UP);
        data port T_AP_Compute.Delta_Yaw_Output -> Delta_Yaw_Output in modes (GPS_UP_AP_UP);
        data port T_AP_Compute.Delta_Pitch_Output -> Delta_Pitch_Output in modes (GPS_UP_AP_UP);
        data port T_AP_Compute.Engine_RPM_Output -> Engine_RPM_Output in modes (GPS_UP_AP_UP);
        event data port AP_Position_Input -> T_AP_Params.AP_Position_Input;          modes
        GPS_UP_AP_DOWN : initial mode;
        GPS_UP_AP_UP : mode;
        <INITIAL_MODE> -[ <EVENT> ]-> <FINAL_MODE>
        GPS_UP_AP_DOWN -[ AP_Toggle ]-> GPS_UP_AP_UP;
        GPS_UP_AP_DOWN -[ GPS_Error ]-> GPS_DOWN;
        GPS_UP_AP_UP -[ GPS_Error ]-> GPS_DOWN;
end Nav_Control_Process.PowerPC_G4; …
…
process implementation HCI_Process.PowerPC_G4
subcomponents
        T_Screen_Disp : thread Screen_Display_Thread.PowerPC_G4;
        T_Pilot_Input : thread Pilot_Input_Thread.PowerPC_G4;
connections
        event port T_Pilot_Input.AP_Toggle -> AP_Toggle;
```

CODE GENERATED

Copyright THALES – Copyright CEA – Copyright INRIA

Conclusion

- SAE (AADL working Group)
  - Has chosen MARTE profile for designing AADL application in UML
  - Will contribute to the Guidelines Annex

- Guidelines consolidation (+ AADL v2 alignment)
- Remarks, corrections, precisions are welcome
  - MARTE issues must to be sent before 22 December 2007

- Examples from the AADL info website have been modeled with these guidelines
- Reused AADL tools (TopCased, Cheddar,…)

Copyright THALES – Copyright CEA – Copyright INRIA

Copyright THALES – Copyright CEA – Copyright INRIA

Copyright THALES – Copyright CEA – Copyright INRIA

AADL component Type and Implementation

AADL Component Types

- Specifies a functional interface in terms of features (ports, port groups, flow specifications, subcomponent access..), properties
- UML package containing AADL component declaration)

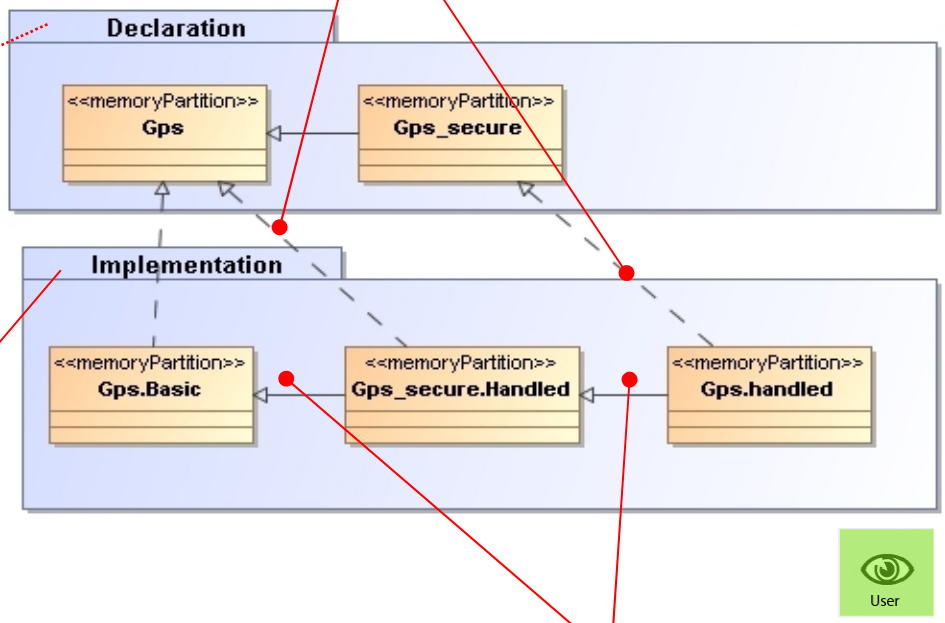
AADL Component Implementations

- describes the internal structure and behavior of that component in terms of subcomponents, connections and flows across them, and behavioral modes
- UML package containing AADL component implementations
- AADL implementation liked to AADL declaration through UML Realization

AADL Component Extension

- UML Generalization

Component Implementation link: UML Realization



Component Extension : UML Generalization

Copyright THALES – Copyright CEA – Copyright INRIA

AADL System and AADL Subcomponents

AADL System

- Represents a composite software, execution platform or system components
- Represented by a “SysML” block

AADL Subcomponents

- represented by UML parts

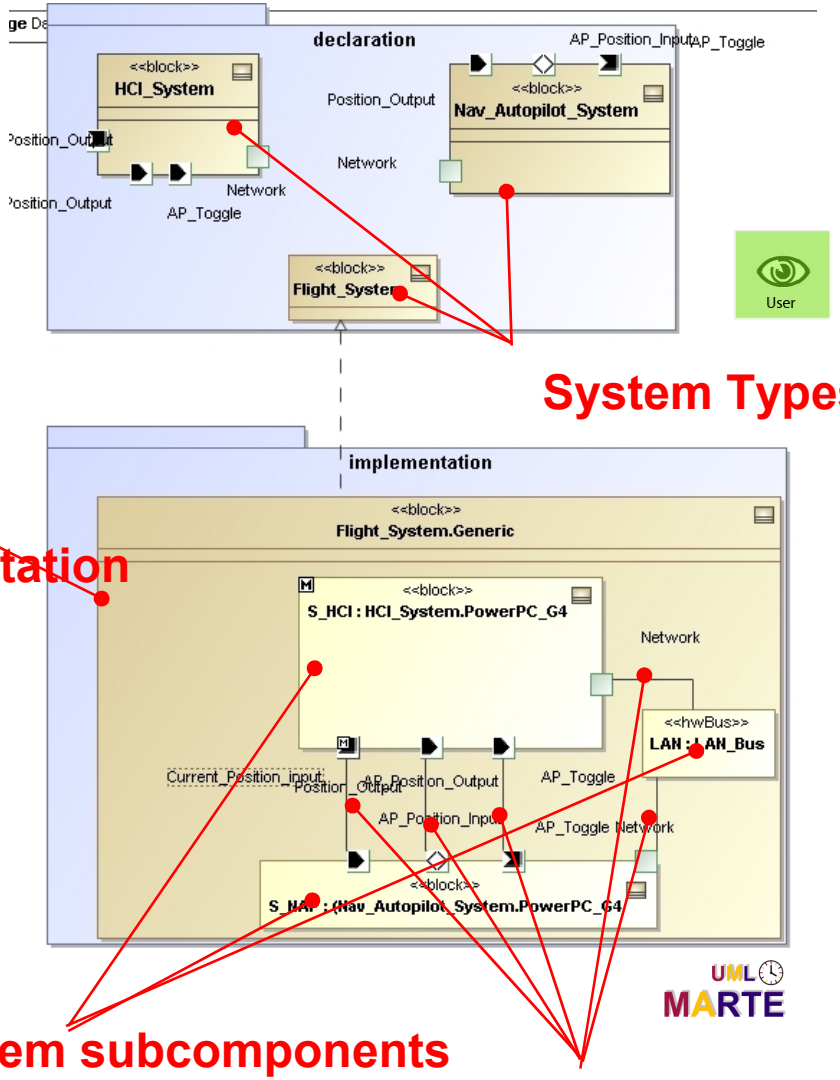
```
system Flight_System
end Flight_System;

system implementation Flight_System.Generic
subcomponents
        S_HCI : system HCI_System.PowerPC_G4;
        S_NAP : system Nav_Autopilot_System.PowerPC_G4;

        LAN : bus LAN_Bus;
```

AADL

Generated Code



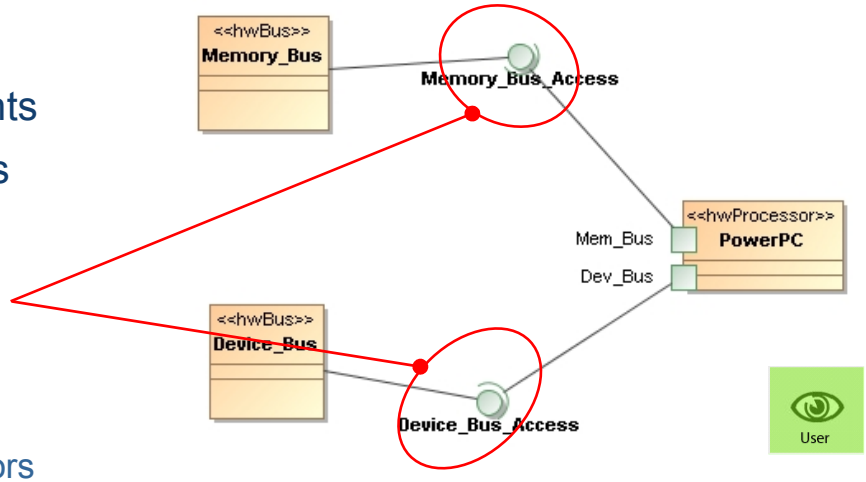
Copyright THALES – Copyright CEA – Copyright INRIA

Required data/bus access

AADL Bus and Data components access

- Provide data/service to other AADL components
- Access required from other AADL components

- Declaration part :
  - Provide/required access modeled in MARTE via provided / required UML Interfaces
- Implementation part :
  - access design by delegation / assembly connectors
  - Resources may be specialized with real time features or associated to services (synchronization, concurrency access …)



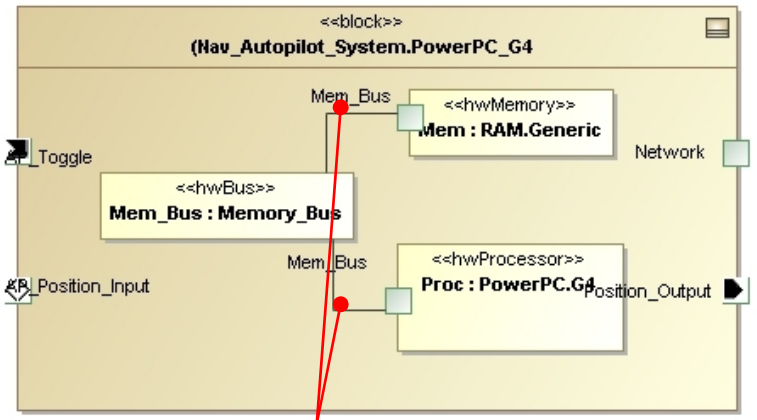
Access declaration

Generated Code

```
processor powerPC
Features
                MemBus : requires bus access Memory_Bus;
                Dev_Bus : requires bus access Device_Bus
End PowerPC;

bus Memory_Bus
end Memory_Bus;
```

AADL



Access connections

yright INRIA

cea list THALES INRIA UML MARTE

Ports and Connections

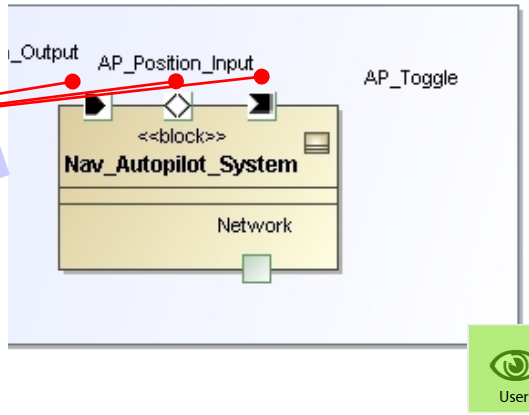
Ports

- Flow ports are represented by MARTE Flow Ports

```
system Nav_Autopilot_Systemfeatures
        AP_Toggle : in event port;
        AP_Position_Input : in event data port Nav_Types::Position.GPS; …
```

Generated Code

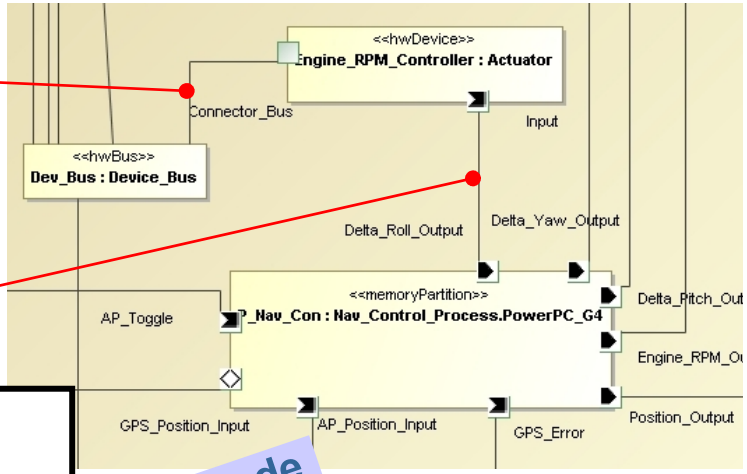
AADL



Port, Bus and Memory Connections

- Bus/data access data are represented by UML connectors between the port providing the access and the device

- Flows ports connections are represented by UML connectors (delegation or assembly connectors)




```
system implementation Nav_Autopilot_System.PowerPC_G4
….
bus access Dev_Bus -> Engine_RPM_Controller.Connector_Bus;
….
data port P_Nav_Con.Engine_RPM_Output -> Engine_RPM_Controller.Input ;
…
```

Generated Code

AADL

UML MARTE

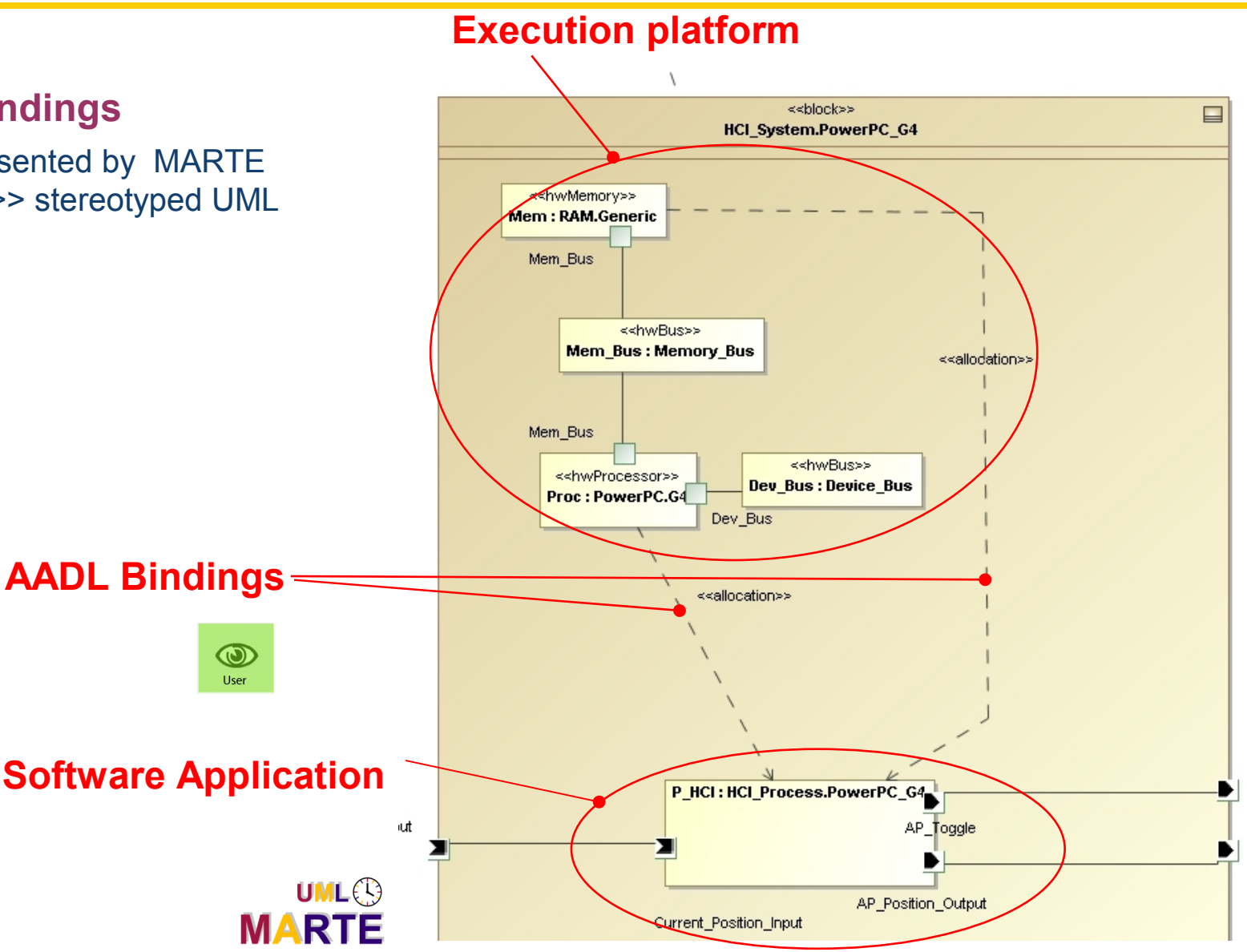
Copyright THALES – Copyright CEA – Copyright INRIA

cea list THALES INRIA

System binding

System bindings

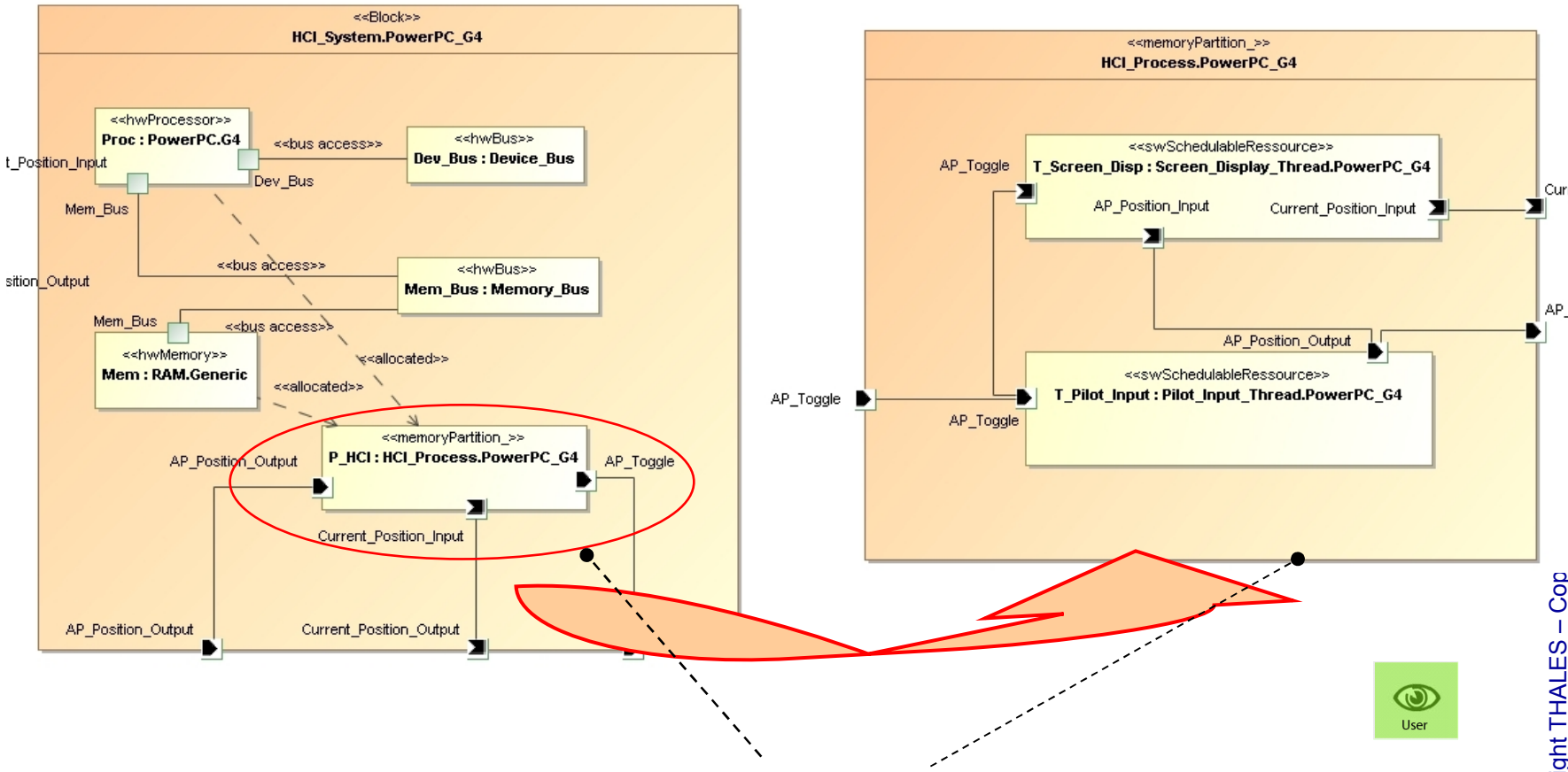
- Are represented by MARTE <<allocation>> stereotyped UML Dependency



Copyright THALES – Copyright CEA – Copyright INRIA

UML Hierarchical view

Nested hierarchical views in UML



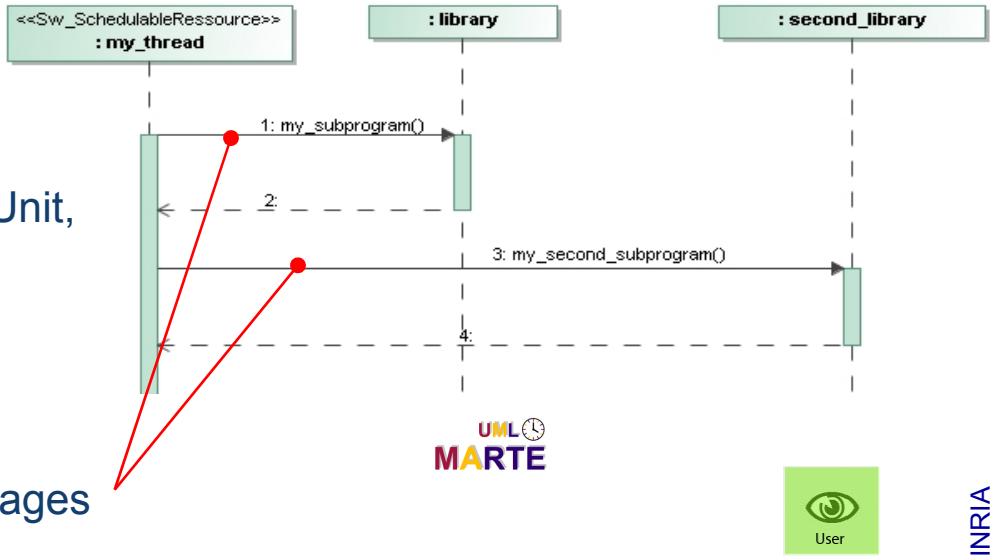
Software application

Copyright THALES – Cop

Subprogram call

Subprogram

- Are represented by UML Operations
- May be specialized with RT features (PpUnit, …)



Subprogram calls

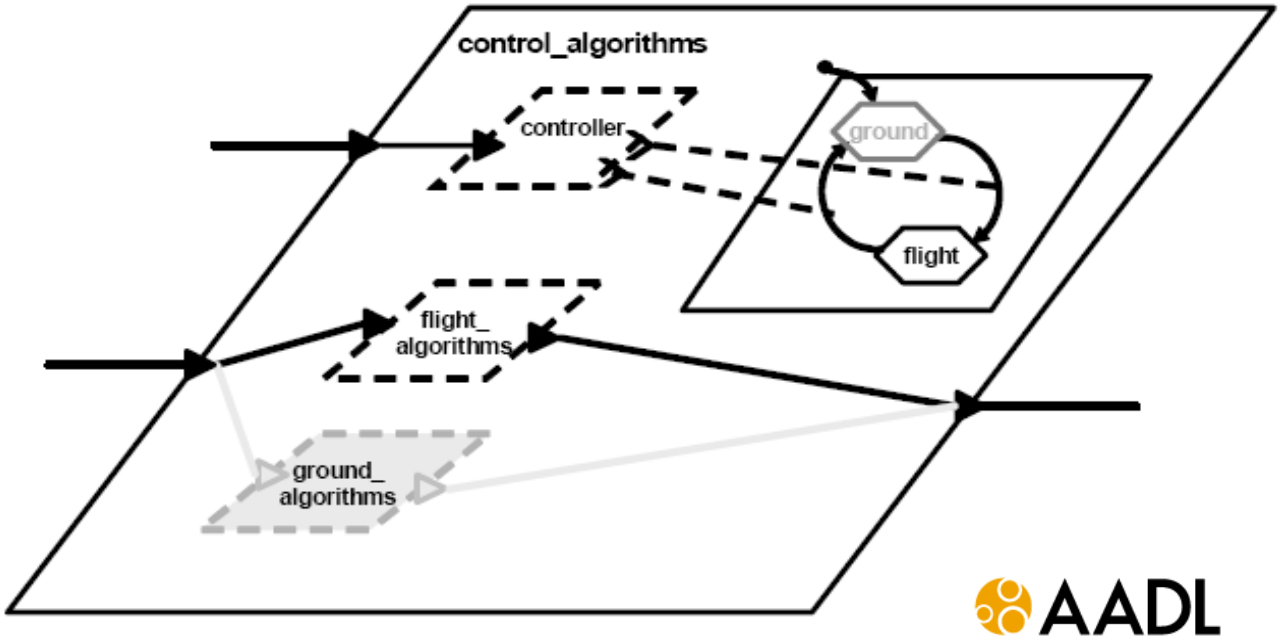
- Are represented through UML Messages on sequence diagrams

```
thread implementation my_thread.impl
calls {
        first_subpgr : subprogram my_subprogram;
        second_subpgr : subprogram my_second_subprogram;
    }
end my_thread.impl;
```

AADL

Copyright THALES – Copyright CEA – Copyright INRIA

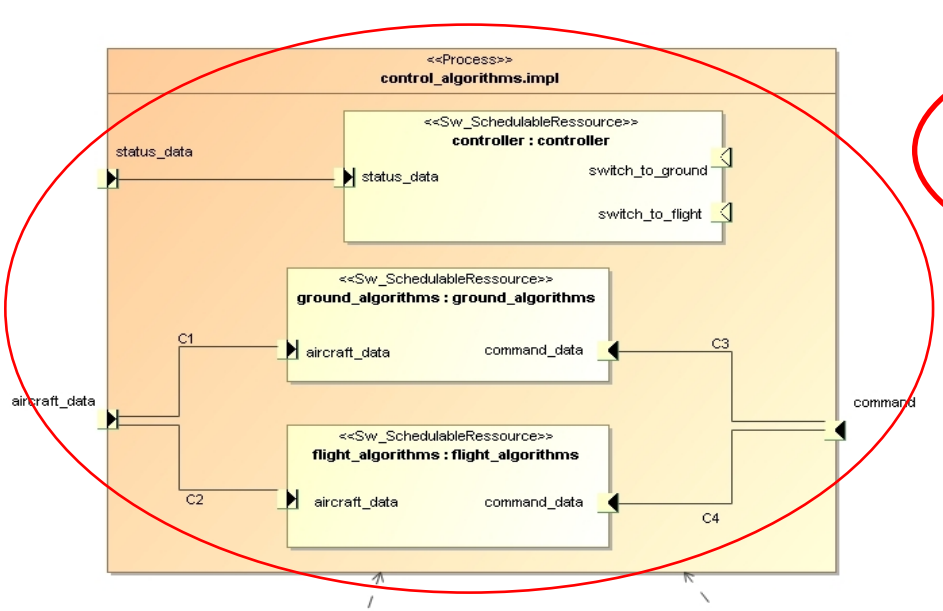
Modes in AADL

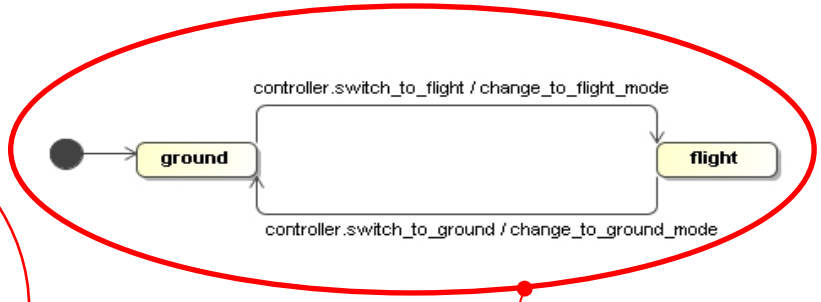


Flight simulator: a dynamic re-configuration thought mode switching

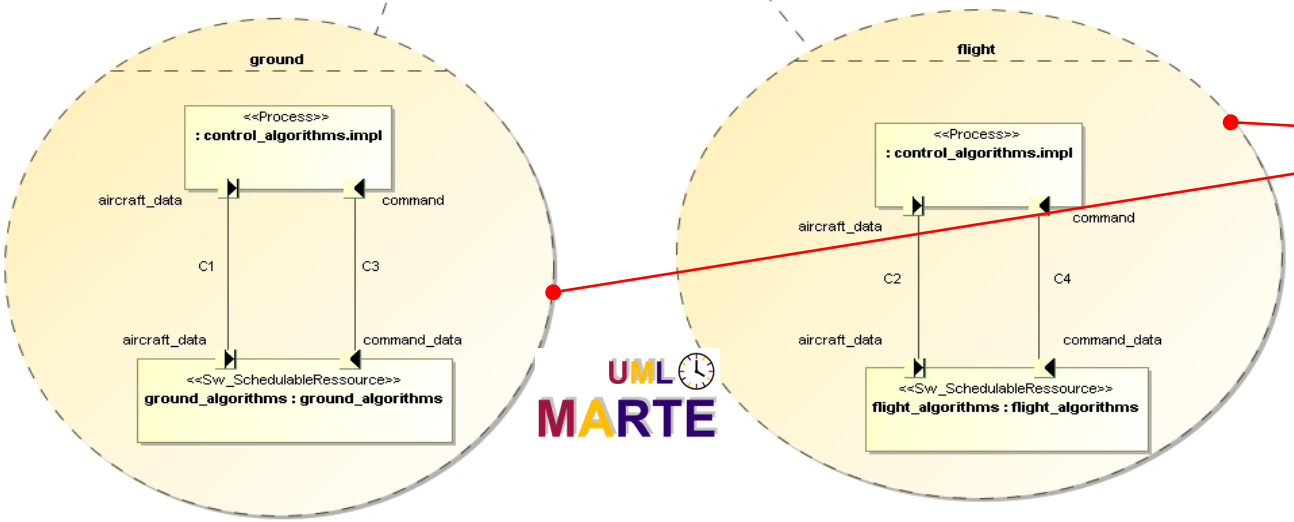
Copyright THALES – Copyright CEA – Copyright INRIA

Modes with MARTE guidelines





Mode transition modelized by an UML state machine



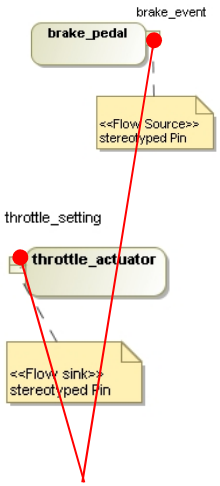
Different mode configuration through Collaboration diagrams

Copyright THALES – Copyright CEA – Copyright INRIA

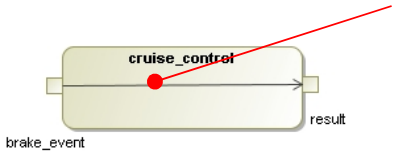
AADL Flows

Flows

- Used for specifying end-to-end flows for flow analysis (end-to-end timing and latency, reliability, error propagation, QoS,…)
- Are defined as flow sinks, flow source, flow path and end-to-end flows



Flow sink,
flow source
(Object pins on Action nodes)



Flow path
(object flow between object pins)

End to end flow
(Object node between Object Flows)

Copyright THALES – Copyright CEA – Copyright INRIA

AADL Properties

AADL properties language extensibility

- Will be addressed using NFPs, VSL language
- Alignment between both languages is going on

AADL properties

- Mapped on MARTE concepts properties

UML MARTE

Copyright THALES – Copyright CEA – Copyright INRIA