MARTE Time Model & SysML

Aoste Project
Frédéric Mallet
June 2008

Different types of clock

- **ClockType** extends Class
  - logical/chronometric, discrete/dense
  - unitType: Enumeration (set of acceptable units)
  - Labelling function (to any domain, even continuous)

- Tagged Signal Model => comparison of Models of Computation and Communications (MoCCs)
- Support multiform time and multiclock systems

Set of related clocks

- **Clock** extends Property/InstanceSpecification
  - Conform to a given ClockType
  - one unit
- **TimedElement**
  - Elements (Event, ValueSpecification, Observation, Constraint) associated with one or several clocks
- **TimeStructure**
  - Set of *a priori* independent clocks
  - Set of instant/clock constraints

➢ Partial ordering of instants: concurrency/trace theories

Two intended usages

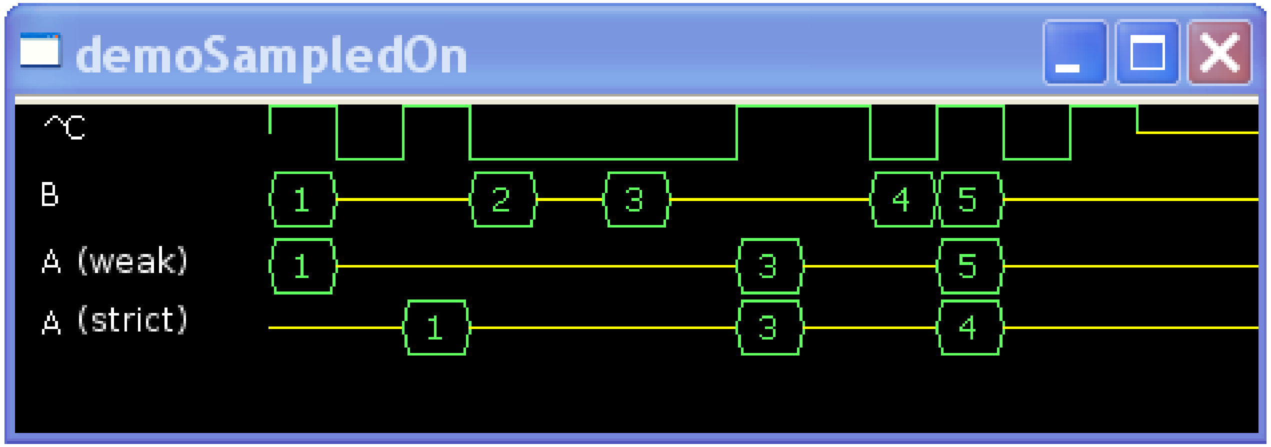
- **For end-user**
  - Use clocks from a predefined library
  - Apply constraints
    - The **C**lock **C**onstraint **S**pecification **L**anguage gives a OCL-like (declarative, non causal) syntax
    - Could use SysML parametric diagrams
- **For MoCC designer**
  - AADL, Synchronous/reactive, AutoSAR, …
  - Use advanced topics

MARTE **C**lock **C**onstraint **S**pecification **L**anguage

- Coincidence-based constraints (synchronous)
  - Subclock (less frequent than)
    - Equality
    - Restricted to (according to a predicate)
    - Filtering (according to k-periodic patterns)
- Precedence-based constraints (asynchronous)
  - Precedence, alternation, speed
- Mixed constraints
  - Sampling, watchdog, …

CCSL semantics

$$A = B \ \mathsf{sampledOn}\ C \ (A = B \between C) \iff$$
$$(\forall a \in \mathbb{N}^{\star})(\exists b, c \in \mathbb{N}^{\star})((A[a] \equiv C[c]) \wedge (C[c-1] \prec B[b] \preccurlyeq C[c]))$$



- Defined outside of the OMG Specification
  - INRIA Research Report RR-6540, RR-6545
  - Is there a place for that kind of information ?

CCSL / Signal / Time Petri Net

- Signal and its Eclipse-based implementation (SME)

```
process strictlySampledOn =
    (? event inp, clk ! event outp )
(| c ^= zc ^= inp ^+ clk
 | zc := c$ init 0
 | c := 1 when clk when inp
          default   0 when clk
          default zc+1 when inp
 | outp := when zc/=0 when clk
 |) where integer c, zc end;
```

- Time Petri net

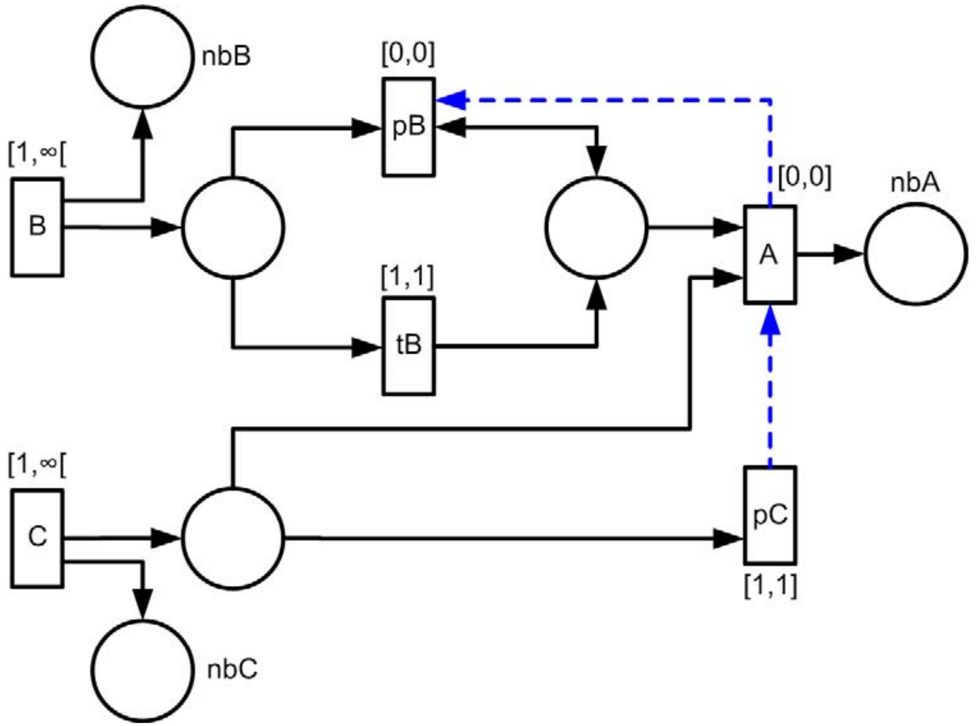
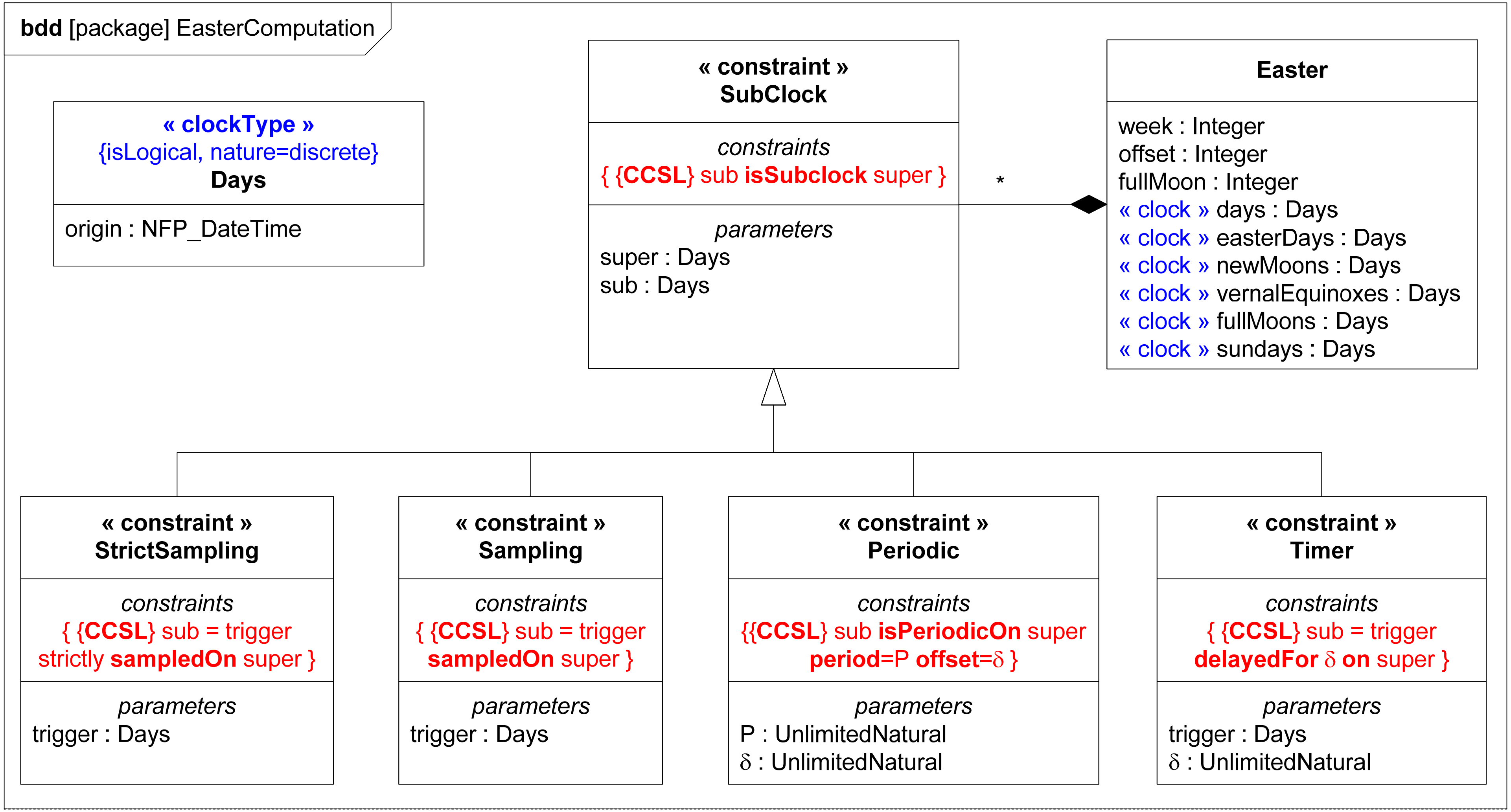


Illustration: Easter

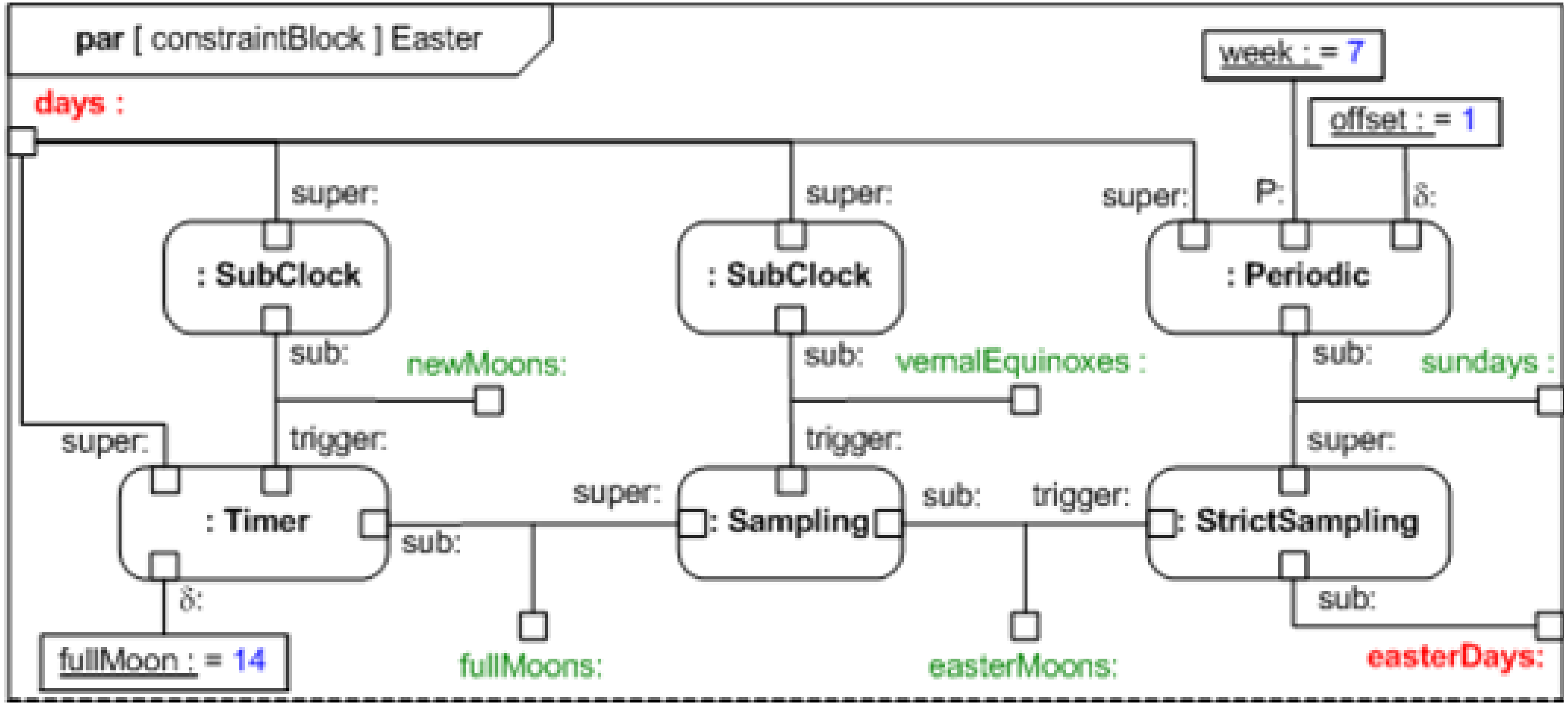
“**Easter Day** *is* the first **Sunday** *after* the 14th day of the **lunar month** that falls *on or after* March 21st (nominally the day of the **vernal equinox**)”

- There are (logical) clocks hidden in the specification: sequence of days (**Days**)
- There are also clock constraints:
  - is (=), after (>), on or after(>=)

Library of CCSL constraint blocks

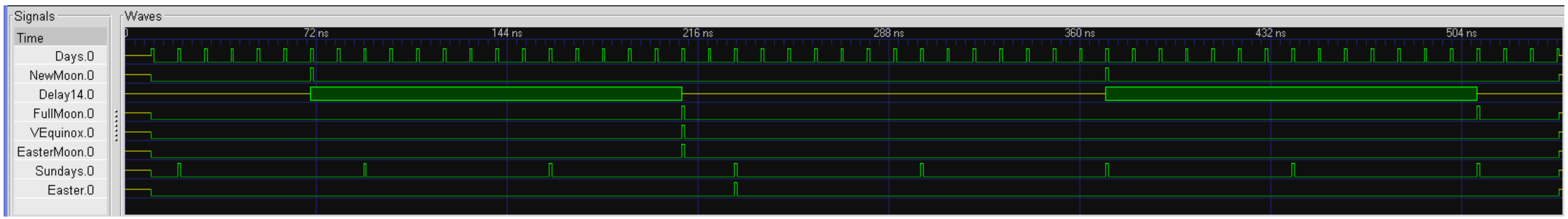


Parametric Diagram with CCSL clocks

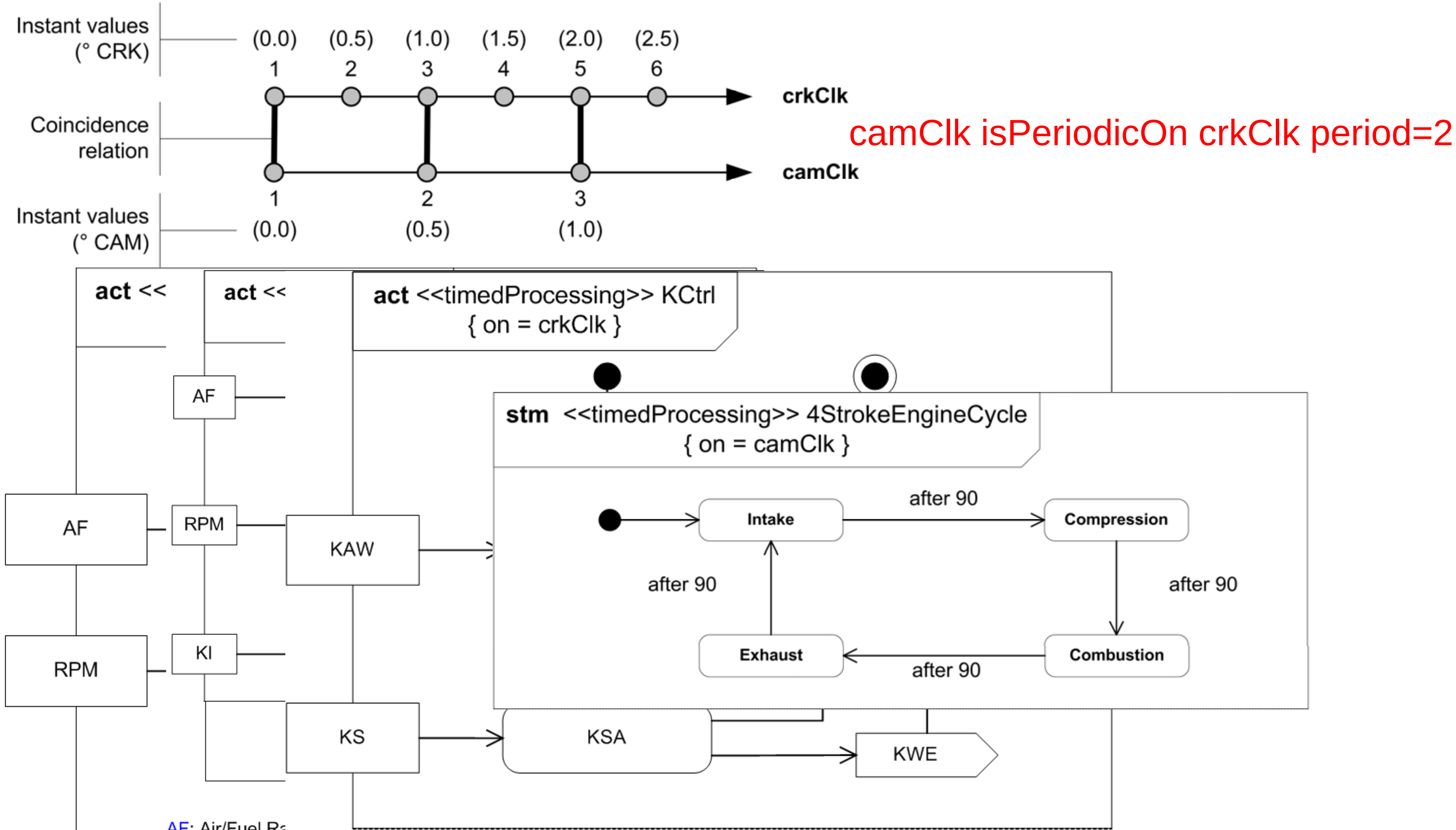


Analysis Tool

- Give one acceptable solution conform to the set of constraints
  - Result in GTKWave
  - **V**alue **C**hange **D**ump (VCD) format, part of the IEEE specification for Verilog-HDL
  - Could be adapted to UML Timing Diagrams



Automotive example: knock

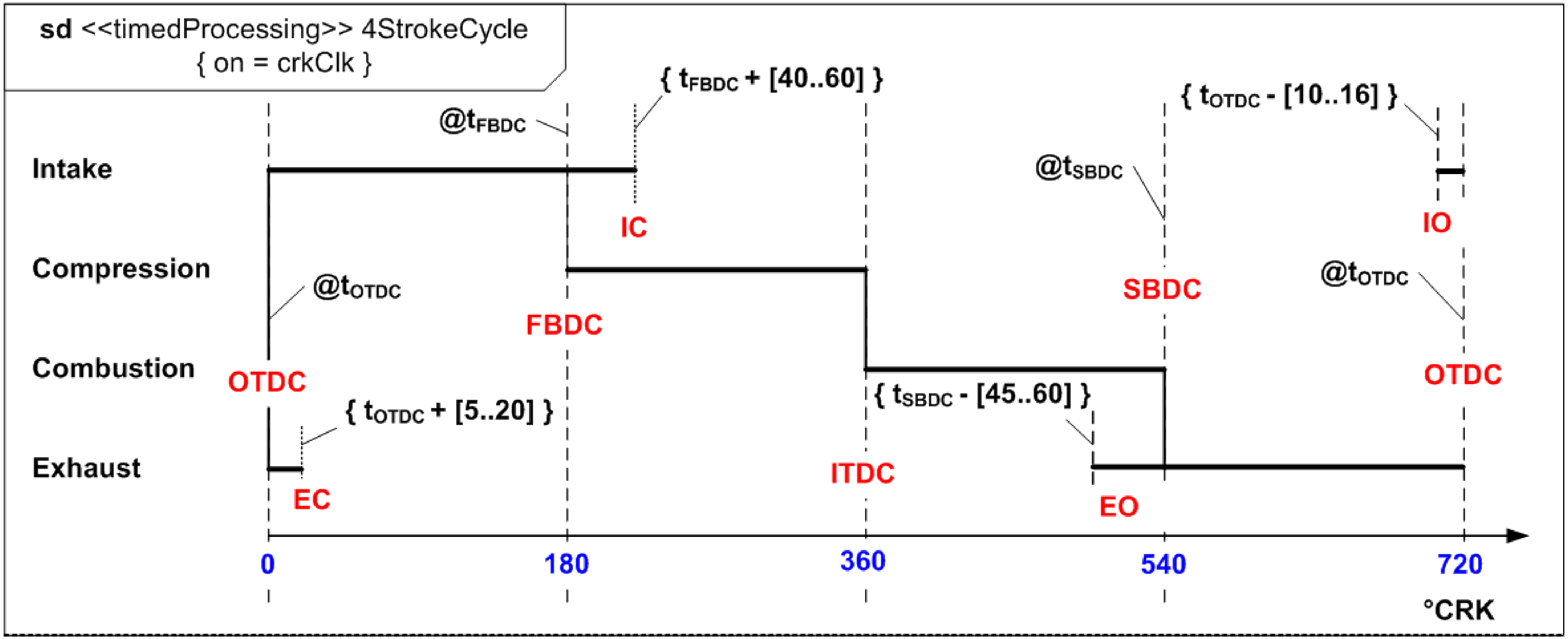


AF: Air/Fuel Ra
KI: Knock Inter
crkClk: cranksh
KS: Knoc BIAA: Basic Igi

KAW: Knock Acquisition Window
KWE: Knock Window End

KWB: Knock Window Begin
KSA: Knock Signal Acquisition

UML Timing Diagrams for results



crkClk: crankshaft Clock
°CRK: degree crank

TDC: Top Dead Center
ITDC: Ignition TDC
OTDC: Overlap TDC

BDC: Bottom Dead Center
FBDC: First BDC
SBDC: Second BDC

IC: Intake closes
IO: Intake opens
EC: Exhaust closes
EO: Exhaust opens