

# AADL : a radar case study



## Back to radar case study

---



- Goal: to model a simple radar system
- Let us suppose we have the following requirements
  1. System implementation **is composed by physical devices** (Hardware entity): antenna + processor + memory + bus
  2. and **software entity** : **running processes and threads** + operating system functionalities (scheduling) implemented in the processor that represent a part of execution platform and physical devices in the same time.
  3. The **main process is responsible for signals processing** : general pattern: **transmitter -> antenna -> receiver -> analyzer -> display**
  4. **Analyzer is a periodic thread** that compares transmitted and received signals to perform detection, localization and identification.
  5. [..]

# Tools used for modeling

---

- AADL syntax is both textual and graphical, with several editors available
  - Modes exist for emacs, vi
  - OSATE2 provides a comprehensive IDE on top of Eclipse, and additional plug-ins
    - IMV : Instance Model Viewer
    - Consistency checkers, statistics, etc.
  - AADLInspector, textual editor from Ellidiss
  - Adele: graphical editor for Eclipse
- In the following, we will use OSATE2 and IMV

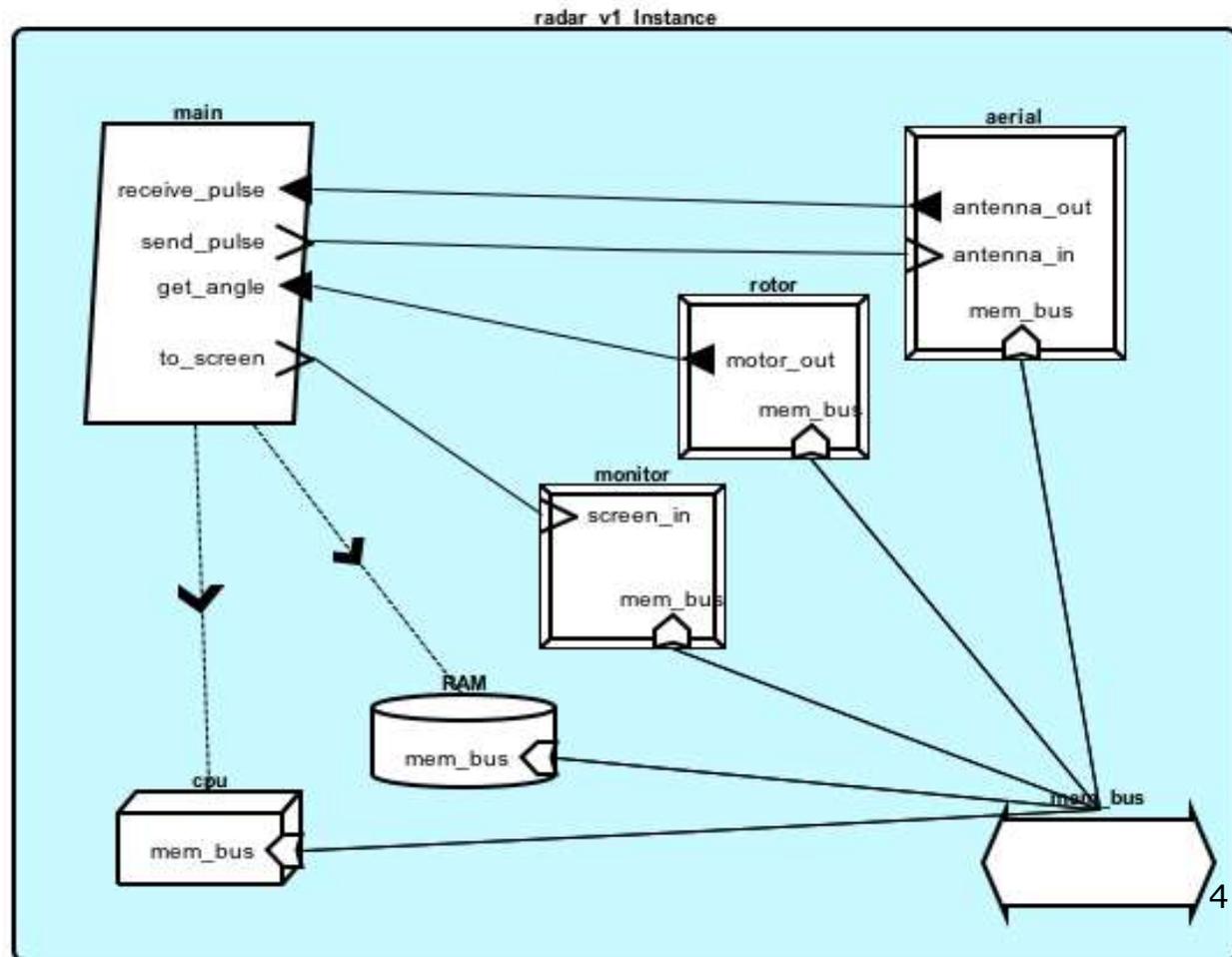
# Radar case study

## Hardware/Software breakdown: components

```
PACKAGE radar
PUBLIC

PROCESS processing
-- ...
END processing;
DEVICE antenna
-- ...
END antenna;

END RADAR;
```



# Radar case study

## Hardware/Software breakdown: features

```
PROCESS processing
```

**FEATURES**

```
to_screen : OUT EVENT PORT;  
send_pulse : OUT EVENT PORT;  
receive_pulse : IN DATA PORT;  
get_angle : IN DATA PORT;
```

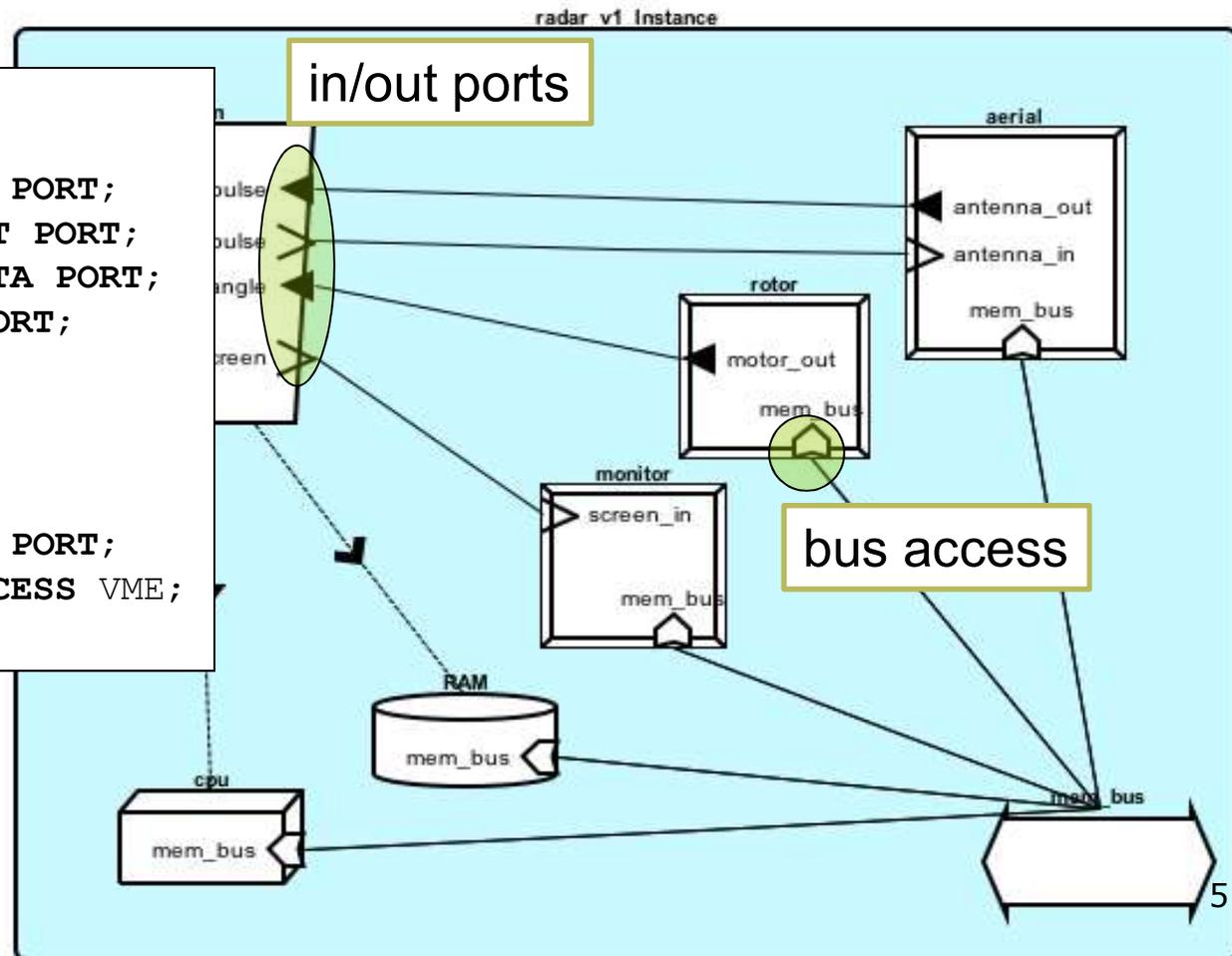
```
END processing;
```

```
DEVICE antenna
```

**FEATURES**

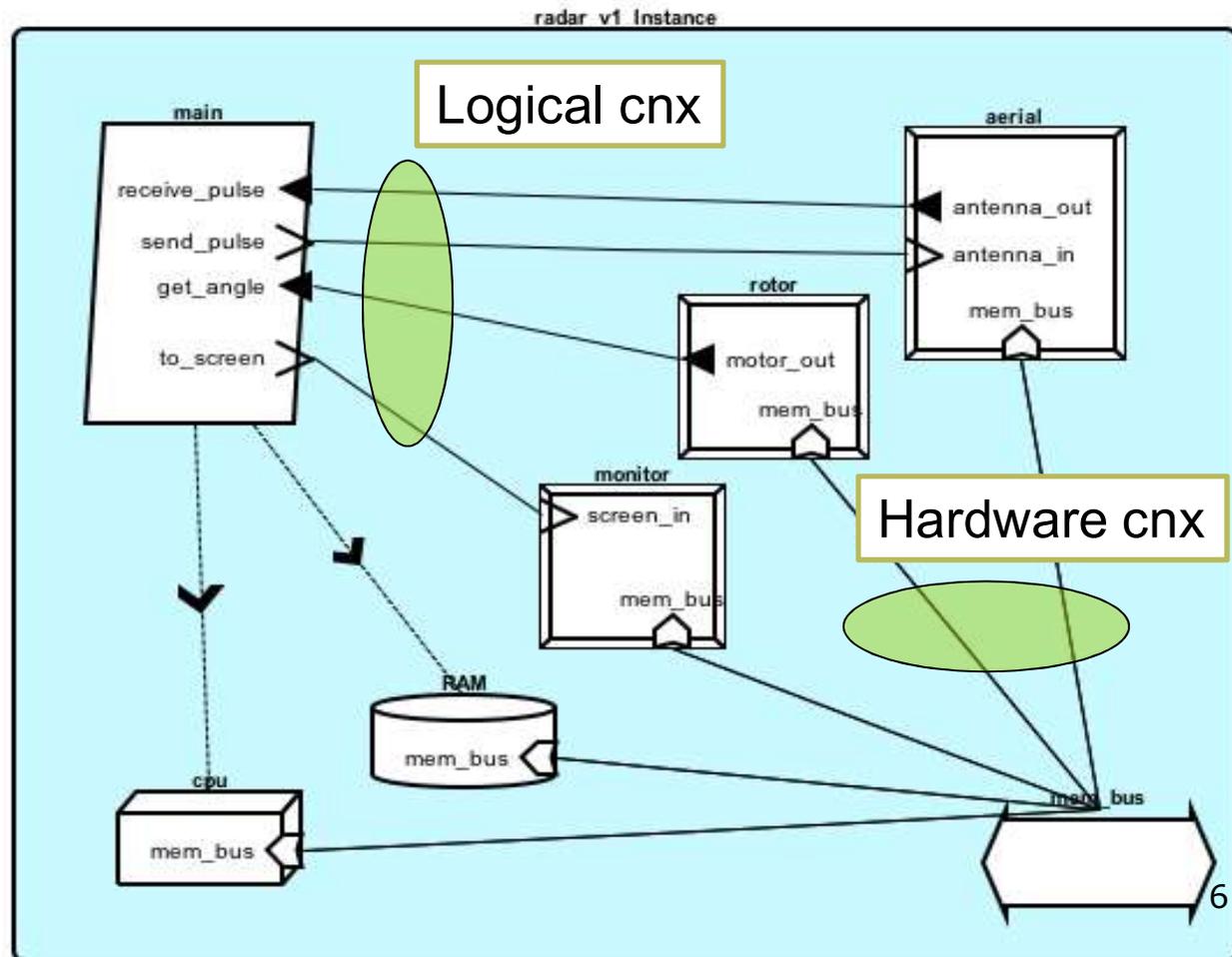
```
antenna_in : IN EVENT PORT;  
VME : REQUIRES BUS ACCESS VME;
```

```
END antenna;
```



# Radar case study

## Hardware/Software breakdown: connections



# Radar case study

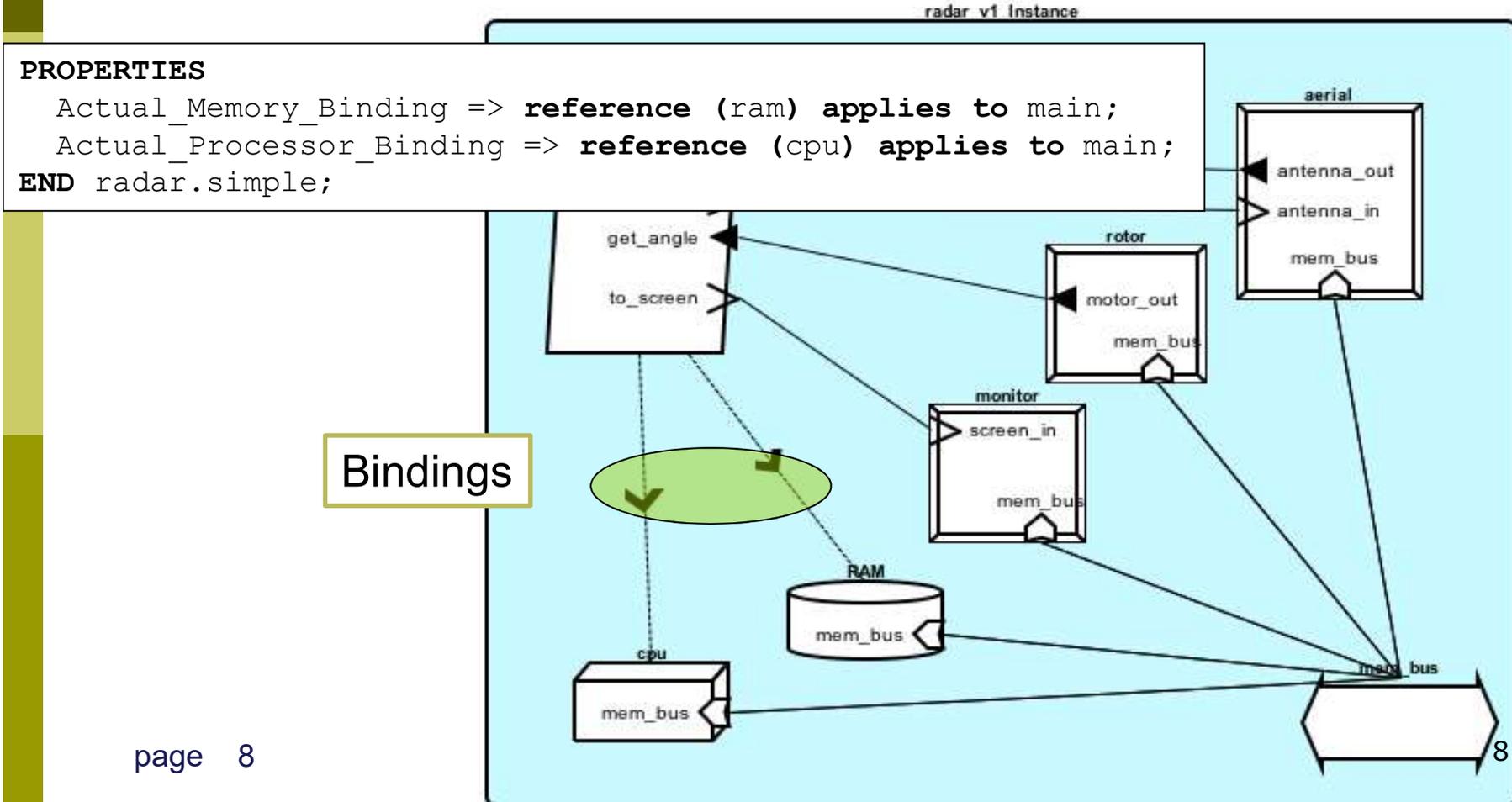
---

## □ Hardware/Software breakdown: connections

```
SYSTEM IMPLEMENTATION radar.simple  
SUBCOMPONENTS  
  aerial : DEVICE antenna;  
  rotor  : DEVICE motor;  
  monitor : DEVICE screen;  
  main   : PROCESS processing.others;  
  cpu    : PROCESSOR leon2;  
  VME    : BUS VME;  
  RAM    : MEMORY RAM;  
CONNECTIONS  
  PORT aerial.antenna_out -> main.receive_pulse;  
  PORT rotor.motor_out   -> main.get_angle;  
  PORT main.send_pulse   -> aerial.antenna_in;  
  PORT main.to_screen    -> monitor.screen_in;  
  BUS ACCESS VME -> aerial.VME;  
  BUS ACCESS VME -> rotor.VME;  
  BUS ACCESS VME -> monitor.VME;  
  BUS ACCESS VME -> cpu.VME;  
  BUS ACCESS VME -> RAM.VME;
```

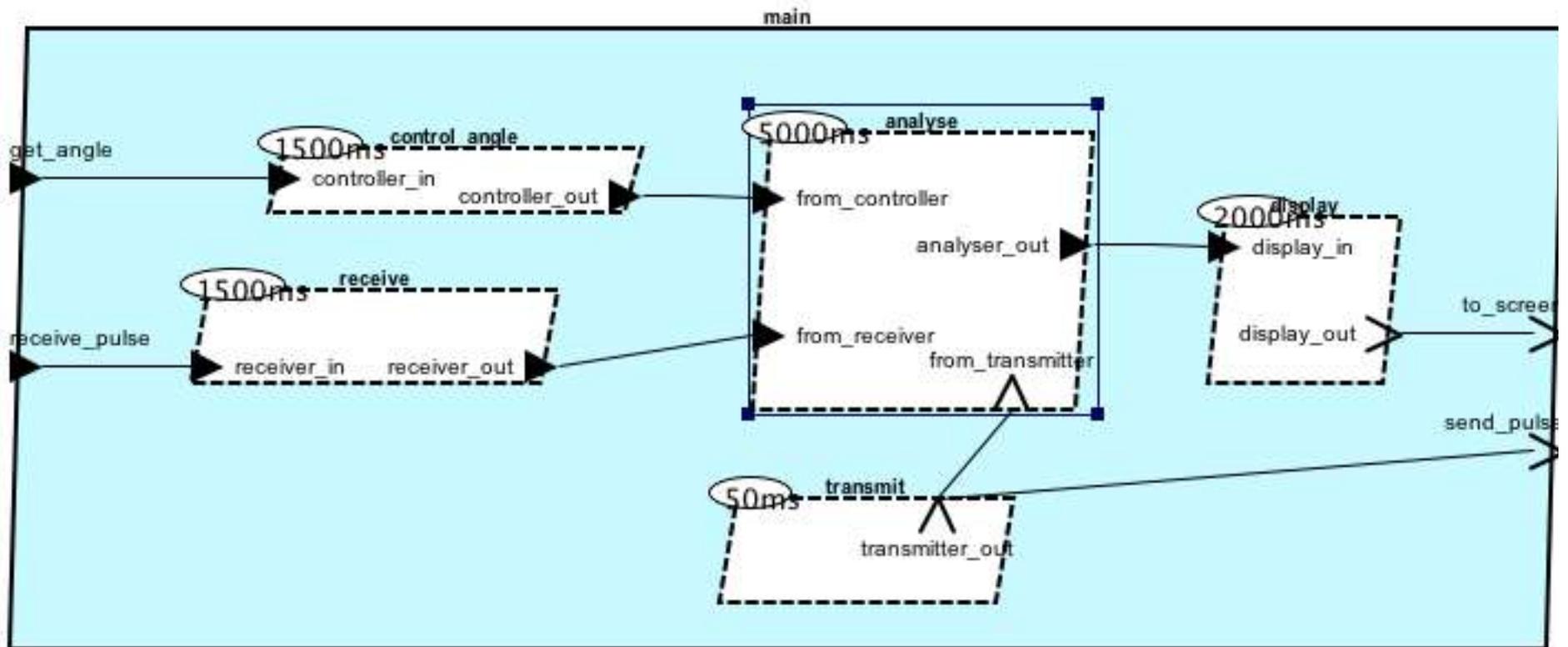
# Radar case study

## Hardware/Software breakdown: bindings



# Radar case study

## Software elements



## A few words on AADL usage

---

- AADL is for architectural description
- Not to be compared with UML suites
  - Behavior, types, link with source code is not required
- Keep in mind models support an objective
  - For now, it is just a high-level view of the design
- In the next sections, we will complete the models with
  - Properties to support schedulability analysis
  - Elements to generate actual implementation