Of AADL and MARTE

AOSTE (I3S/INRIA)

Frédéric Mallet

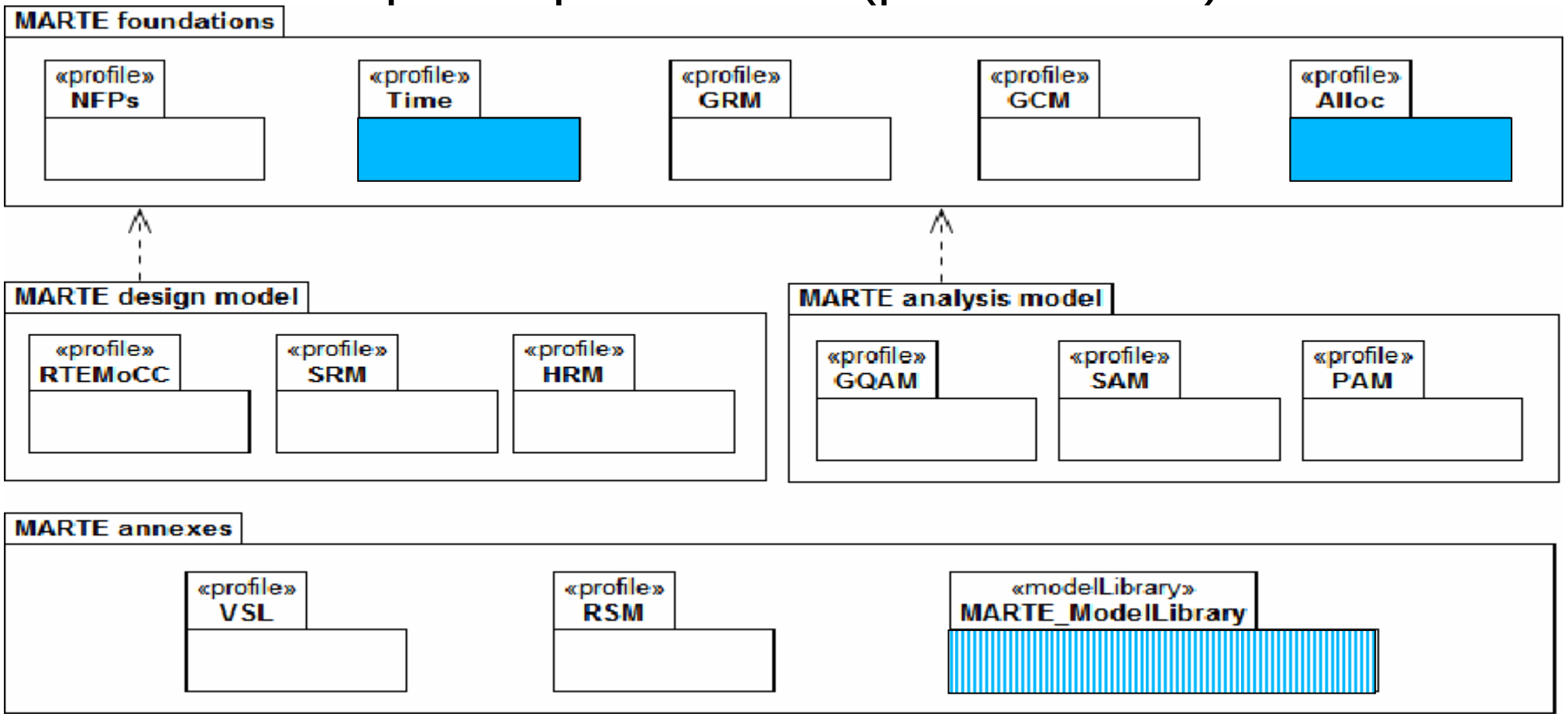
SAFA – 5 décembre 2007

AADL

- **A**rchitecture **A**nalysis & **D**esign **L**anguage
  - Avionics/Automotive standard from **S**ociety of **A**utomotive **E**ngineers.
  - Design & Analysis of performance-critical RT systems
- Application/Software Components
  - Thread, Thread Group, Process, Subprogram
- Execution Platform Components
  - Bus, Device, Processor, Memory
- Binding
  - Bind Application onto Execution Platform

MARTE

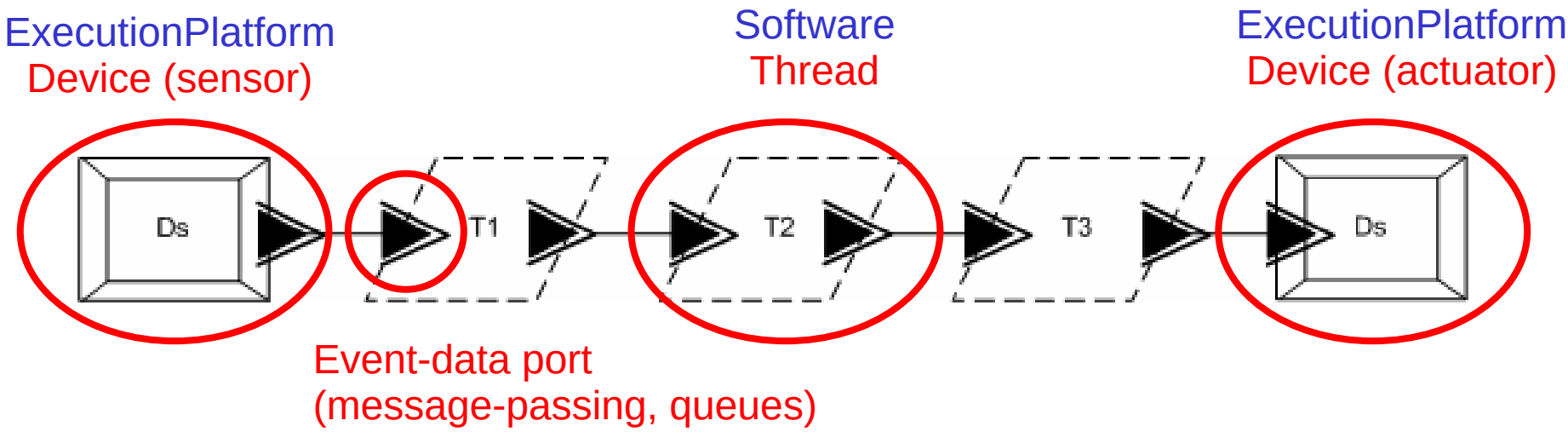
- OMG UML2 Profile for **M**odeling and **A**nalysis of **R**eal-**T**ime and **E**mbedded systems
  - OMG Adopted Specification (ptc/07-08-04) => FTF

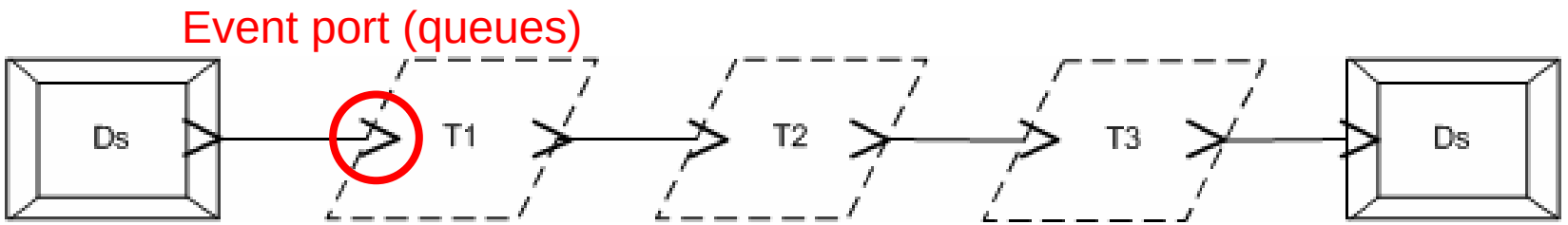


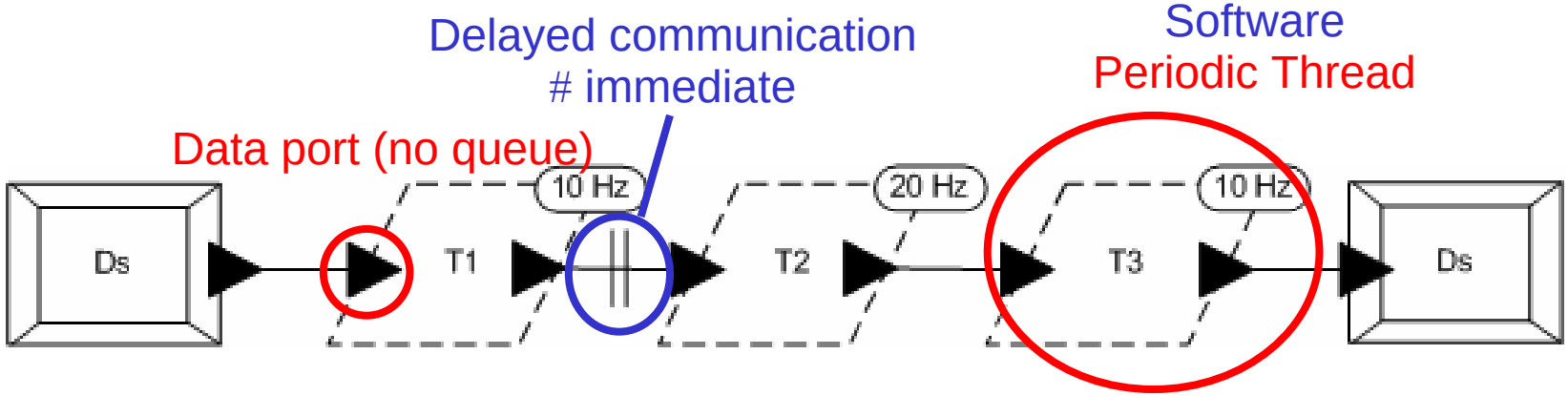
MARTE

- OMG UML2 Profile for **M**odeling and **A**nalysis of **R**eal-**T**ime and **E**mbedded systems
  - OMG Adopted Specification (ptc/07-08-04) => FTF
- Time
  - Define a Timed Causality Model for UML
  - Broad enough to cover several Models of Computation (exercise: Model AADL MoCC)
- Allocation sub-profiles
  - Describe various possible allocations (and their costs)
  - Time analysis needs to cope with communication costs and context switching costs, etc.

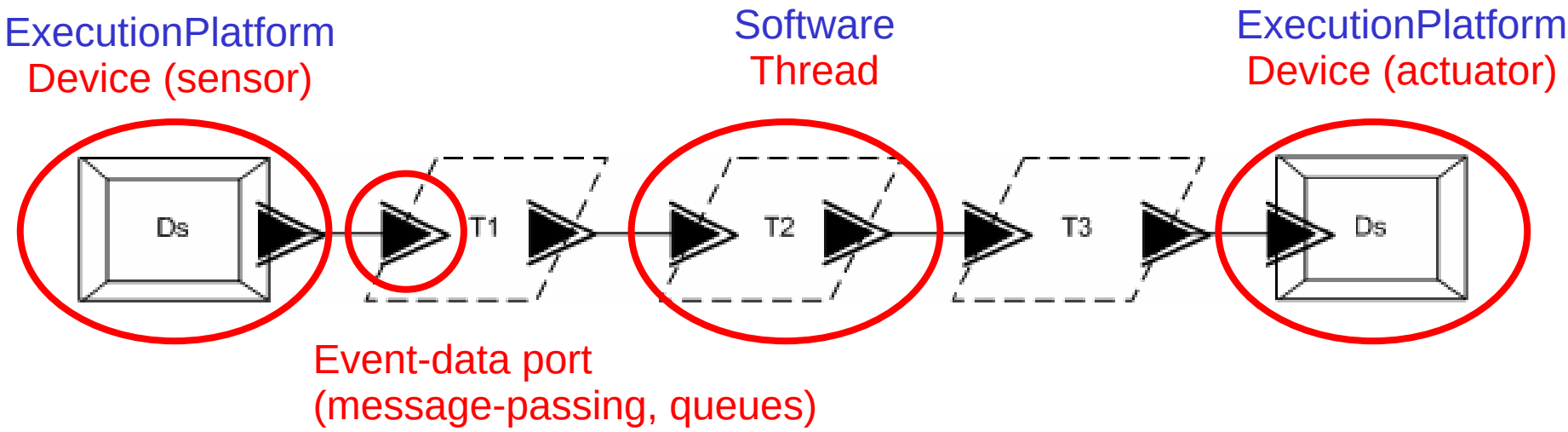
An example in AADL



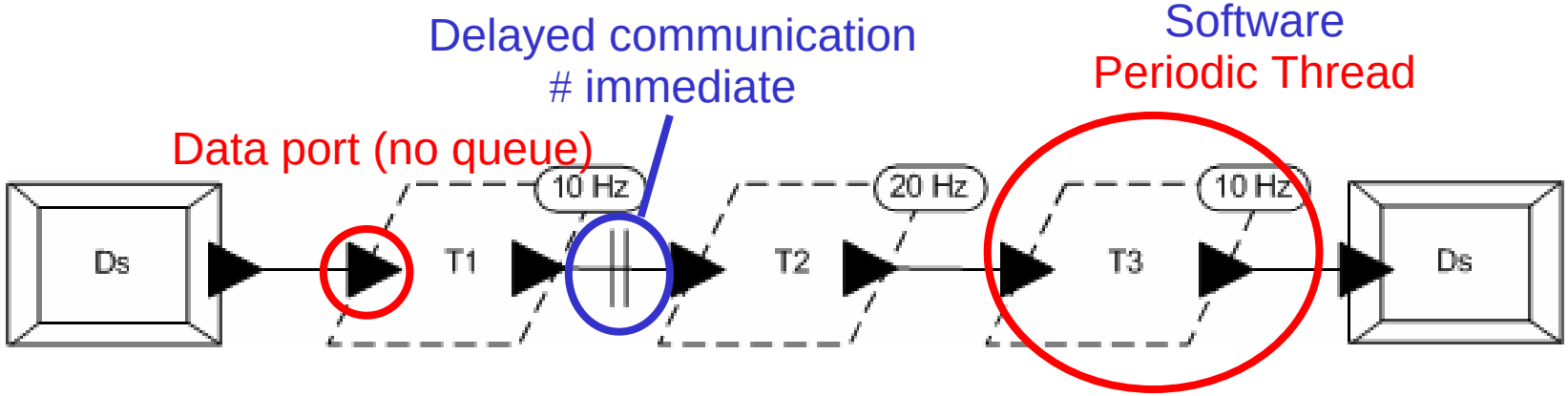




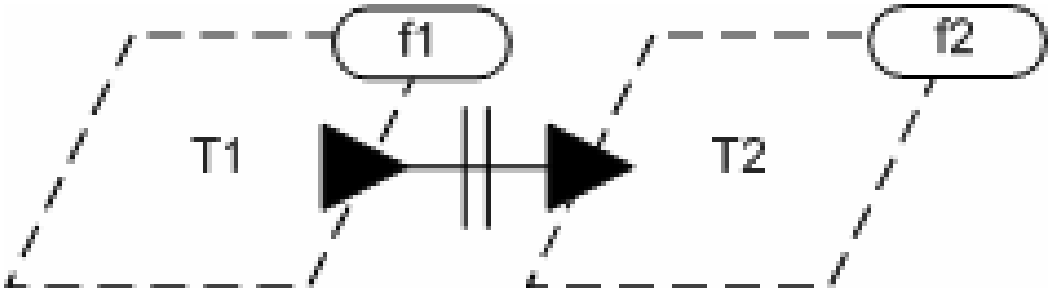
An example in AADL



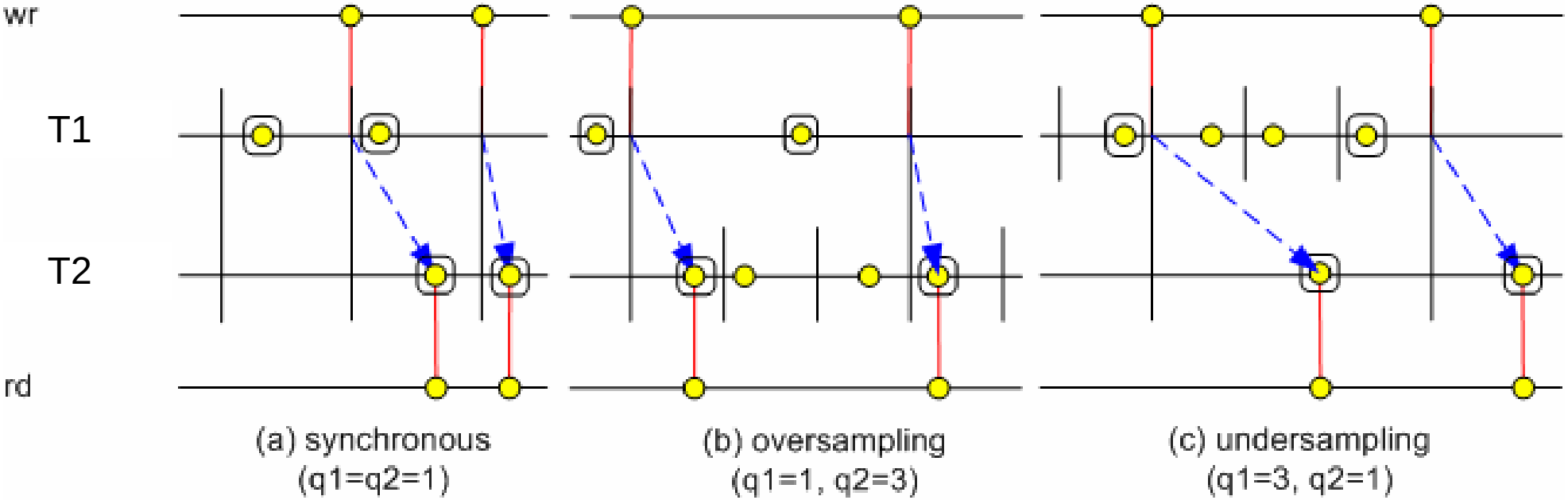
- Mix of **Execution Platform** and **Software**
- **Case-based** definition of the underlying **Model of Computation and Communication (MoCC)**



Delayed Communications

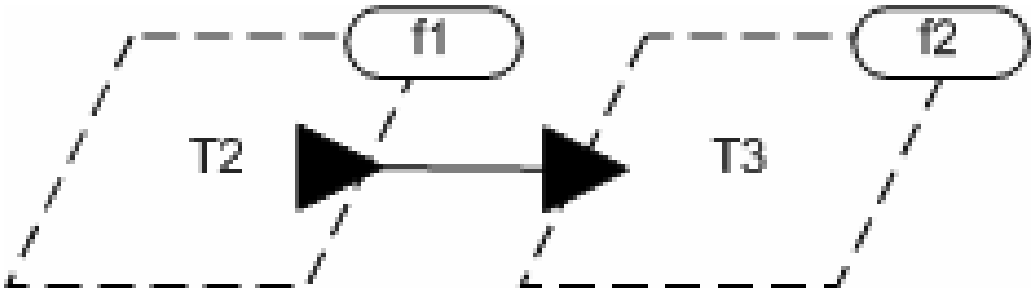


f1=q1*f
f2=q2*f

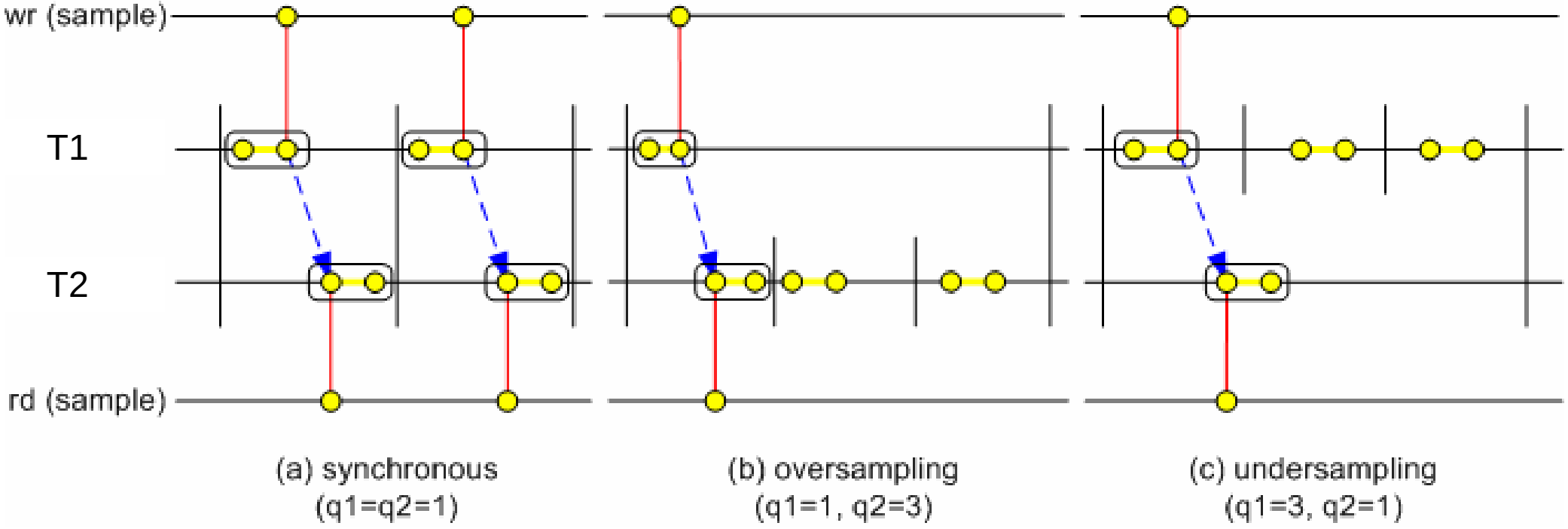


- Communication à travers un *latch* synchrone

Immediate Communications

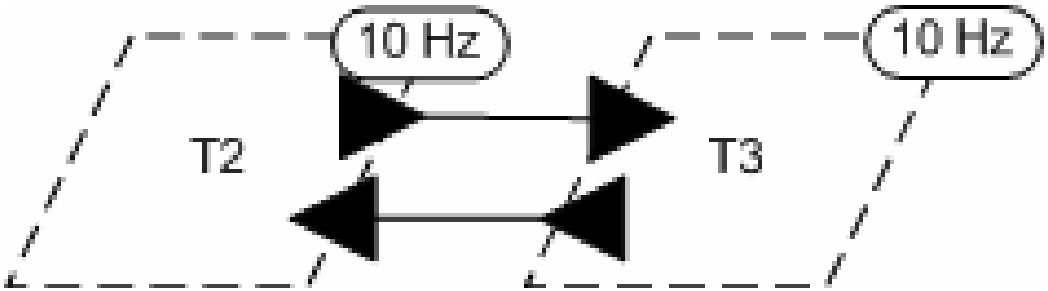


f1=q1*f
f2=q2*f



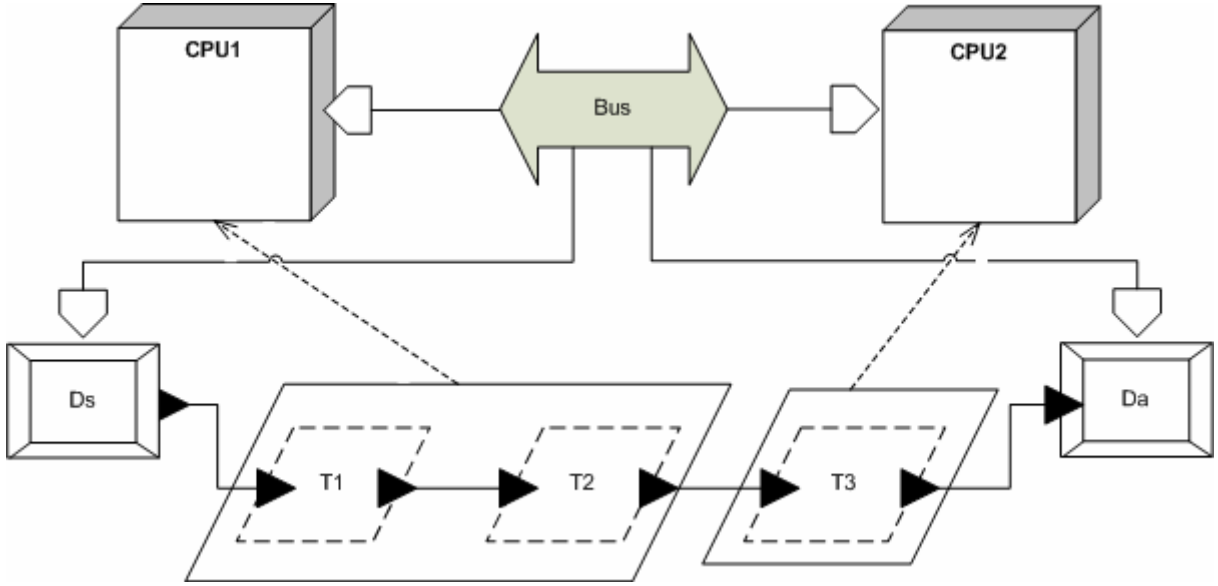
- Communication « instantanée »

Causality problems

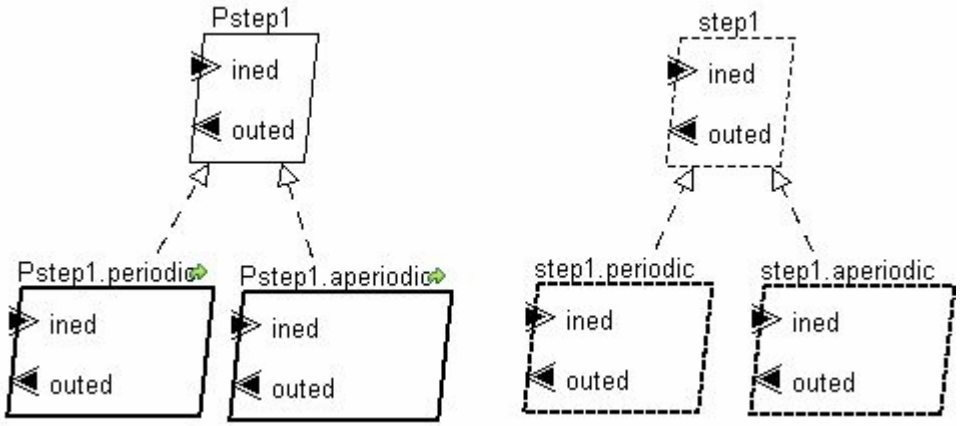


- To be deterministic, one must
  - Perform clock computations
  - Define a schedule
  - Compute a Fix-point
- **How to deal with the causality problems ?**
- Synchronous languages !

A simple sketch hides a complex model



- Most of the model is hidden as text in AADL (formerly MetaH)

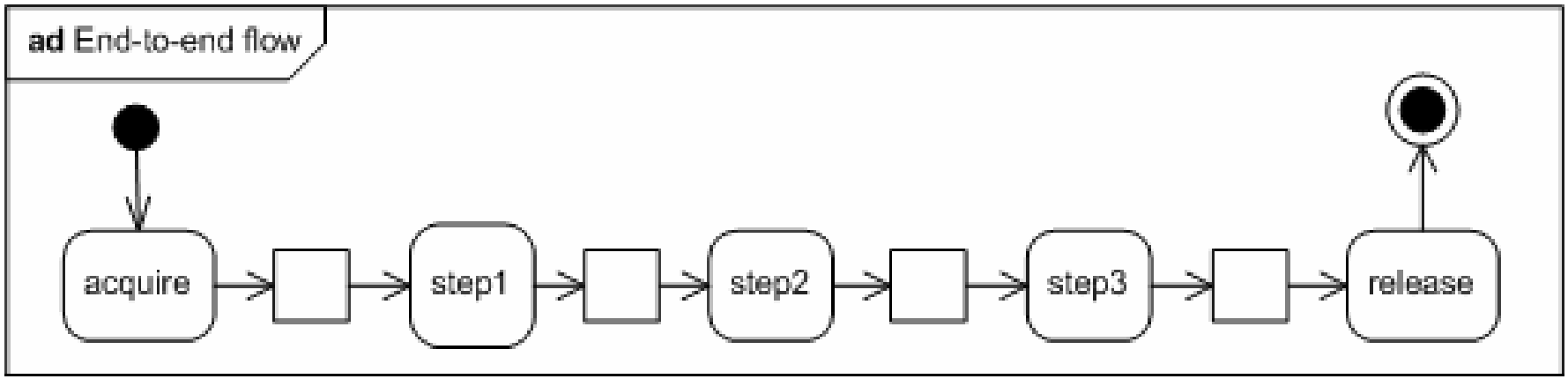


Summary of Issues (and solutions)

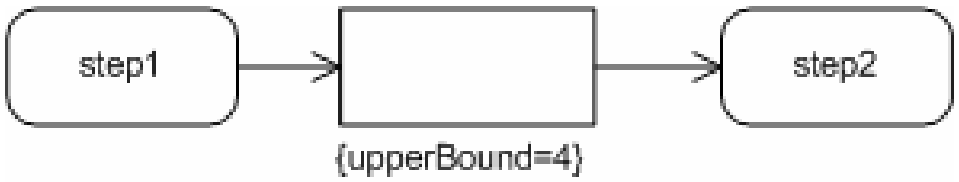
- Mix of Execution Platform and Software
  - Separate them, use Allocation to associate costs
- Case-based definition of the underlying Model of Computation and Communication (MoCC)
  - Make It Explicit
- Do not deal with the causality problems
  - Use well-known solutions
  - … from synchronous languages
- Most of the model is hidden as text
  - Make the time a first-class citizen

UML and MARTE

- Application (pure causal/untimed relations)
  - UML Activity Diagrams



  - Different queuing policies (event, event-data, data) ?

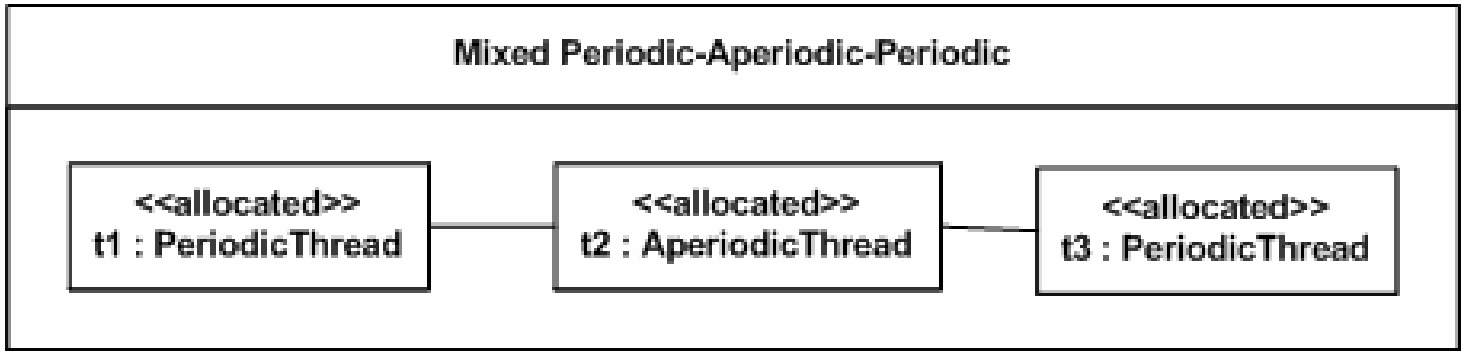


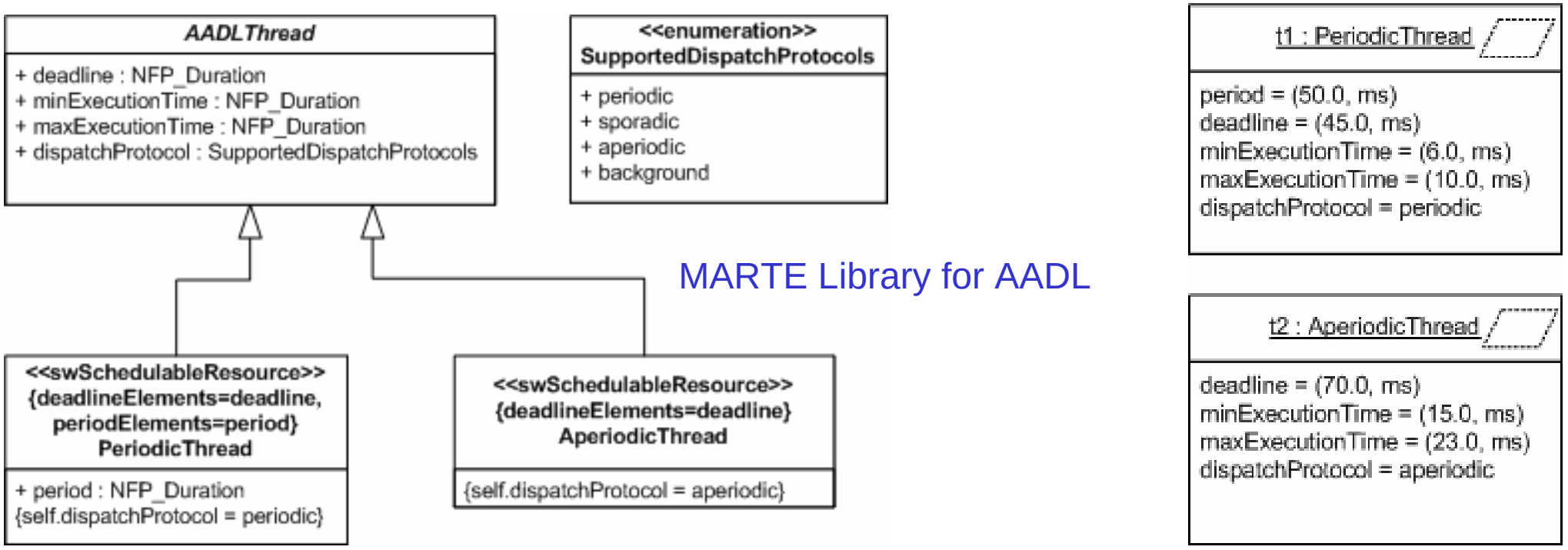
With queues (event or event-data)

Without queues (data)

UML and MARTE

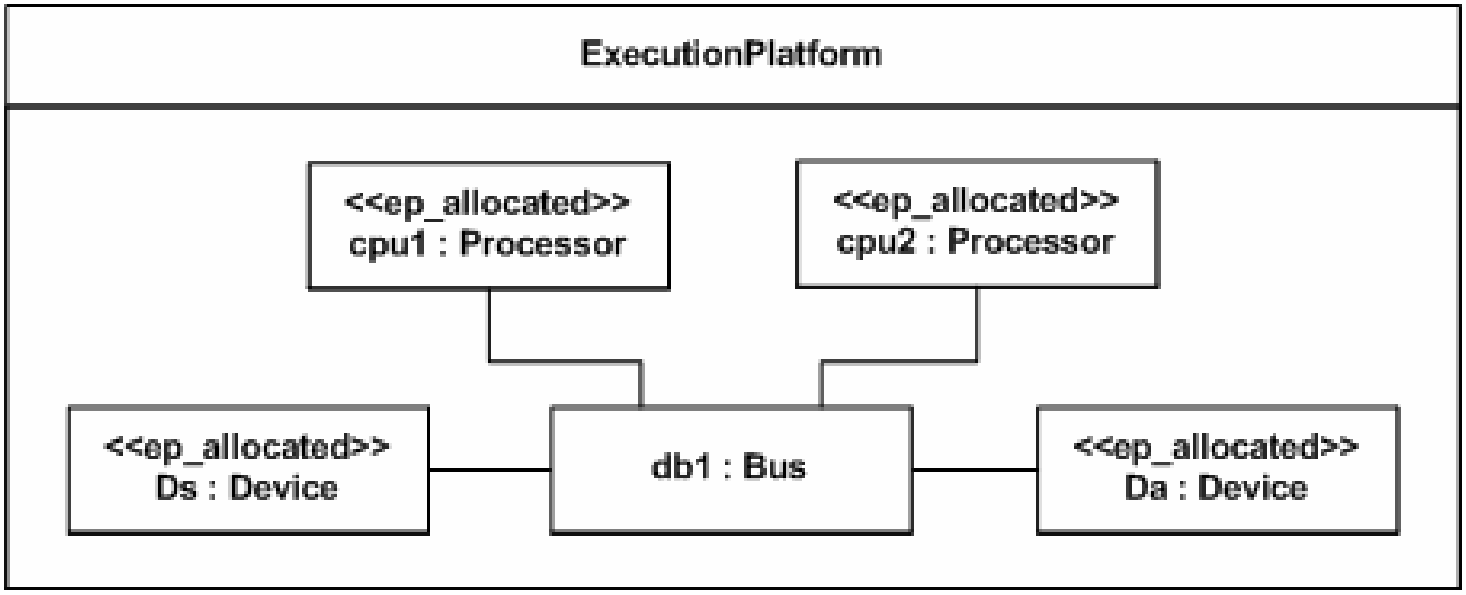
- Software Resources (OS, middleware, …)
  - UML Composite Structure Diagrams

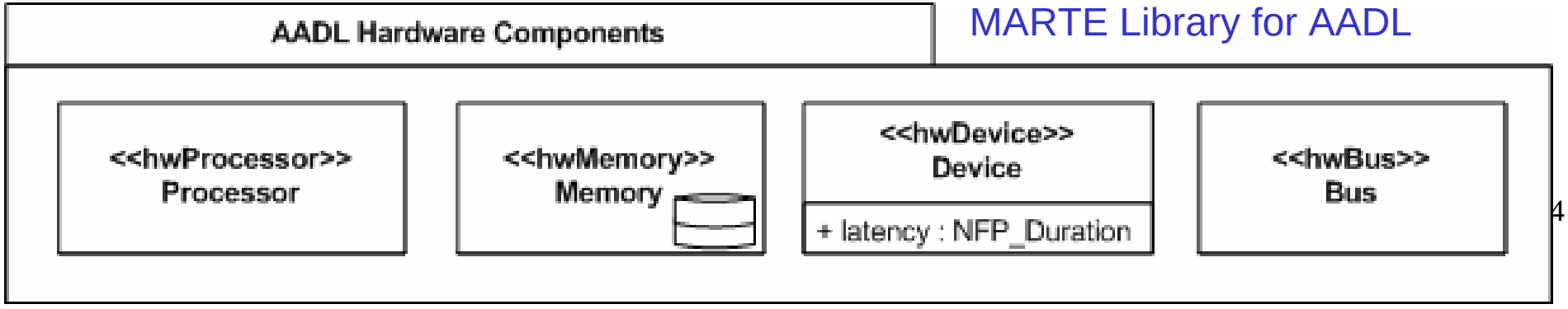




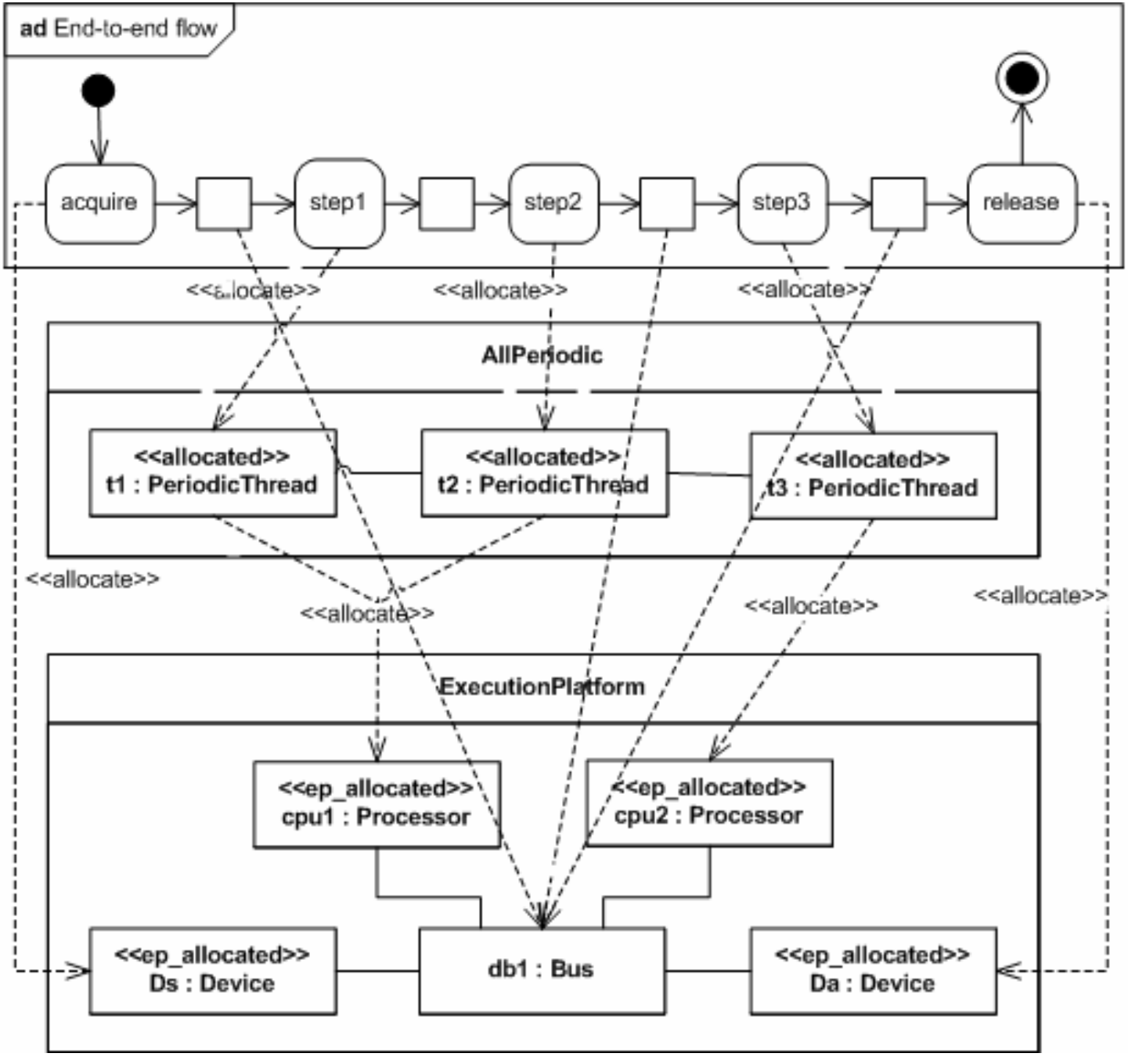
UML and MARTE

- Execution platform
  - Composite Structure Diagrams



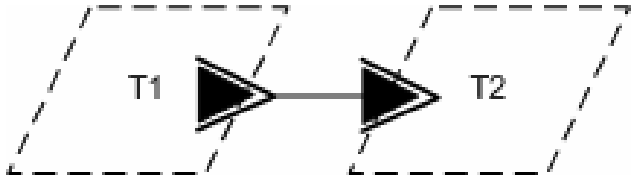


Allocation in MARTE

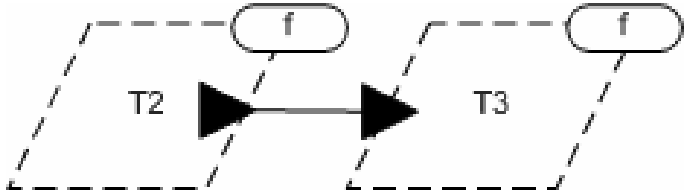


MoCCs with MARTE

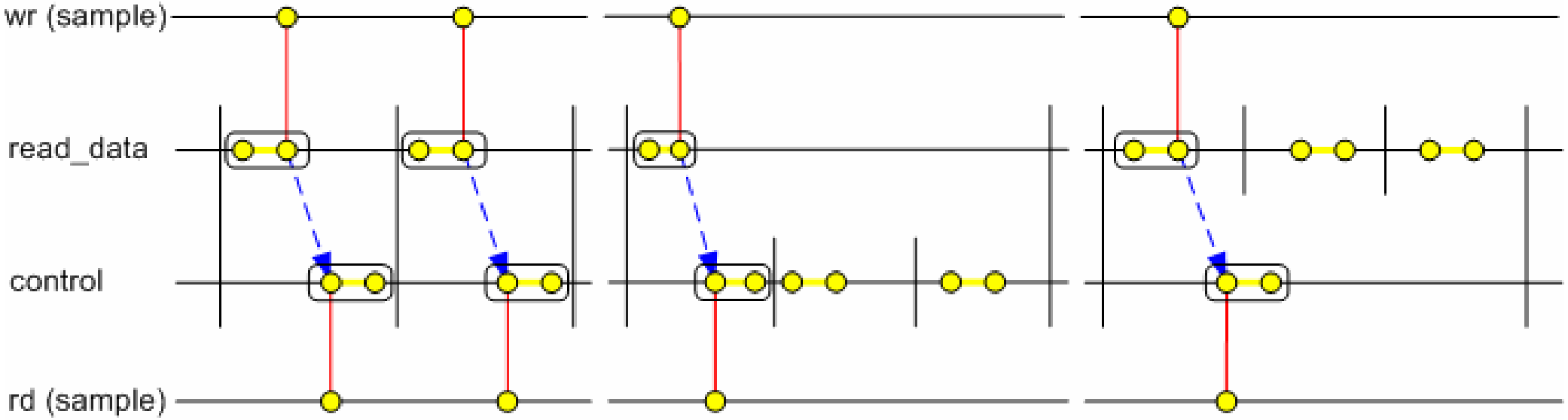
- Overall only two different kinds of communications in AADL
  - Dataflow



Event-data ports + Aperiodic threads



Data ports + immediate Periodic threads



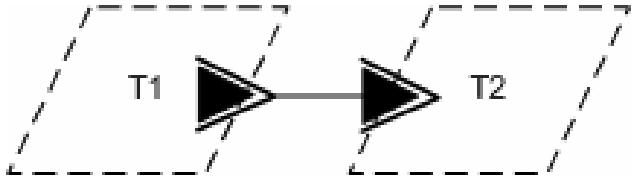
(a) synchronous (q1=q2=1)

(b) oversampling (q1=1, q2=3)

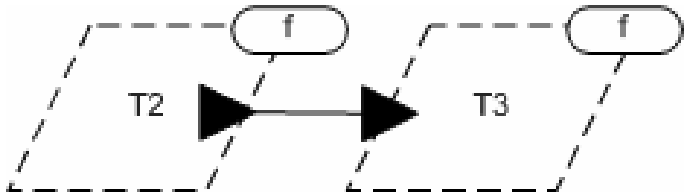
(c) undersampling (q1=3, q2=1)

MoCCs with MARTE

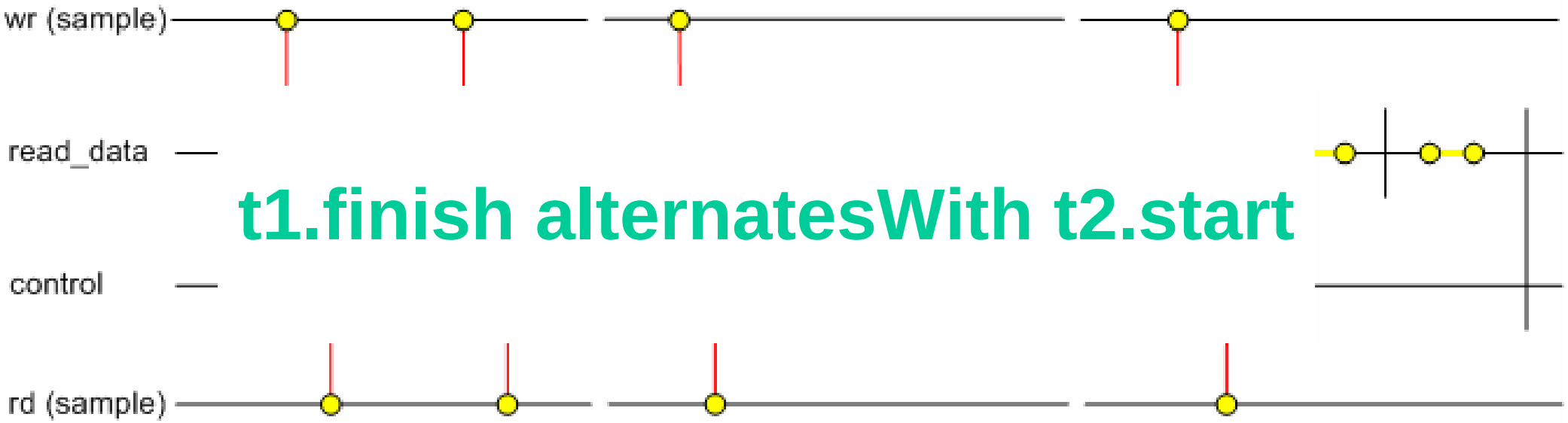
- Overall only two different kinds of communications in AADL
  - Dataflow



Event-data ports + Aperiodic threads

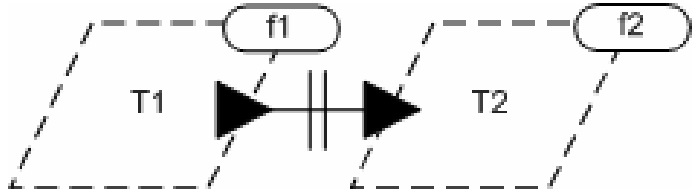


Data ports + immediate Periodic threads

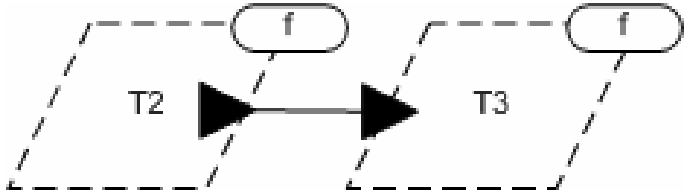


MoCCs with MARTE

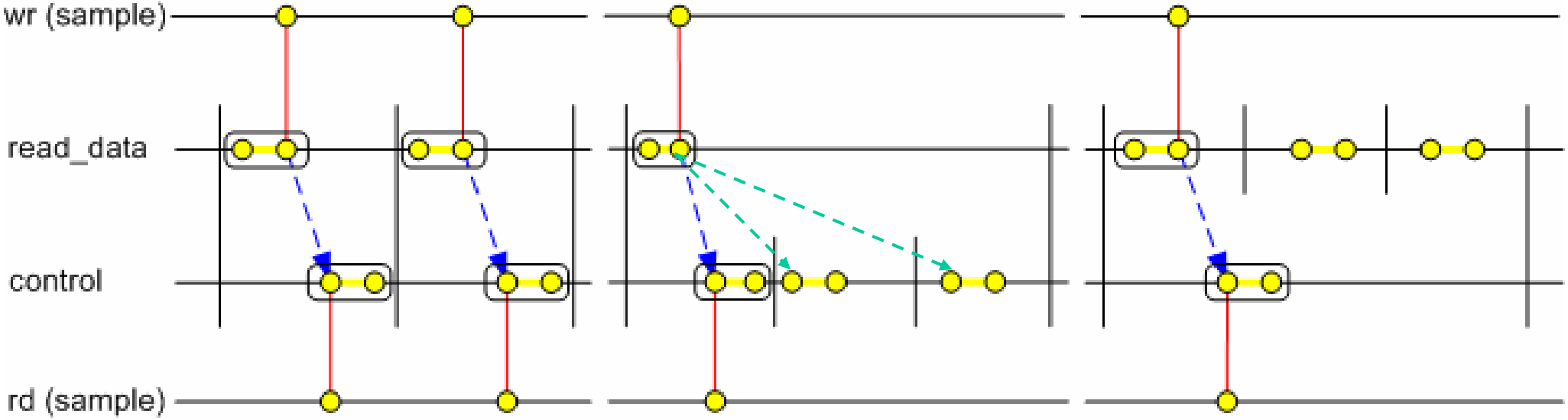
- Overall only two different kinds of communications in AADL
  - Sampled



Data ports + Delayed + Periodic threads



Data ports + immediate Periodic (oversampling) threads



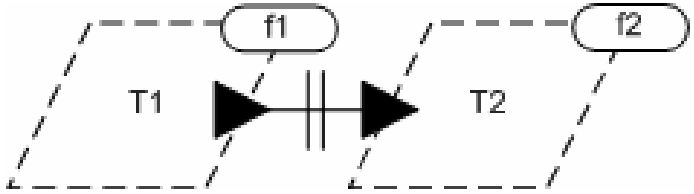
(a) synchronous (q1=q2=1)

(b) oversampling (q1=1, q2=3)

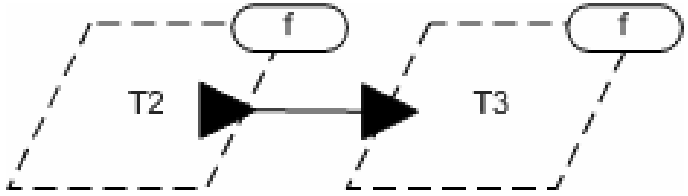
(c) undersampling (q1=3, q2=1)

MoCCs with MARTE

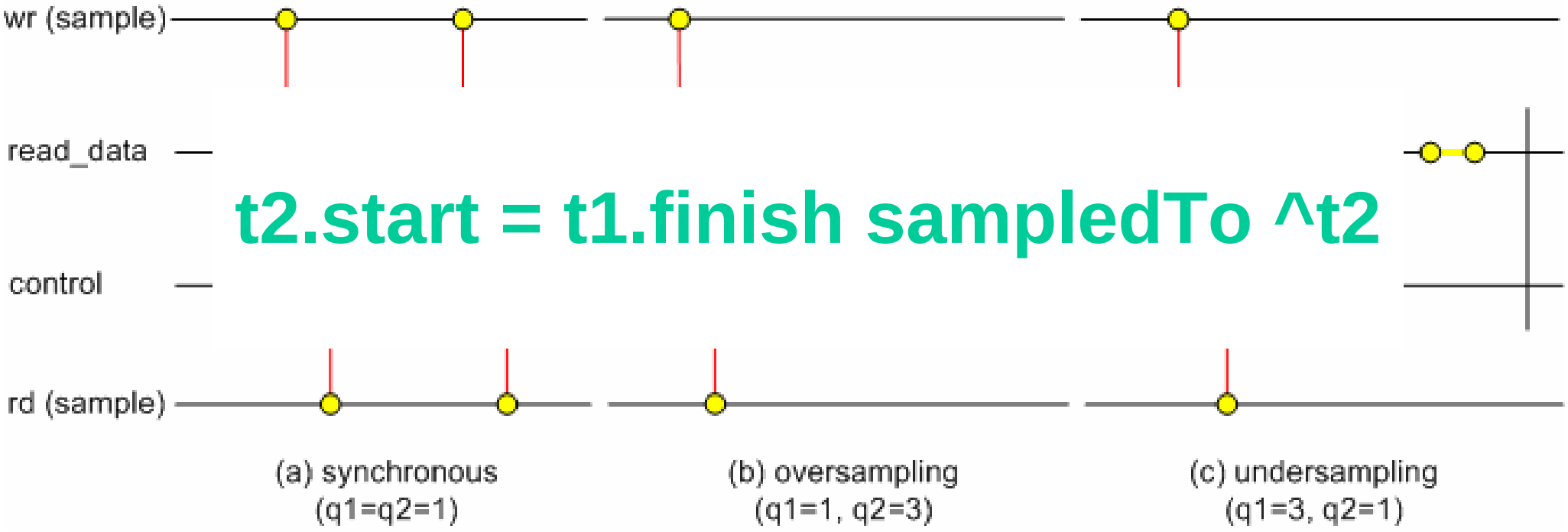
- Overall only two different kinds of communications in AADL
  - Sampled



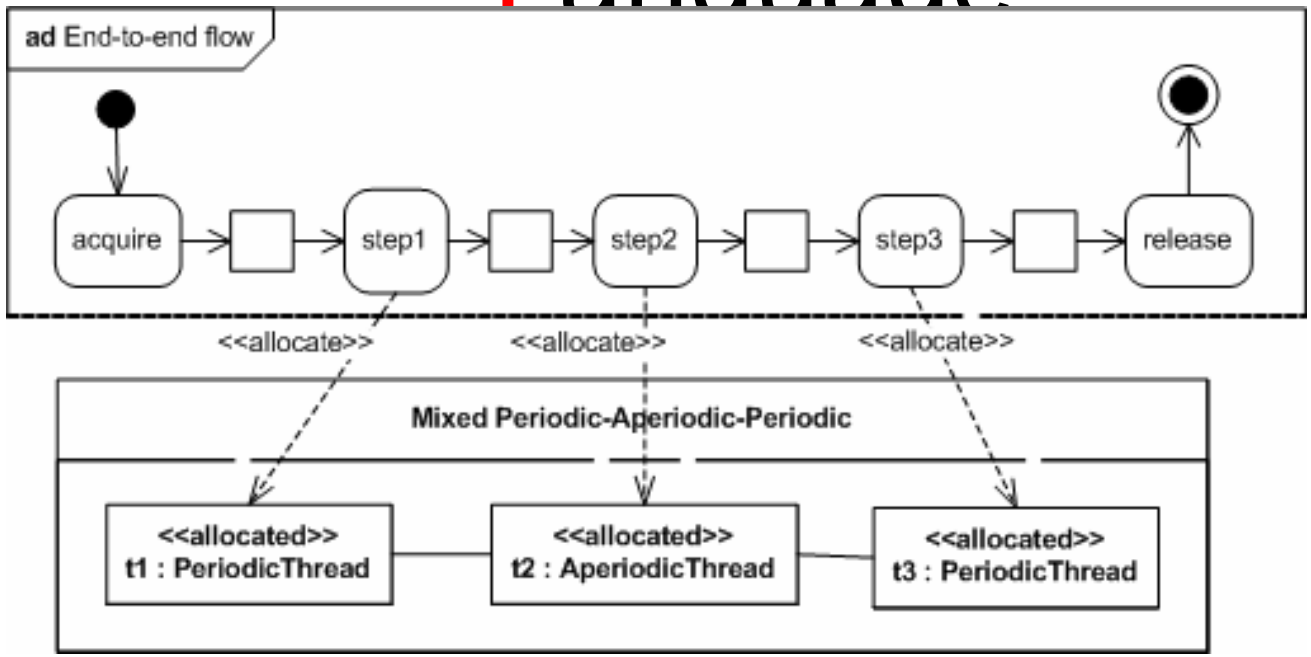
Data ports + Delayed + Periodic threads

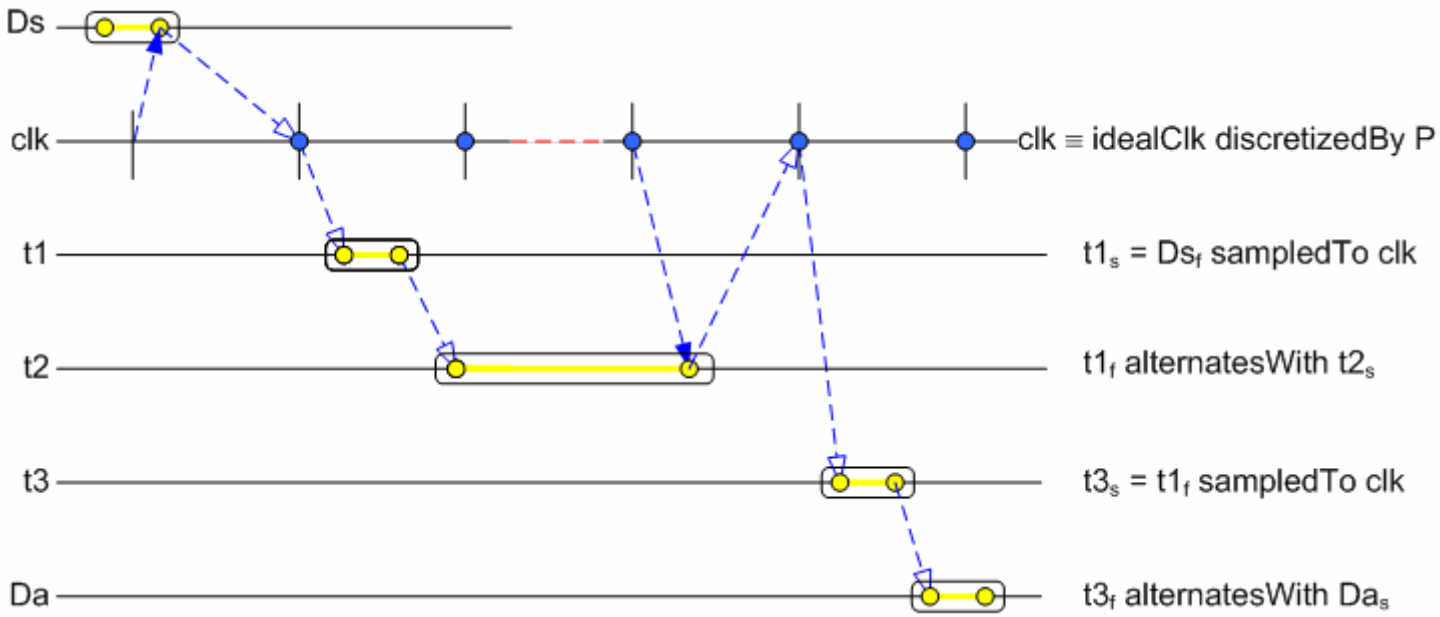


Data ports + immediate Periodic (oversampling) threads



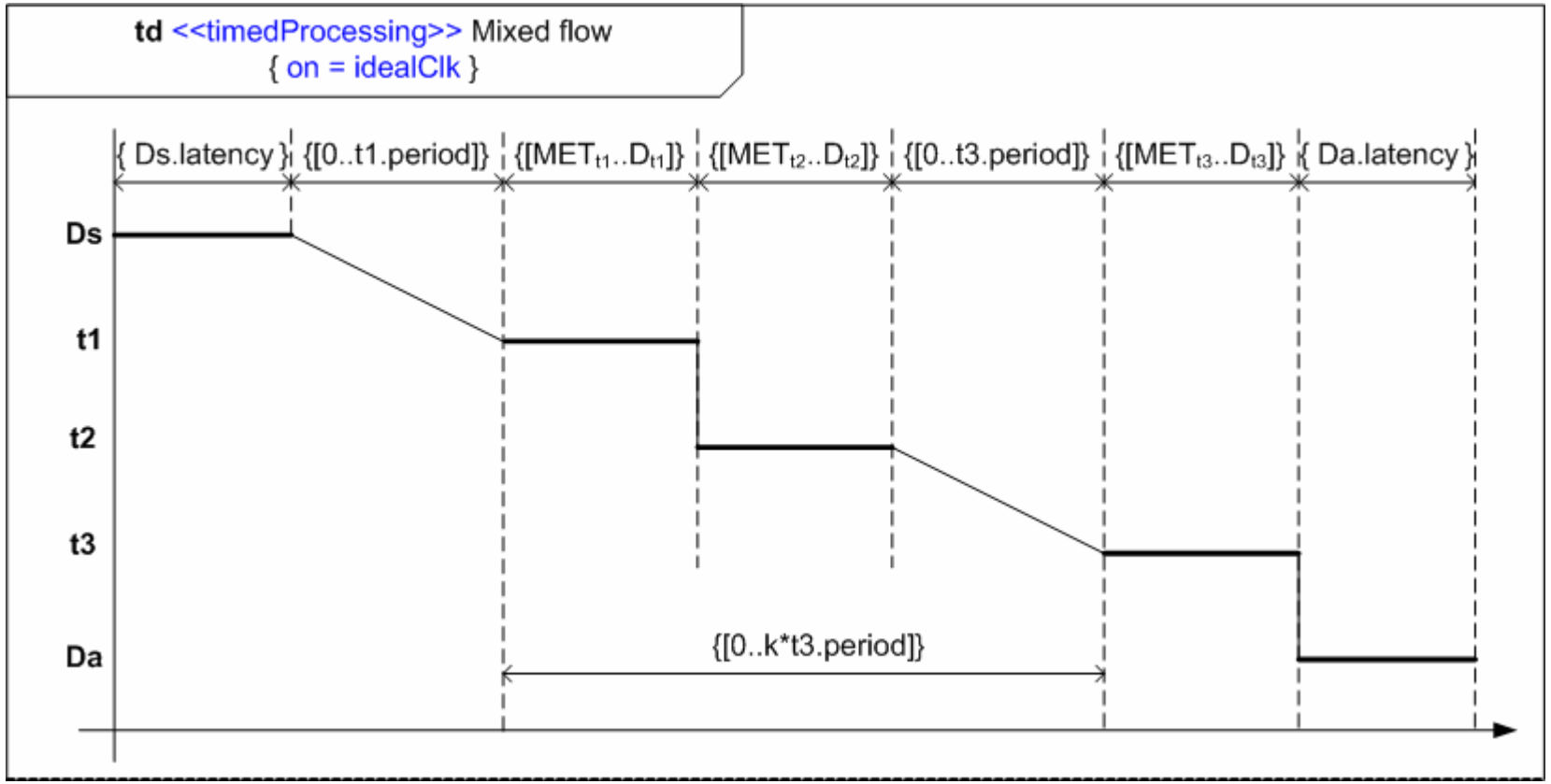
Clock Constraint Specification Language





Timing analysis results

UML Timing Diagrams



End-to-End Flow Latency = Ds.latency + flow latency + Da.latency

Flow latency$_{Worst\text{-}Case}$ = t1.period + k1 * t3.period + t3.deadline

Flow Latency$_{Best\text{-}Case}$ = t1.period + k2 * t3.period + t3.MinExecTime (k2≤k1)

Latency jitter = t3.deadline – t3.MinExecTime + (k1-k2)*t3.period

Conclusion

- MARTE is under revision by FTF
  - Public issues up to December, 21st
- MARTE annex A
  - Guidelines about AADL and East-ADL2 (AutoSAR)
- MARTE as a language to support MoCC definition
- Give a Timed Causality Model to UML