Supporting the Partitioning process in Hardware/Software Co-design with VDM-RT

José Antonio Esparza Isasa, Peter Gorm Larsen and Kim Bjerge
jaei@iha.dk, pgl@iha.dk, kbe@iha.dk

Agenda

1. Introduction: models and Hw/Sw co-design
2. VDM-RT: abstractions and partitioning assessment
3. Modelling hardware partitions
4. Servo case study
5. AVB case study
6. Further work
7. Conclusions

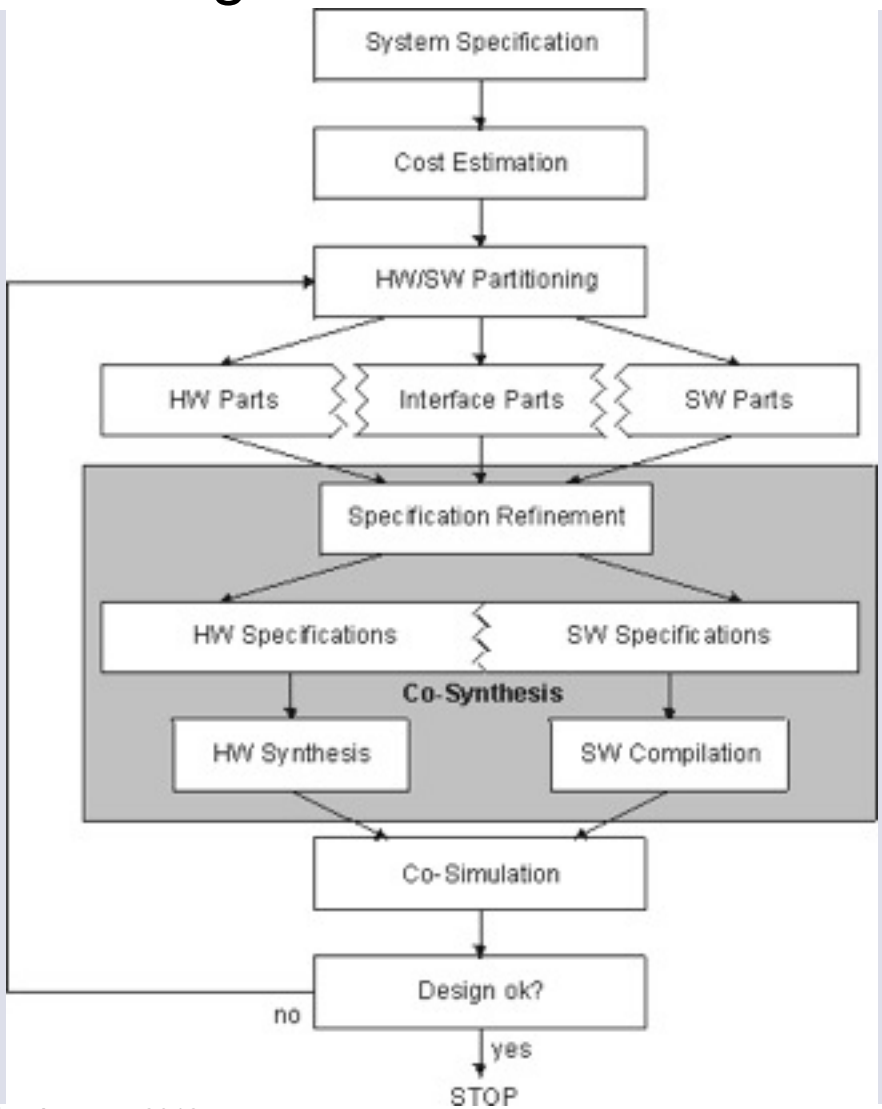
Introduction: models and Hw/Sw co-design

- Models help on approaching complex problems by applying abstraction.

- Abstraction will allow representation and analysis at different system levels.

Analysis performed over models will enable well-founded partitioning decisions!

- Models can be applied during the whole development process.

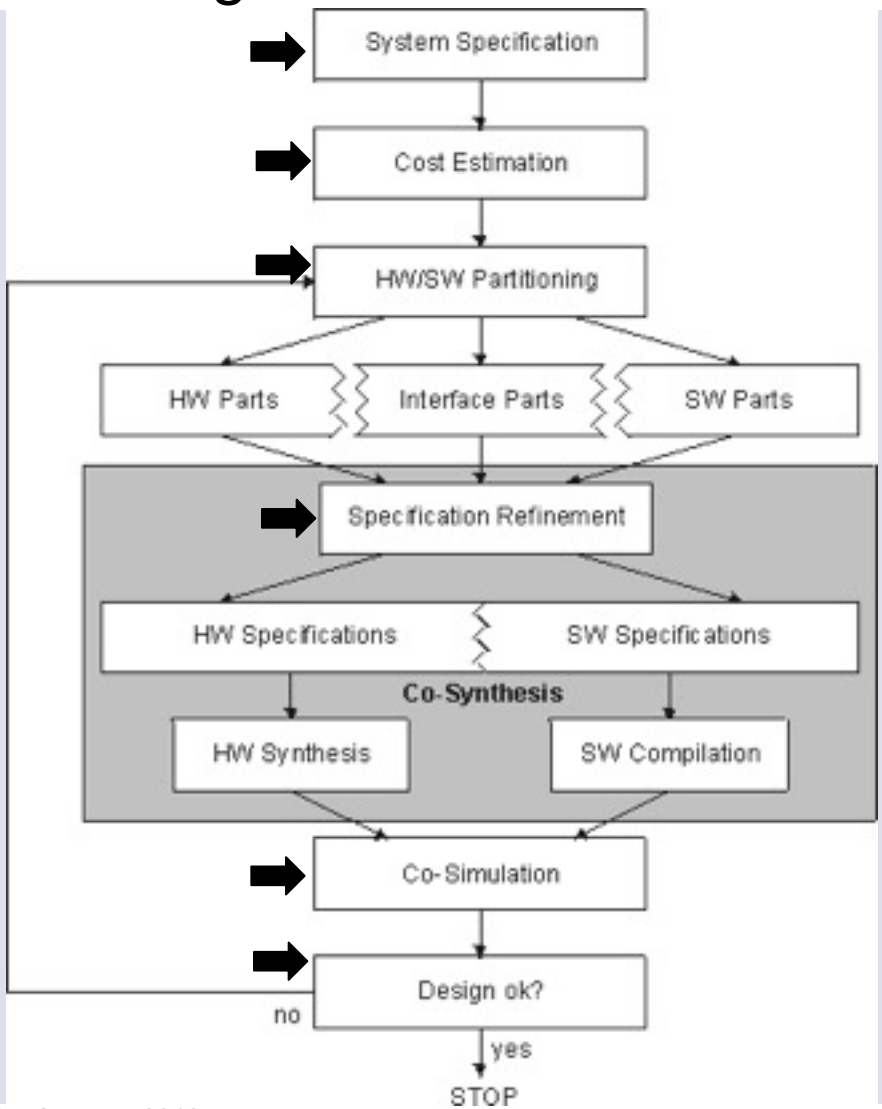


Introduction: models and Hw/Sw co-design

- Models help on approaching complex problems by applying abstraction.
- Abstraction will allow representation and analysis at different system levels.

Analysis performed over models will enable well-founded partitioning decisions!

- Models can be applied during the whole development process.



Introduction: models and Hw/Sw co-design

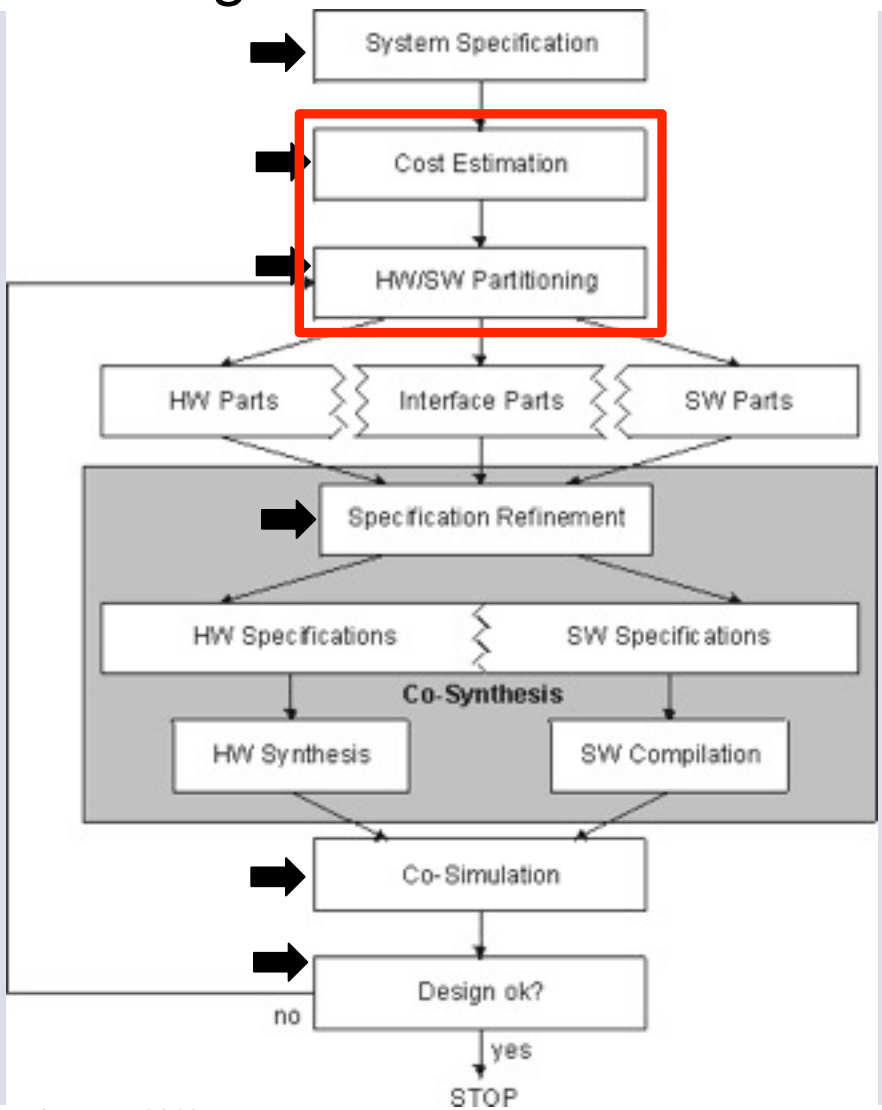
- Models help on approaching complex problems by applying abstraction.

- Abstraction will allow representation and analysis at different system levels.

Analysis performed over models will enable well-founded partitioning decisions!

- Models can be applied during the whole development process.

- **Trade-off** analysis to analyze Real Time constraints satisfaction.



VDM-RT abstractions and partitioning assessment

- A VDM-RT CPU is a computational unit with a simulated OS-layer on top
  - Different scheduling policies
  - Configurable operating frequency

```
myCPU: CPU := new CPU(<FP>, 1E5);
```

- A VDM-RT BUS communicates VDM-RT CPUs
  - Different bus access policies
  - Configurable bus capacity
  - CPU-to-CPU links

```
bus1: BUS := new BUS(<CSMACD>, 72E5,{ myCPU,myCPU2});
```

- No previous work exist about how to evaluate hardware partitions with these abstractions.

Modelling hardware partitions

Guideline 1: A VDM-RT CPU that represents a hardware partition should be configured with a processing speed orders of magnitude higher than a general-purpose CPU.

```
gpCPU: CPU := new CPU(<FP>, 1E5);
hardwarePartition: CPU := new CPU(<FP>, 1E9);
```

Guideline 2: A VDM-RT bus that communicates a hardware partition with a a general-purpose CPU should use a bandwidth orders of magnitude higher than the buses used to communicated g.p. CPUs.

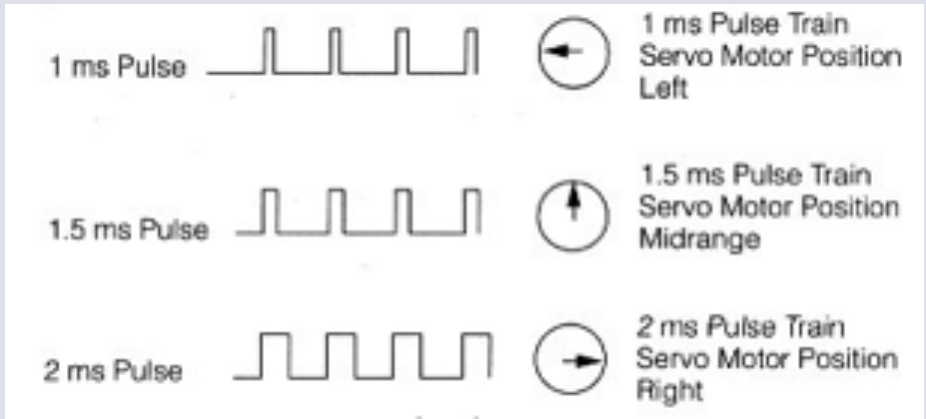
```
register: BUS := new BUS(<CSMACD>, 72E9,
                         { gpCPU,hardwarePartition});
socket: BUS := new BUS(<CSMACD>, 72E5,{ gpCPU,gpCPU});
```

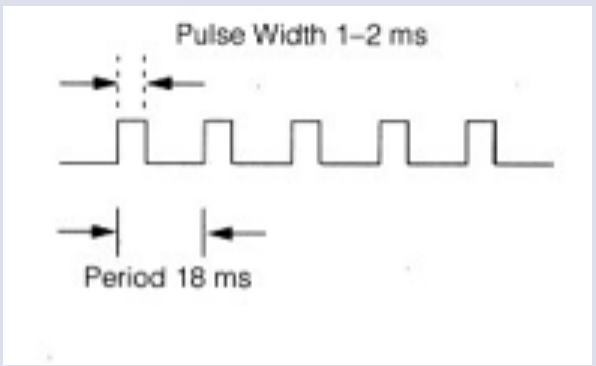
Guideline 3: Each hardware candidate should be deployed on an individual CPU configured as a hardware block.

Servo case study

- Rotates between 0º and 180º
- Controlled by PWM signal

Two real-time constraints:
- Pulse duration 1 – 2 ms
- Pulse needed every 18 ms

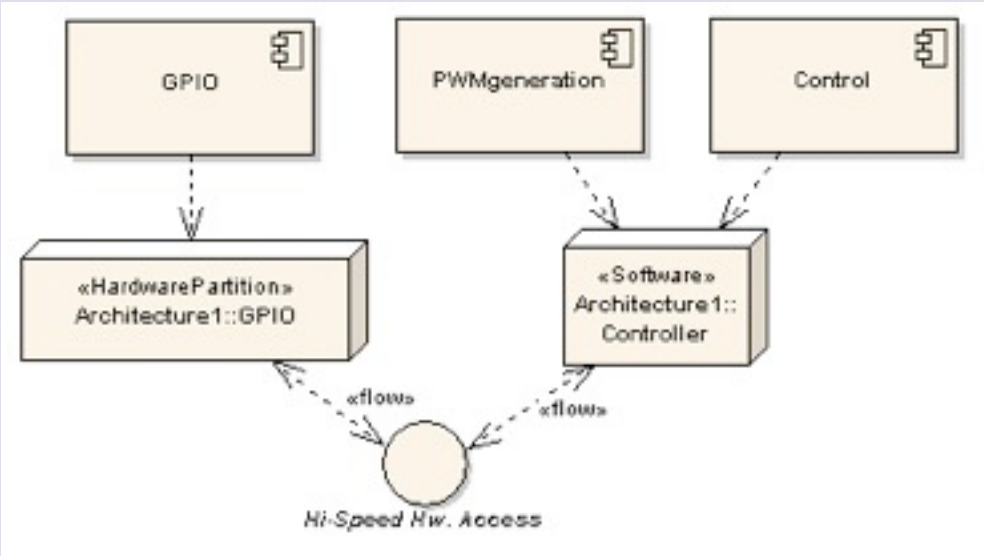




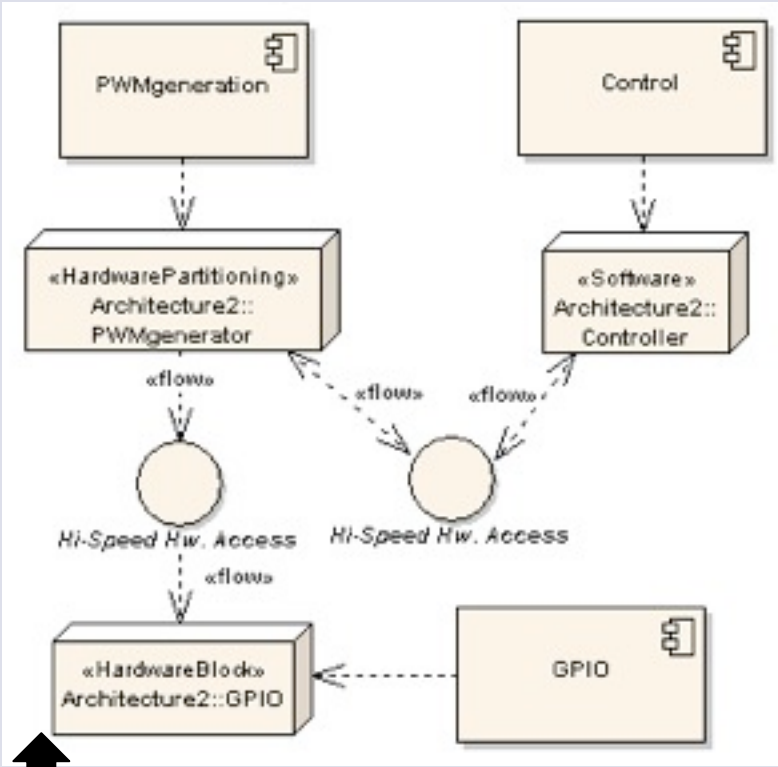

```
private generateSignal : () ==> ()
generateSignal() ==
(
 duration (2E6) outputInt.toggleBit(1);
 duration (18E6) outputInt.toggleBit(1);
 duration (0) runCount := runCount +1;
);
```

Servo case study: different deployments

Software only solution

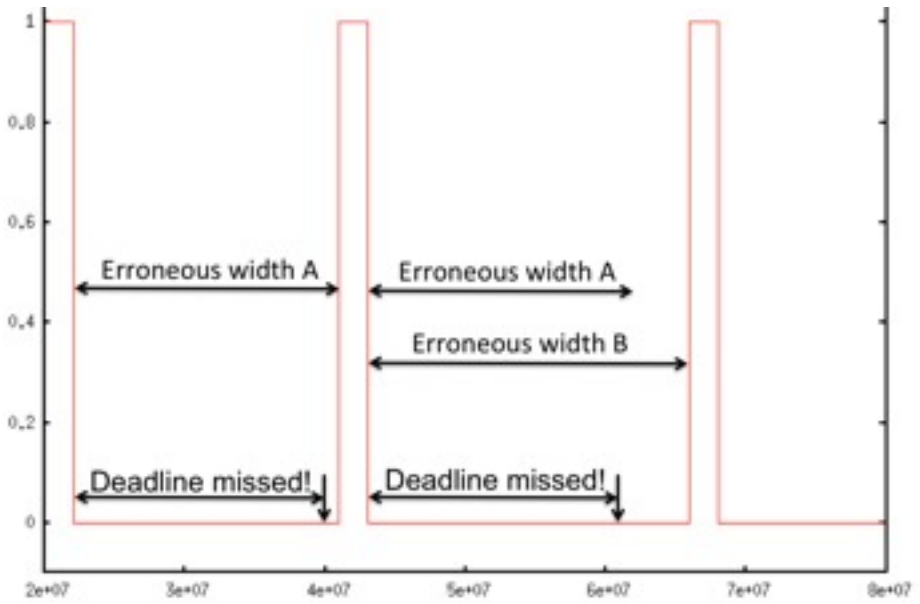


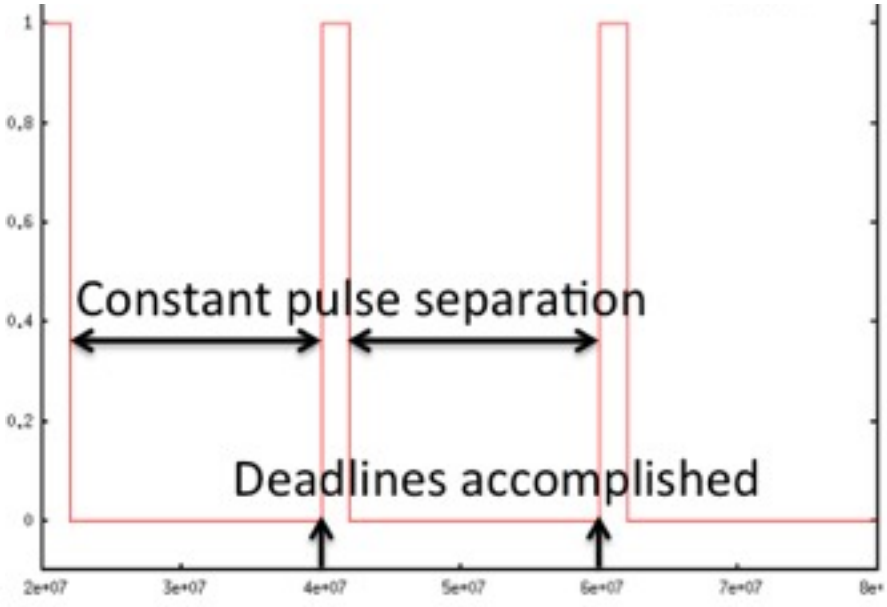
Hardware/Software solution

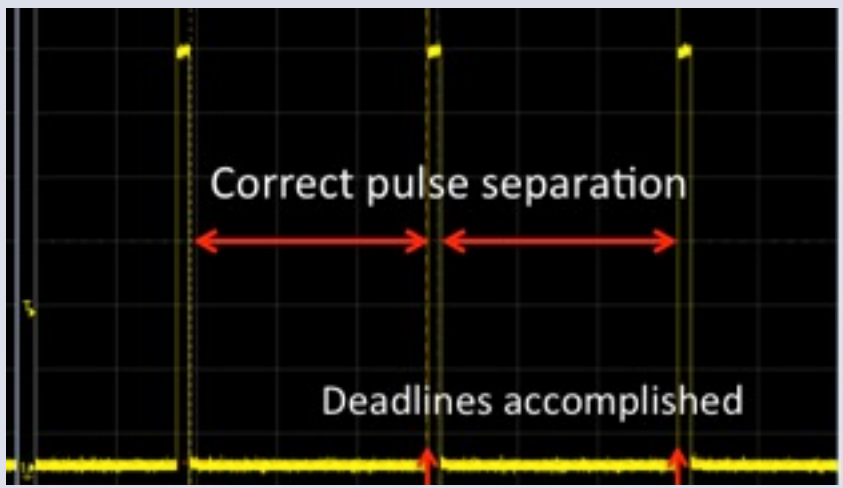



```
PWMgenerator : CPU := new CPU(<FCFS>, 1E9);
controller : CPU := new CPU(<FCFS>, 1E5);
controlRegister : BUS := new BUS(<CSMACD>,
        72E13, {PWMgenerator,controller});
```

Servo case study: modelling and implementation results



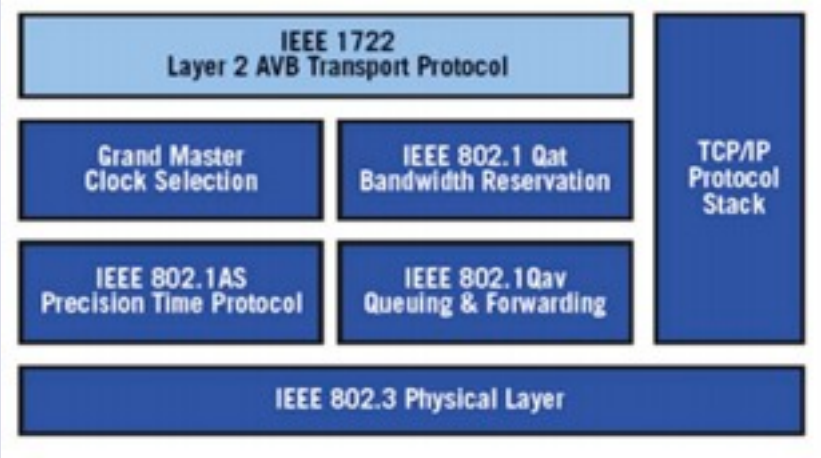




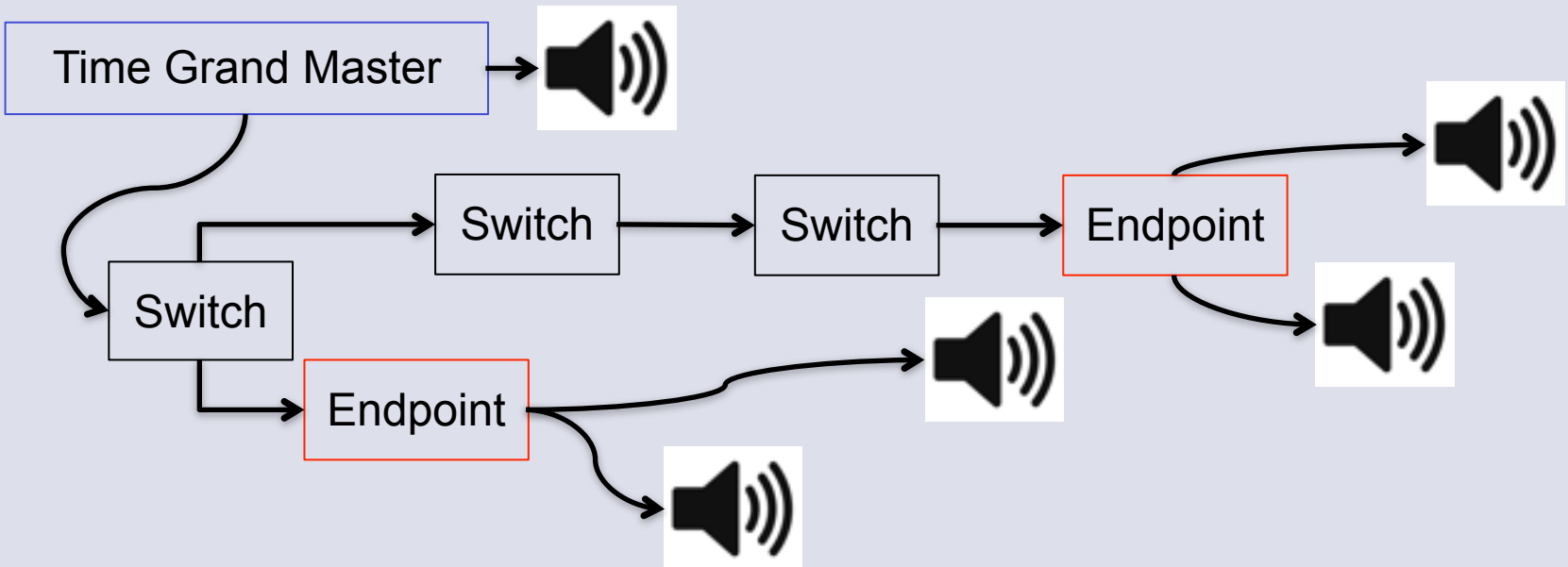
AVB case study

- Audio Video Bridging: set of protocols for multimedia transmission in digital networks
- Case study focused on the synchronization protocol (802.1AS)

- AVB network devices
  - Time Grand Master
  - Switches



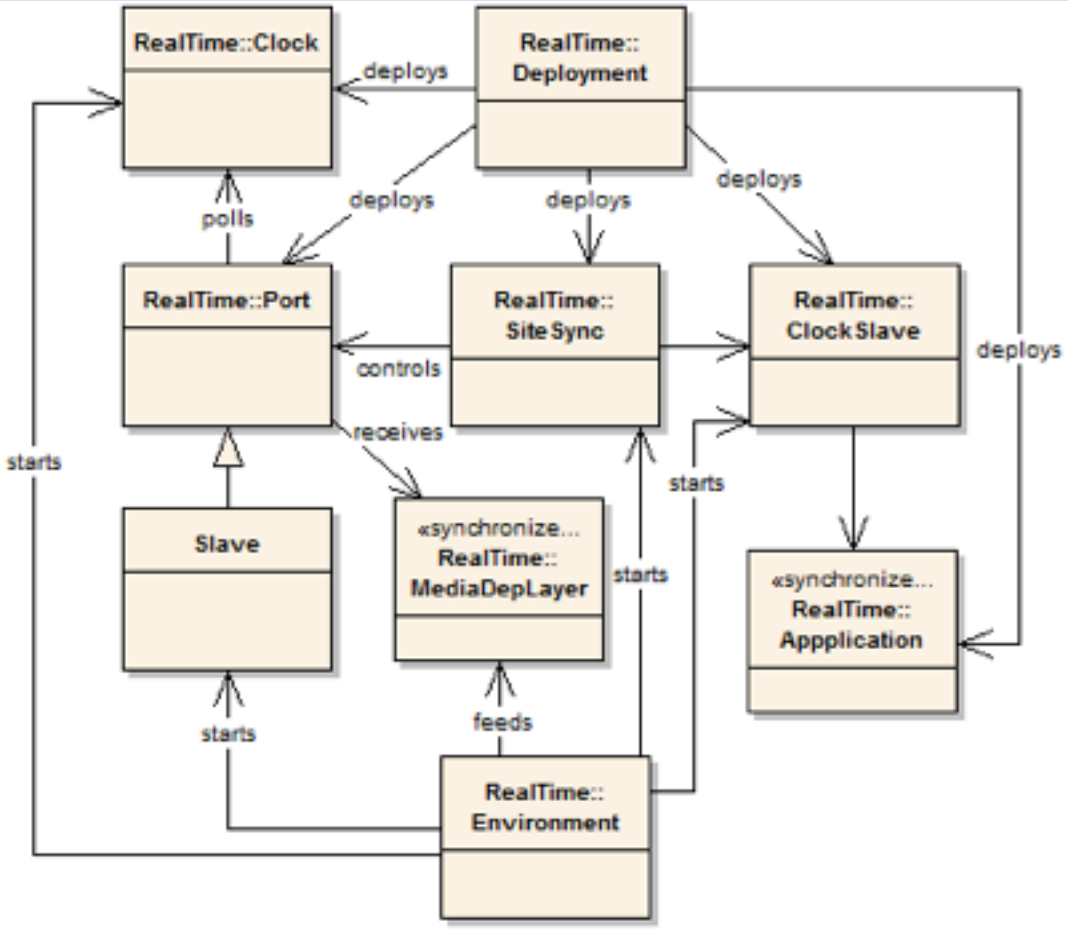
AVB case study: time synchronization in AVB networks



- Different notion of time in different speakers
  - Propagation delays due to cable length
  - Delays due to retransmission of information

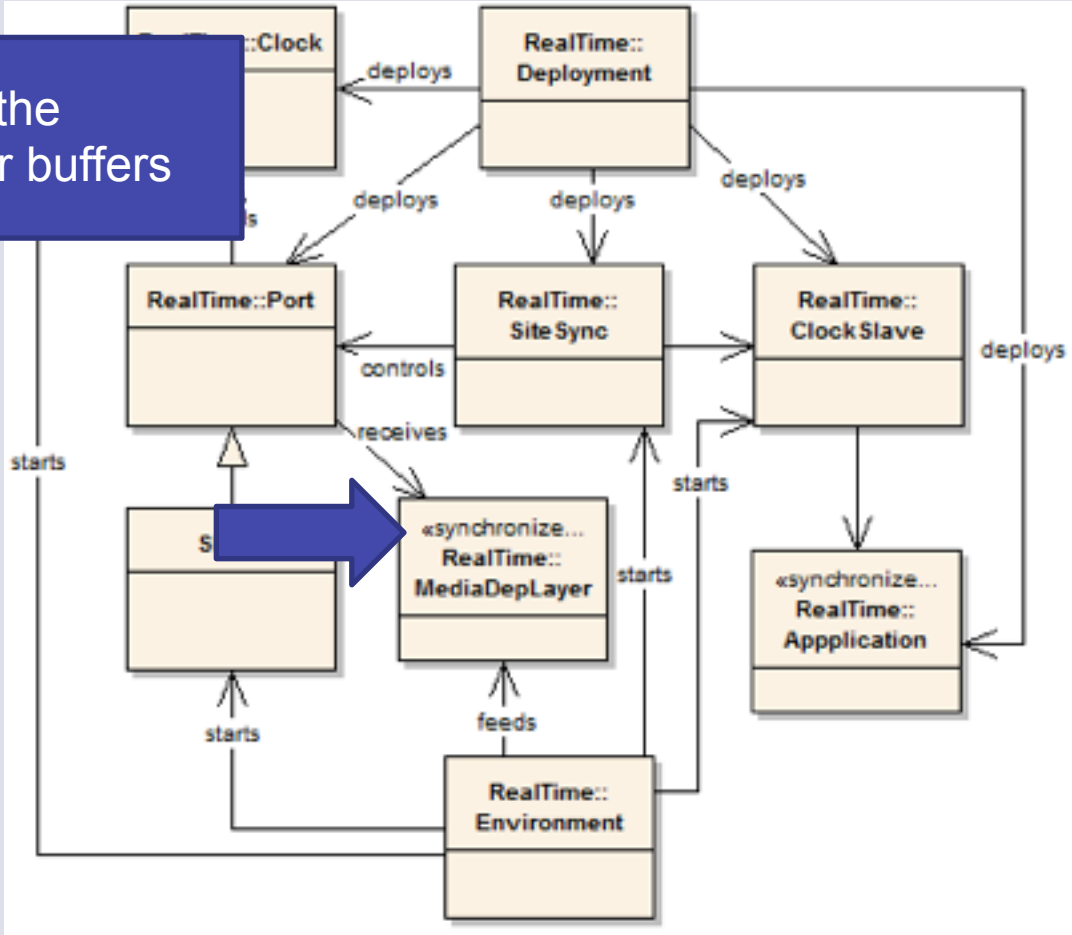
  → Additional delays due to timestamping precision

AVB case study: VDM-RT models

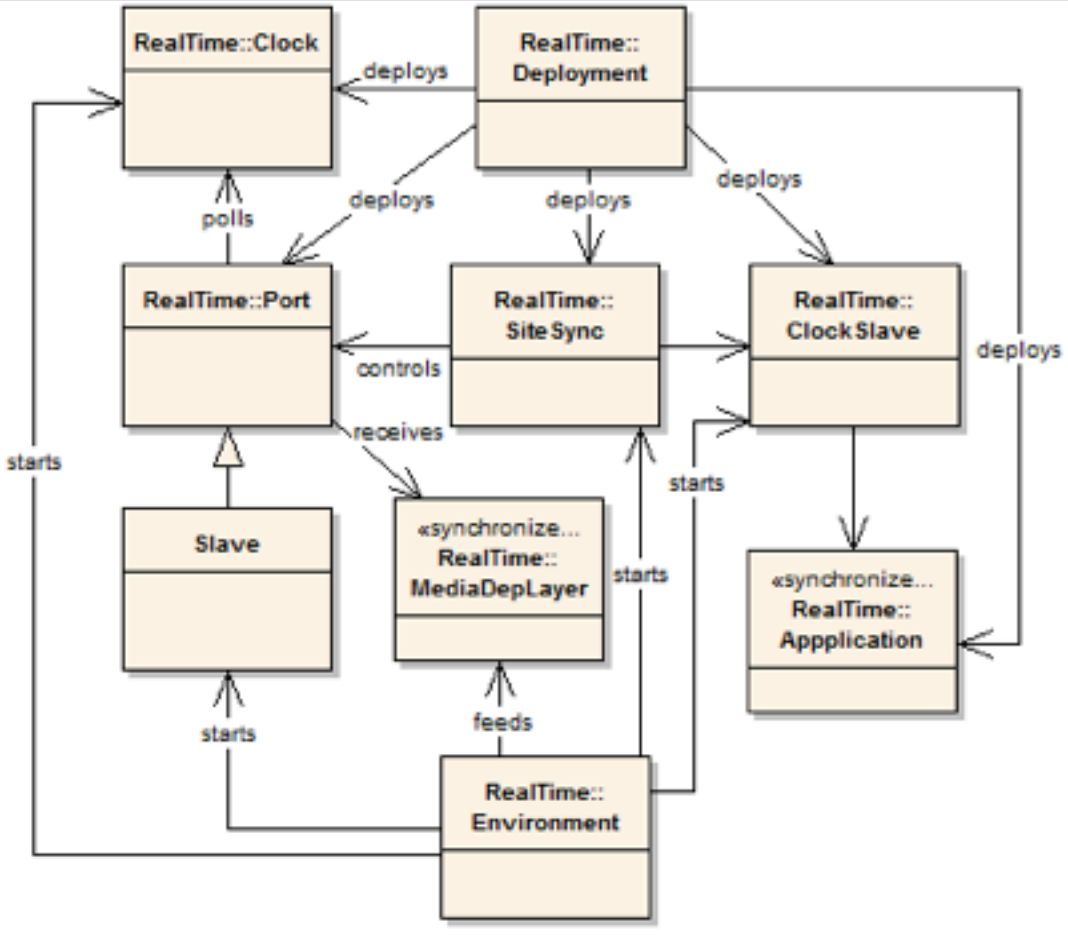


AVB case study: VDM-RT models

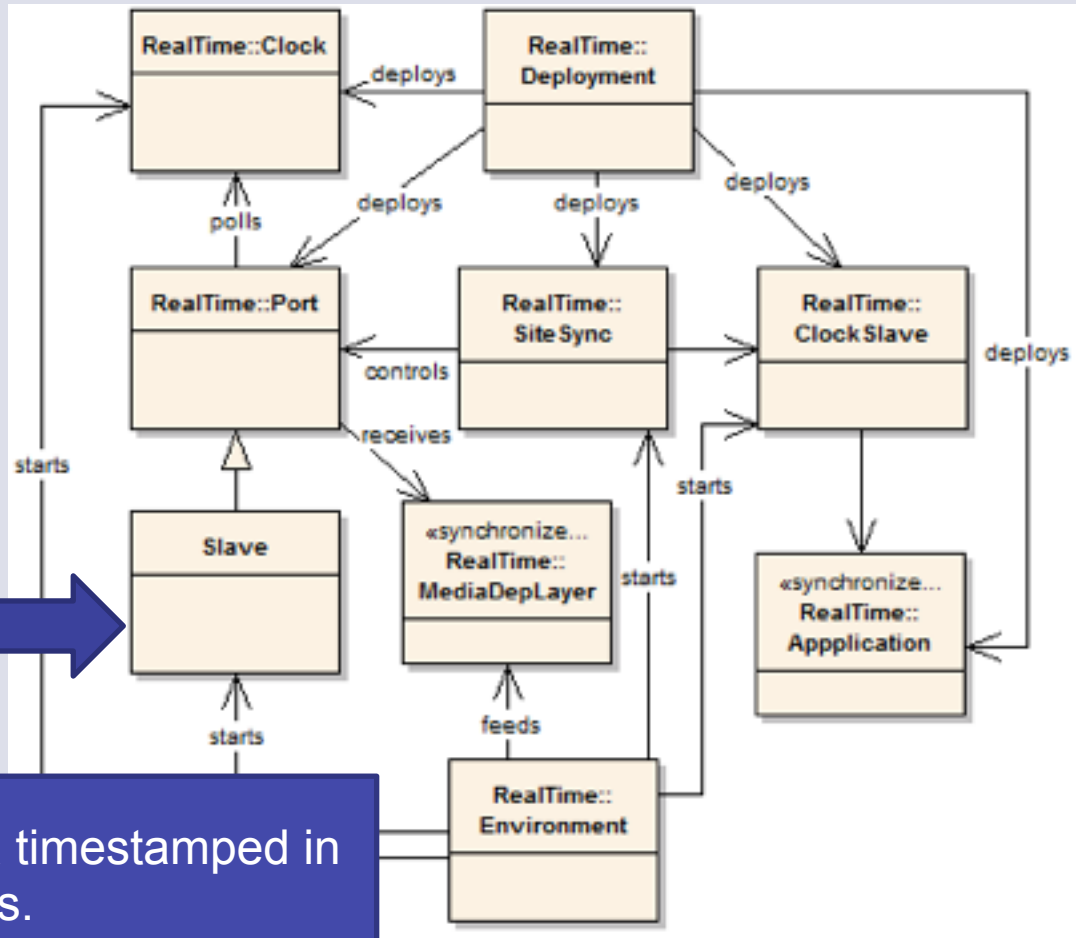
Data placed in the MediaDepLayer buffers



AVB case study: VDM-RT models

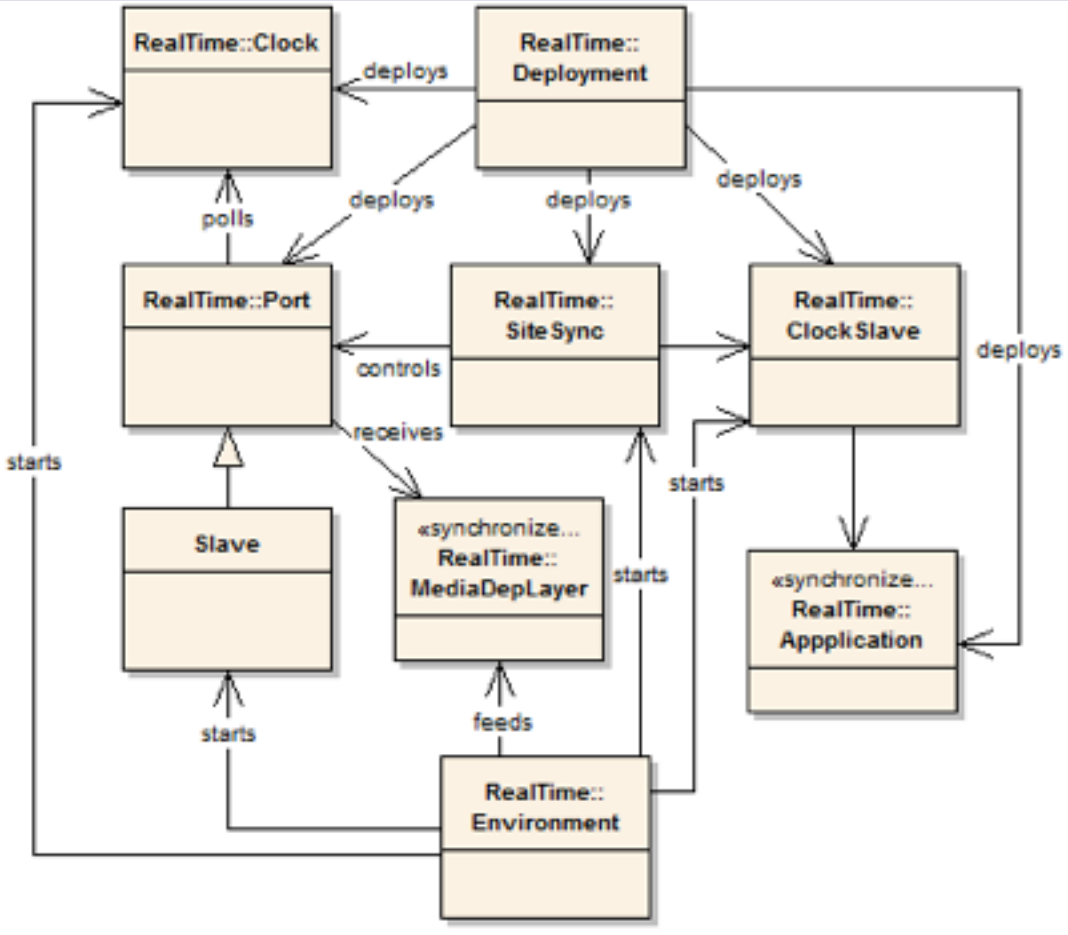


AVB case study: VDM-RT models

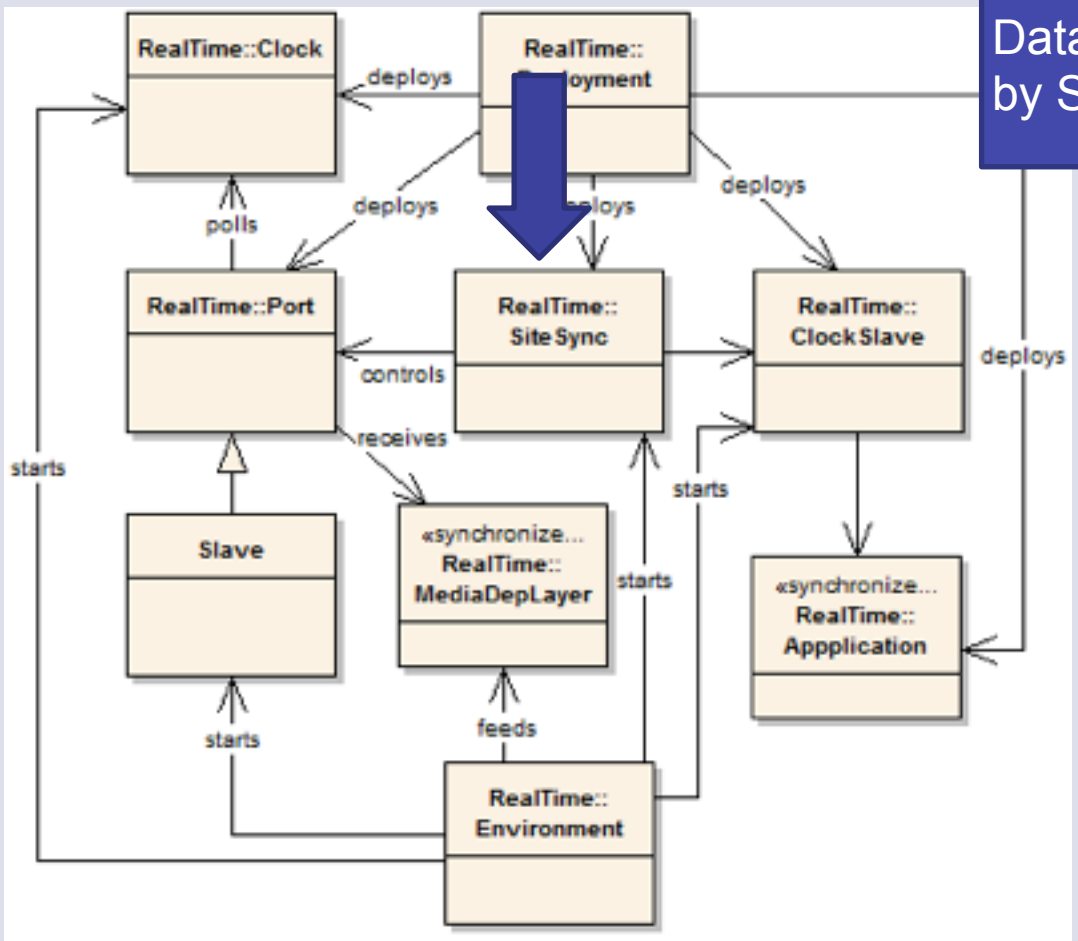


Incoming data timestamped in the Slave ports.

AVB case study: VDM-RT models

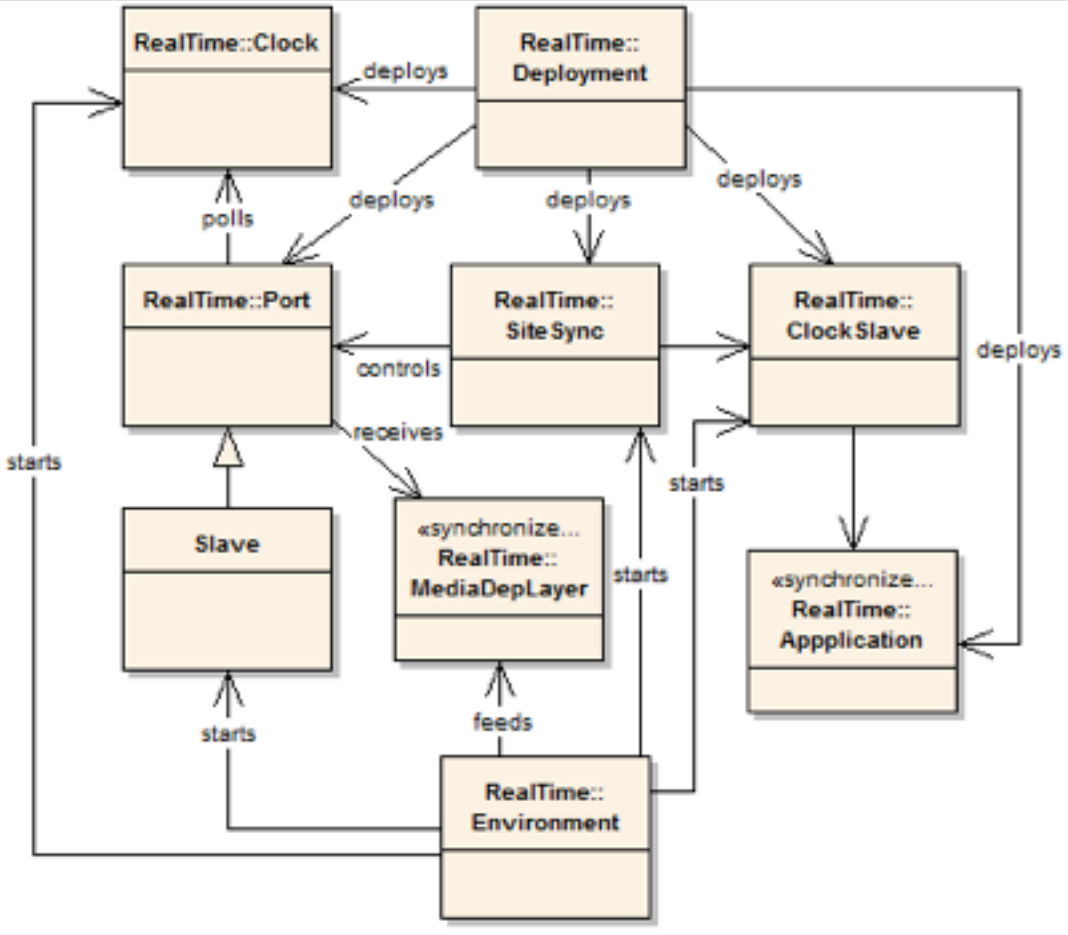


AVB case study: VDM-RT models

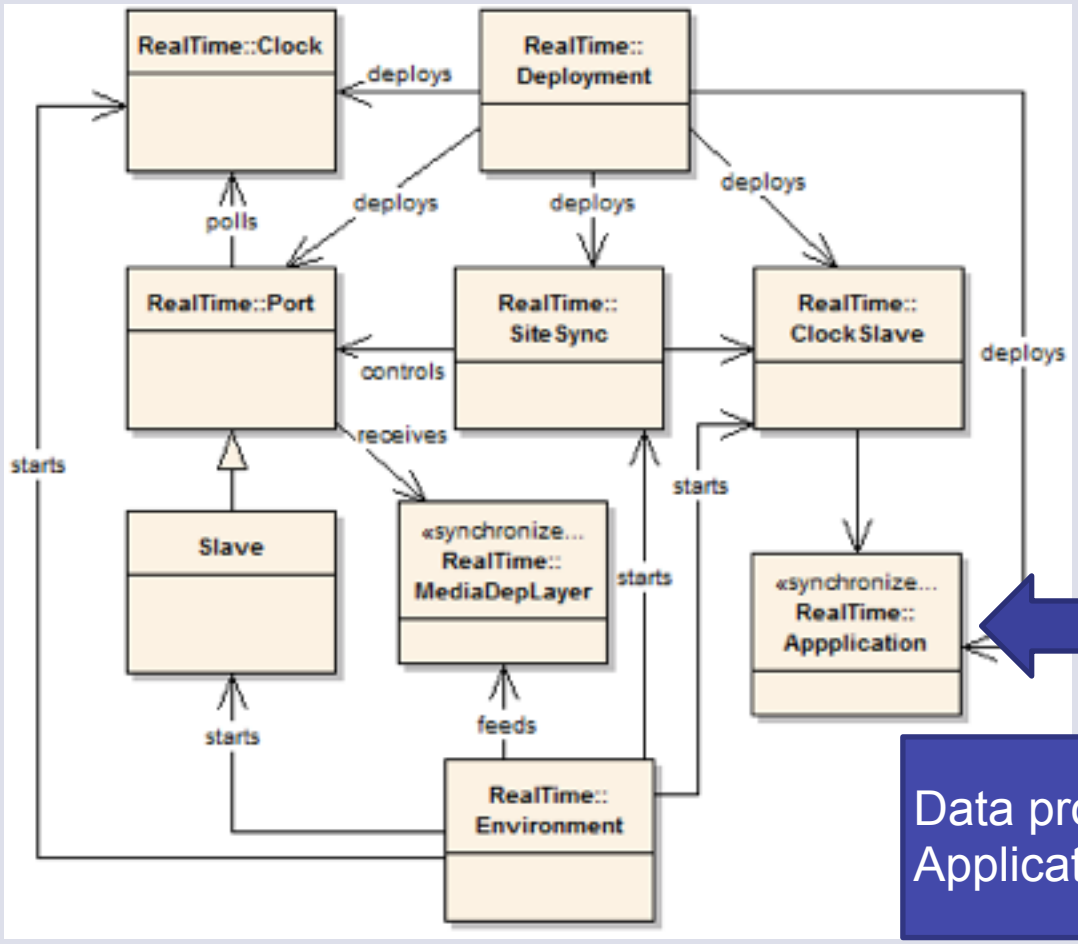


Data distributed by SiteSync

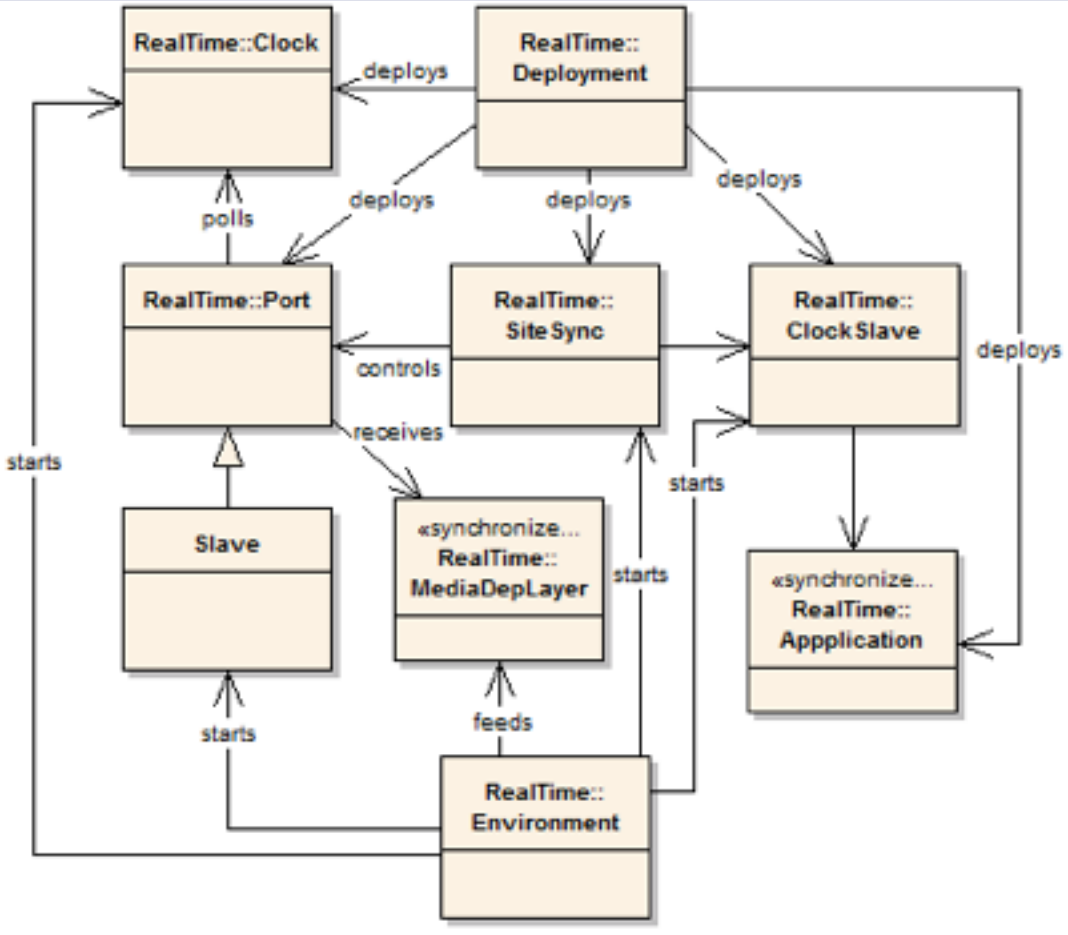
AVB case study: VDM-RT models



AVB case study: VDM-RT models



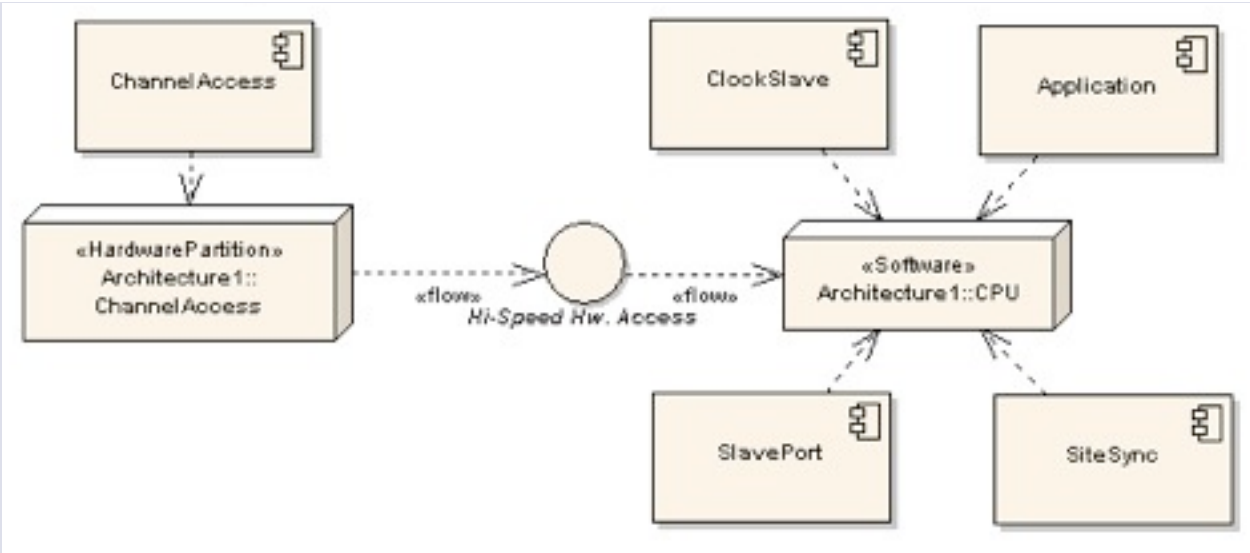
AVB case study: VDM-RT models



Evaluated architectures (I)

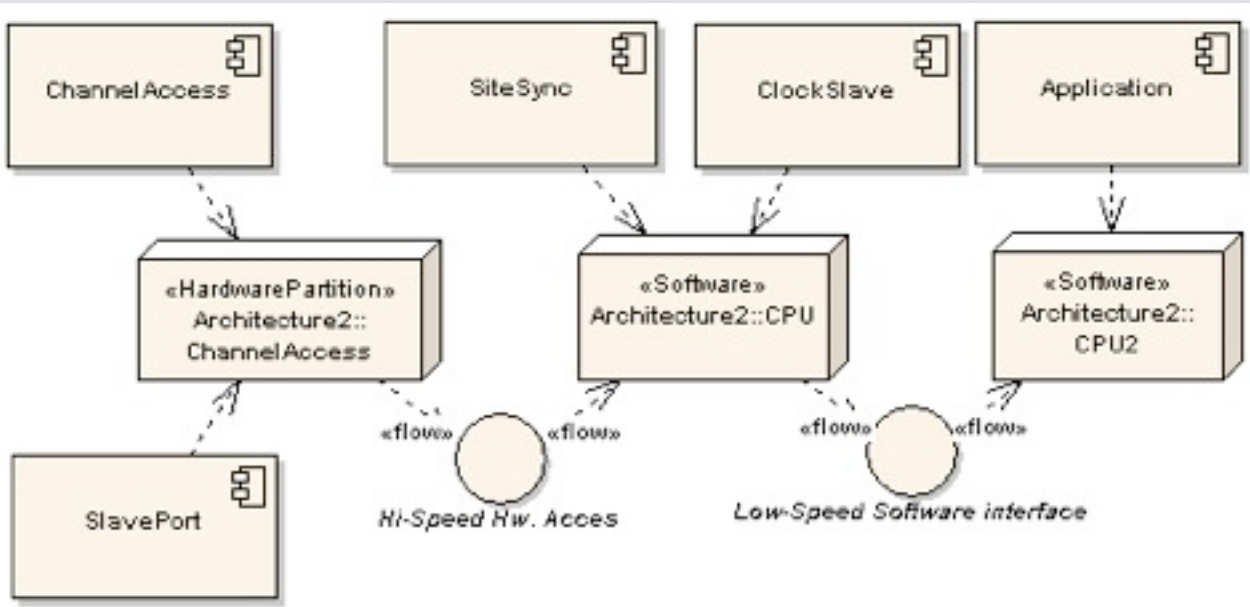
Software based solution

- 4590 ns delay using 1MHz processor.
- Delay depends on processor speed.
- We need a 1GHz processor to meet requirements!



Hardware/Software solution

- 6 ns delay using 1 MHz processor.
- Delay independent from processor speed.

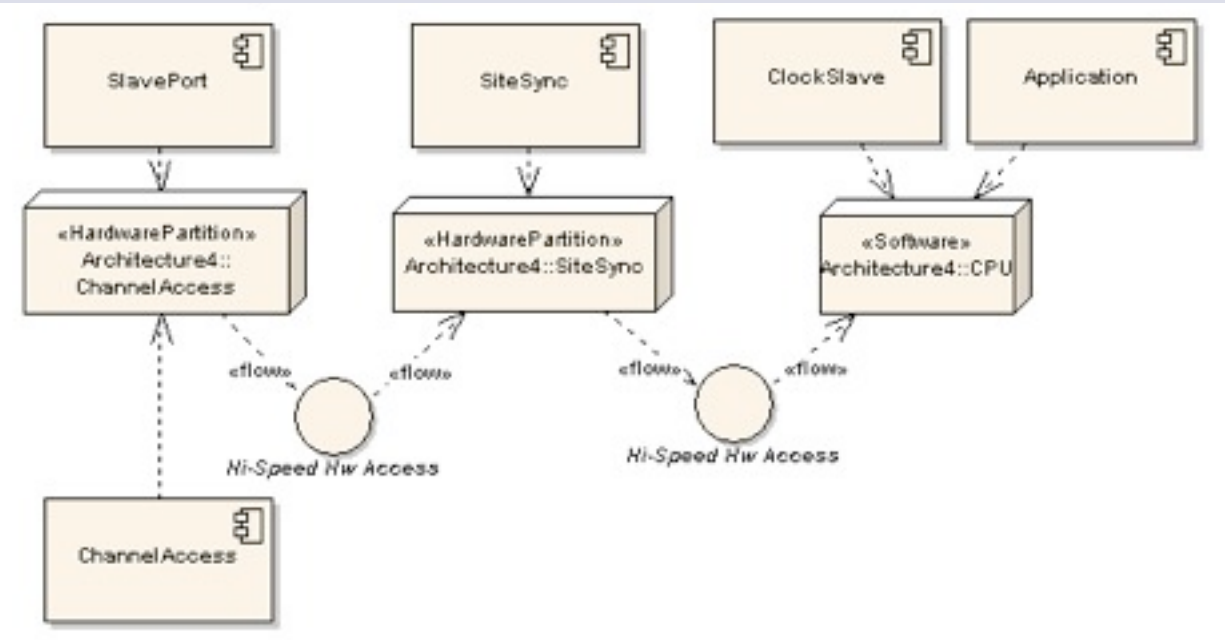


Evaluated architectures (II)

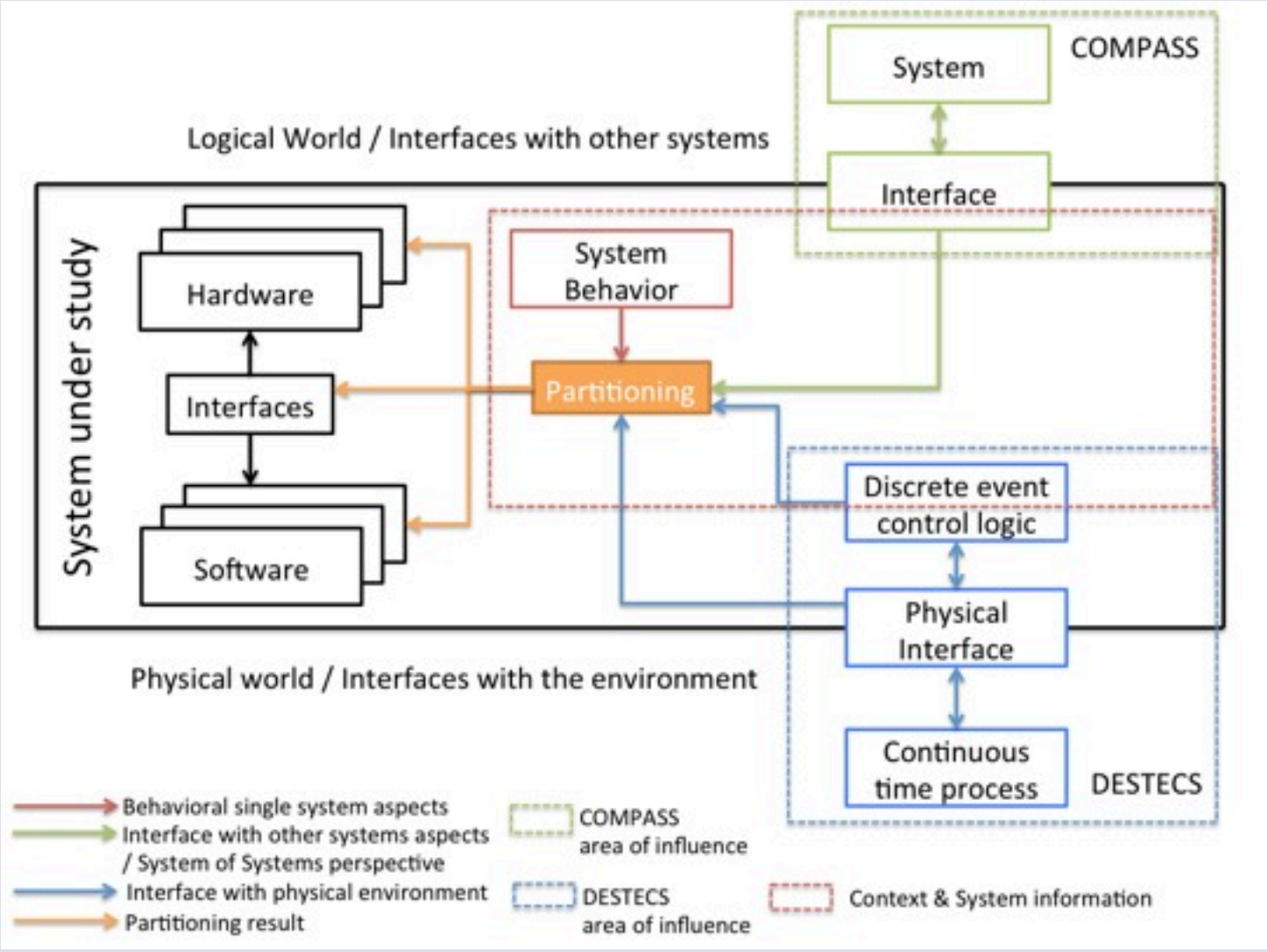
Multiple hardware partitions

- 6 ns speed independent delay kept.
- More scalable design

- More expensive solution
- Additional development time
- Additional silicon area
- More complex implementation



Further work: heterogeneous modelling



Conclusions

- VDM-RT can be used to support Hw/Sw co-design partitioning decisions based on time analysis.
- Additional effort is needed to determine the limits of this methodology.
- This methodology could be complemented by applying other modelling technologies.