UML MARTE

www.omgmarte.org

Logical Time : Specification vs. Implementation

Frédéric **MALLET**, Charles ANDRE, Robert de SIMONE

AOSTE

Université **N**ice **S**ophia Antipolis

I3S Laboratory – UMR 6070 CNRS

INRIA Sophia Antipolis Méditerranée

i3S sophia antipolis

Université Nice Sophia Antipolis

cnrs

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche SOPHIA ANTIPOLIS - MÉDITERRANÉE

(Logical) Time requirements

What do we need ?

- Causality (asynchrony): a **causes** b (a  b)
  - Temporal sequentiality: a **precedes** b (a  b)
  - Temporal simultaneity: a **coincides with** b (a ≡ b)
- Bounded causality :
  - Bound = 1: a **alternatesWith** b (a ~ b)
- Relative time patterns:
  - Periodicity |a[i+1] - a[i]| = *period*
  - Sporadicity |a[i+1] - a[i]| > interval
  - Deadline |end[i] - start[i]| < deadline
  - Jitter, skew, …

5 ms
5 meters
5 processor cycles

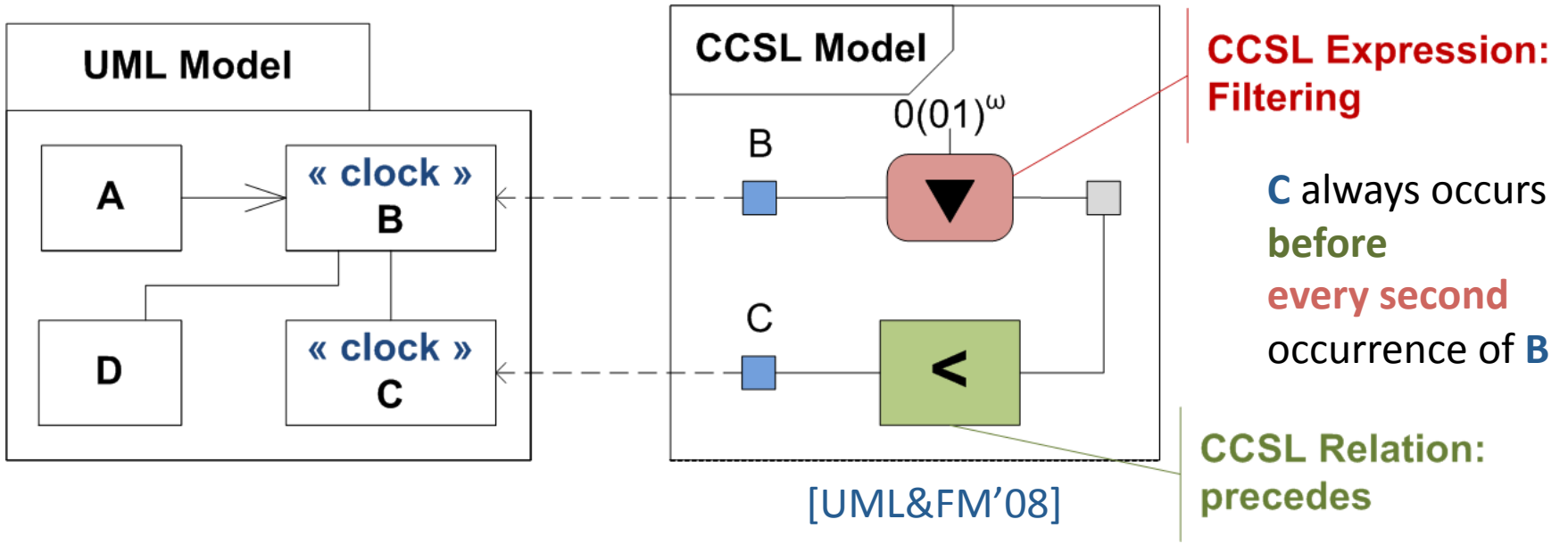
The UML Profile for MARTE

- UML2 Profile adopted by the OMG in 2009
  - **M**odeling and **A**nalysis of **R**eal-**T**ime and **E**mbedded systems
- Allocation Model
  - Extends SysML allocation
- **Time model** to capture
  - Causal relationships
  - Chronological relationships
  - Timed constraints

Logical Time

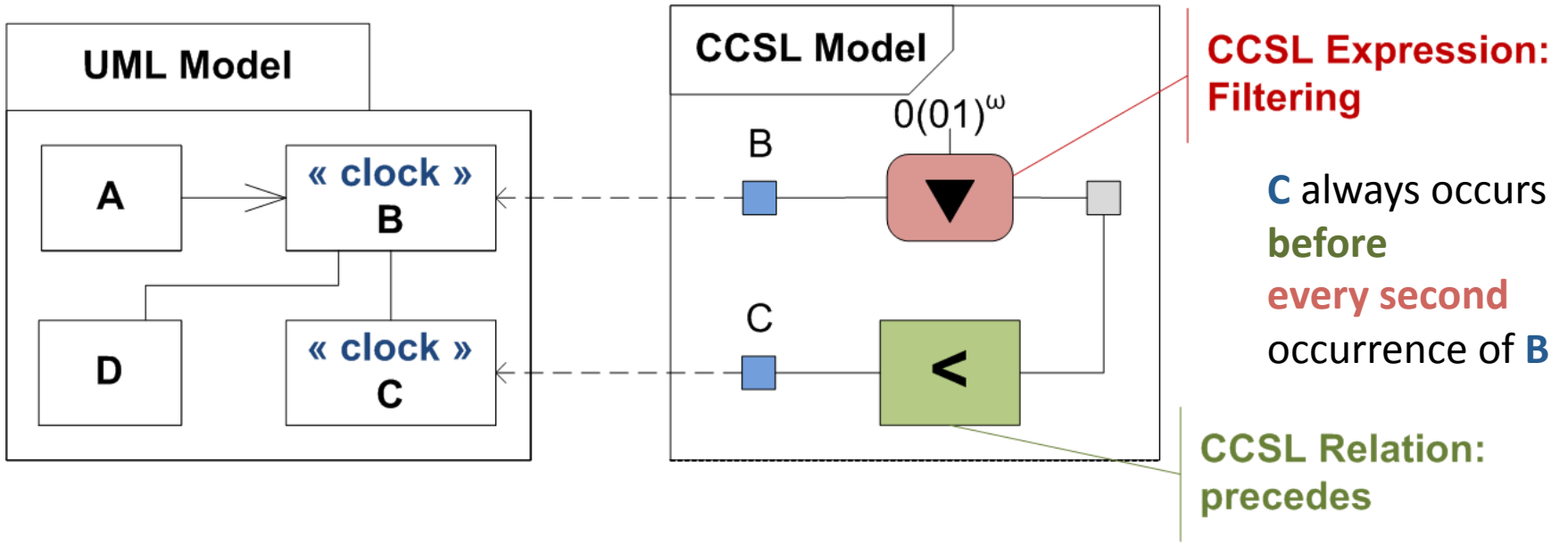
E.g., The car must stop within **5 ms** and before **10 m**

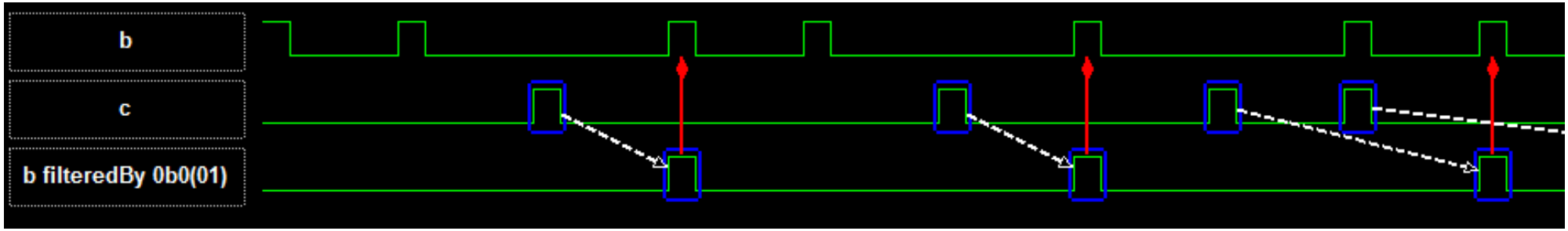
Adorns UML models
with a Timed Causality semantics



[UML&FM'08]

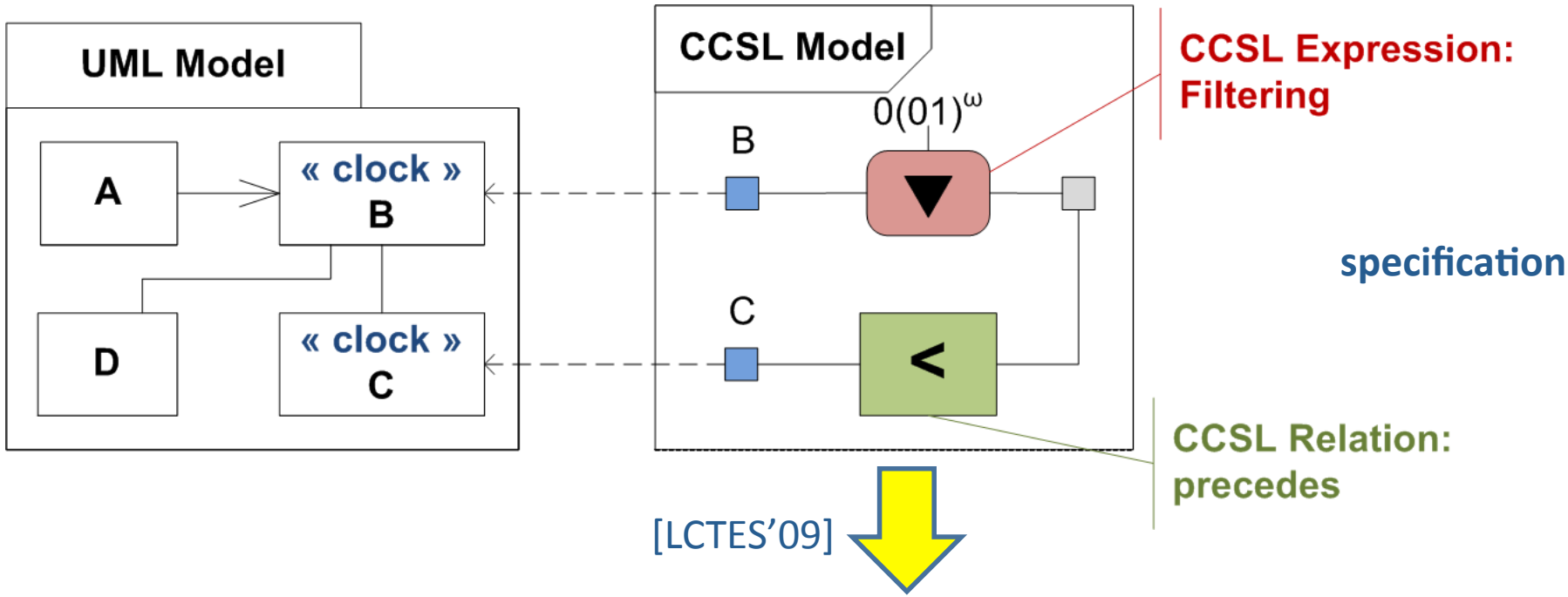
Adorns UML models
for simulation and animation

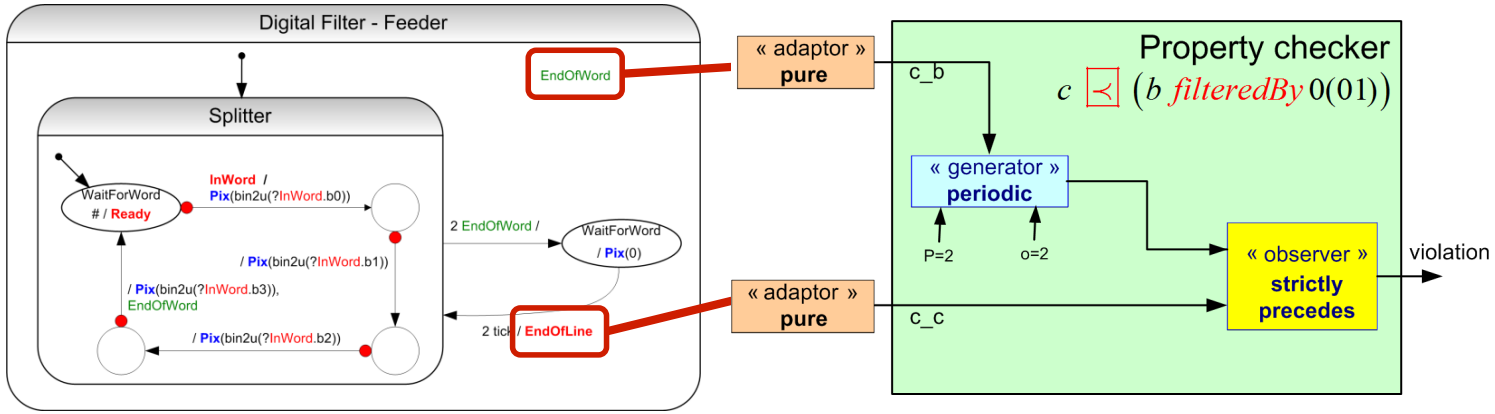




TimeSquare

Adorns UML models
for specification

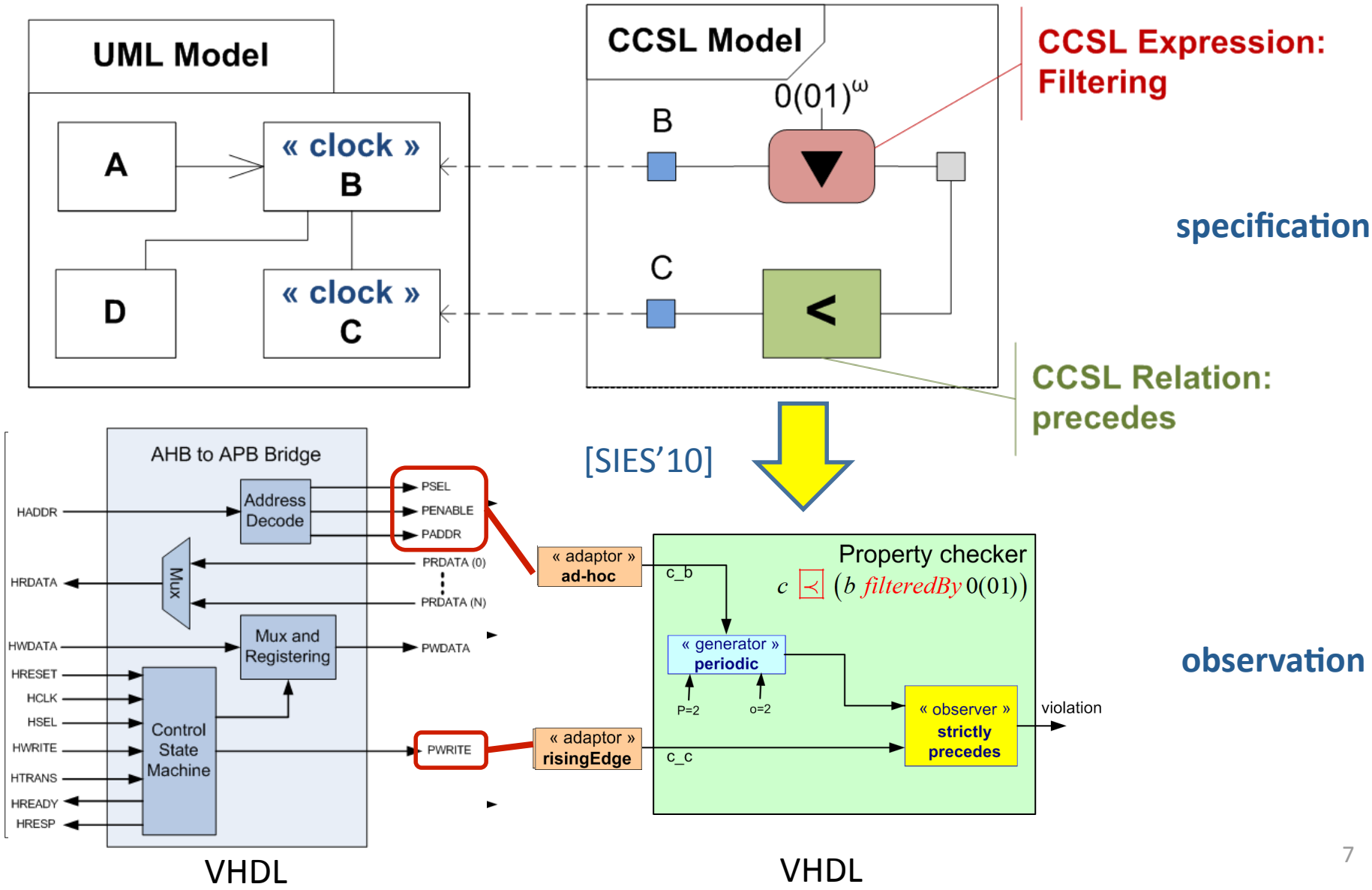




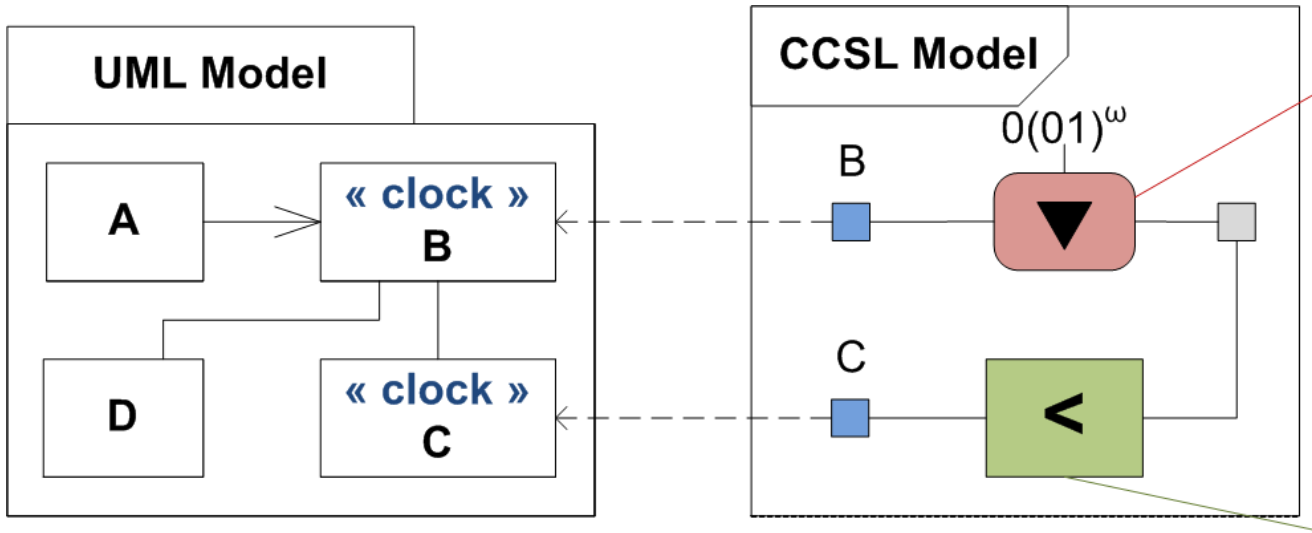
Esterel/SyncCharts

Esterel/SyncCharts

Adorns UML models
for specification

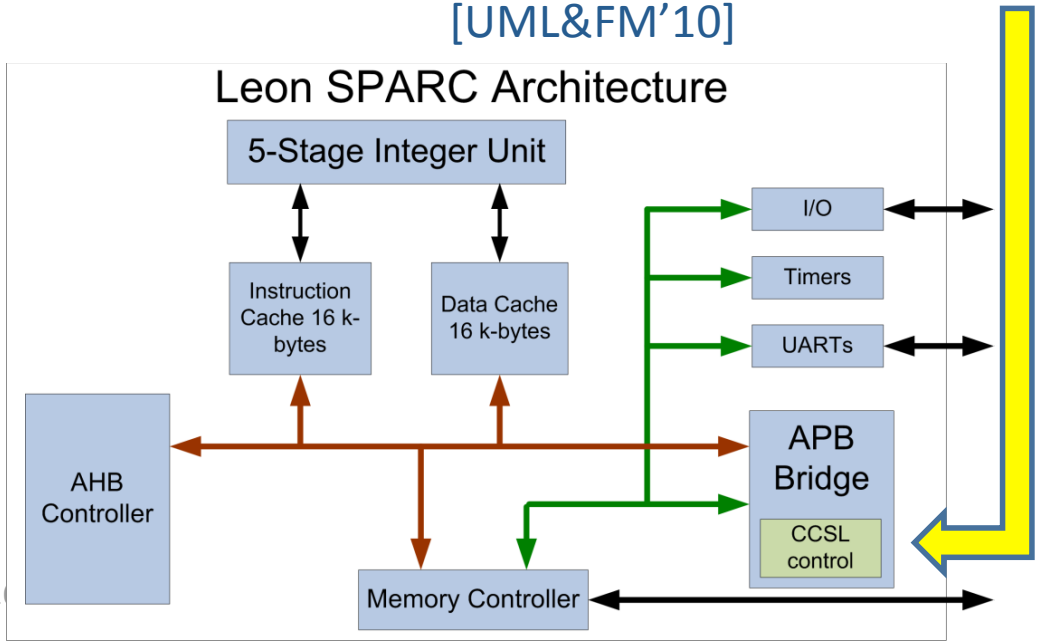


Adorns UML models
for code generation



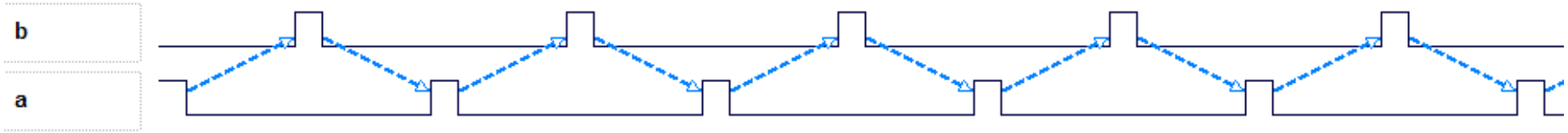
[UML&FM'10]

specification

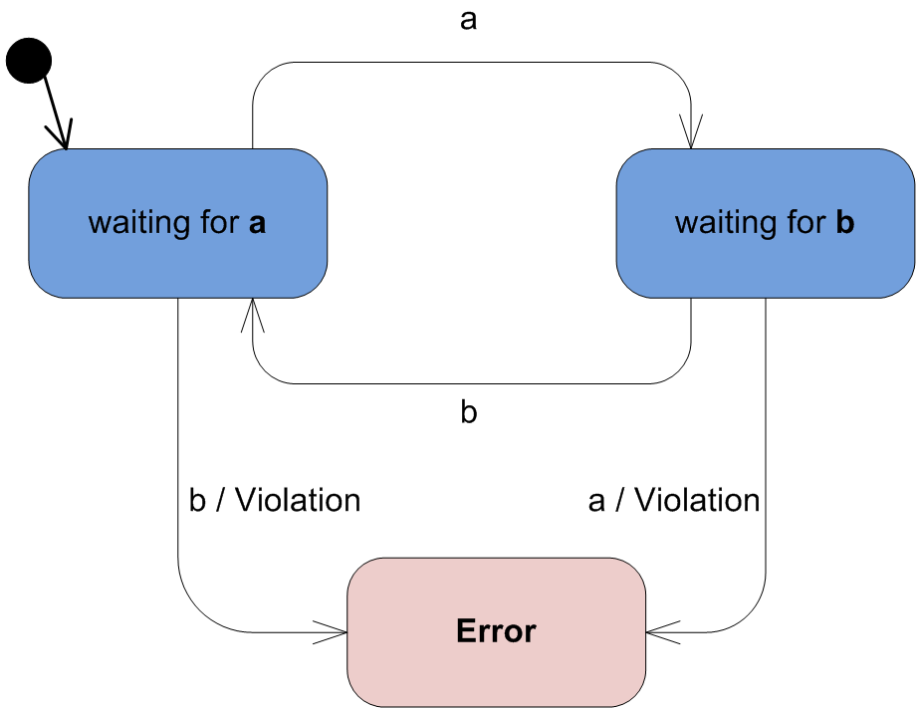


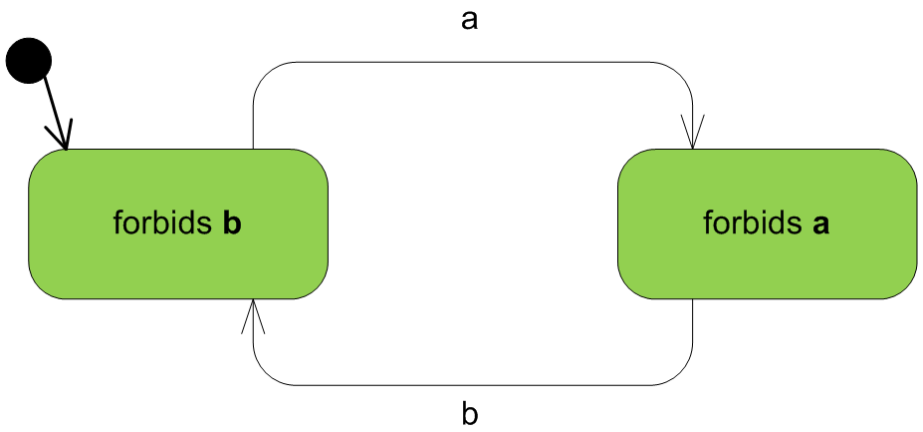
implementation

What are the difficulties ?



Specification/observation **Implementation/Code generation**

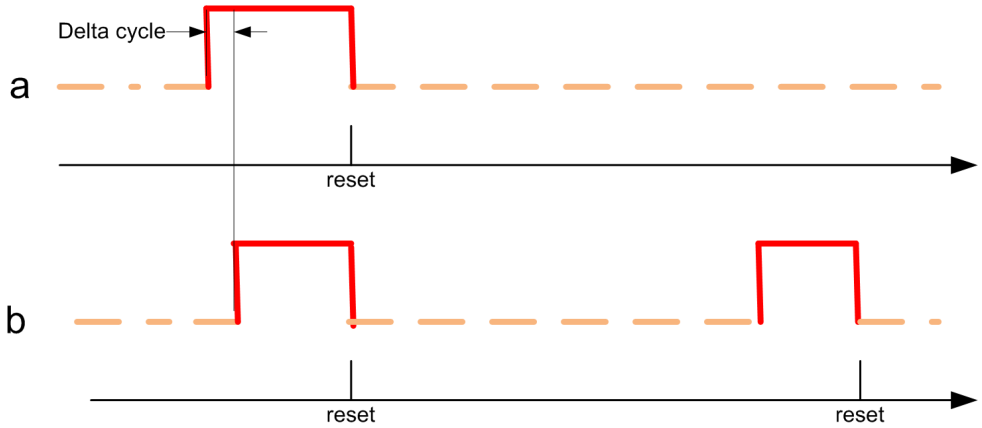




What are the difficulties ?

Specification/observation

- Esterel
  - Easy encoding
  - Fixed-point semantics
- VHDL
  - No built-in coincidence
  - Have to use simulation time
  - Causality cycles not detected



Implementation/Code generation

- Esterel
  - No non-deterministic choice (required if underspecified)
- VHDL
  - No built-in coincidence
  - Delta-cycle simulation semantics
    - May cause false violations
    - Glitch-insensitive implementation
  - Postponed processes
    - Modularity is NOT possible

Is ‘a’ a subclock of ‘b’ ?

Conclusion

- Our goal
  - Express **causal** and **temporal** properties: CCSL
    - Combine *asynchronous* and *synchronous* operators
  - Combine/Encode CCSL in other formalisms
    - Compare the expressiveness
    - Reuse existing analysis tools (exhaustive verification)
  - Problem addresses
    - Encoding of CCSL observers
    - Encoding of CCSL operators

Future work

- Exhaustive Verification of CCSL
  - **Observers:** verify that a model conforms to a CCSL specification
  - **Code generation:** verify that a CCSL model has the expected properties
    - On-going works: PSL/Sugar, Promela
    - How to encode the coincidence in asynchronous languages ?
  - Restrict to a bounded subset of CCSL
    - No unbounded precedence


大上海时代广场
SHANGHAI TIMES SQUARE
谢
谢

Time model – Clock relations

- *Clock relations* constrain the occurrences of *selected* events

- Four categories of clock relations
  - **Precedence-based** (asynchronous) - causality
    - precedes, **alternatesWith** …
  - **Coincidence-based** (synchronous)
    - isSubClock, discretizedBy, **isPeriodicOn**, filteredBy …
  - **Mixed** (asynchronous => synchronous)
    - sampledOn, **delayedFor**, timer, inf, sup …
  - **Quantitative** (related to chronometric clocks)
    - hasStability, hasOffset, hasJitter, hasDrift …

Petri nets

Esterel, Signal, Lustre

Polychronous, Quasi-synchronous, GALS

- Clock Constraint Specification Language = concrete syntax
  - Annex of MARTE