*DiscoRail 2019*
Lyngby, 17 June 2019

# **Modelling a moving block train control system:** different techniques and tools

**Franco Mazzanti** ISTI CNR Pisa Italy

ASTRail project

*Analysis on the state of art of formal methods for the railway sector.*

*Experimentations with 14 different frameworks.*

*Moving block based (toy) case study experimented with 5 different approaches*

*Work still in progress*

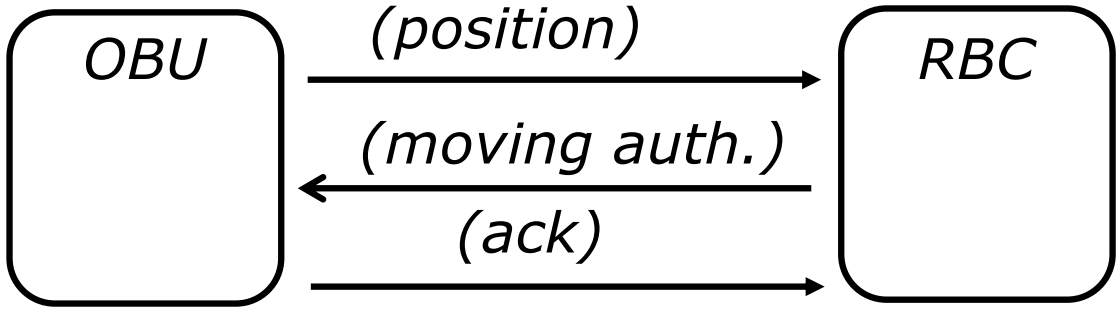
**UML based design** *(UMC)*

Timed/Probabilistic Processes *(Uppaal)*

**Event B** *(ProB)*

**Process Algebras** *(CADP)*

Statecharts *(Simulink)*




- *OBU and RBC start a new cycle every 500 ms*
- *OBU sends position not sooner than 5 secs after the last one*
- *The train position sent by OBU is not older than 1 secs*
- *OBU stops the train if no MA received within 10 seconds*
- *RBC replies to train positions with a MA*
- *if no ack is received, MA is resent three times*
- *…*

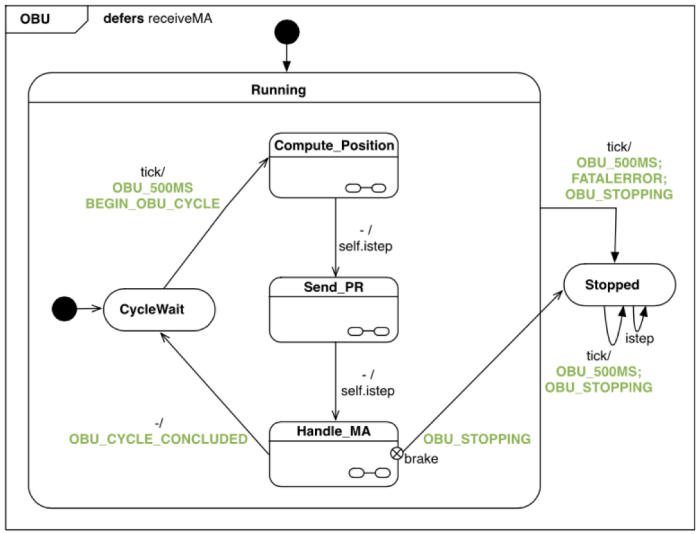
# *Formal Verification of Systems Compositions*

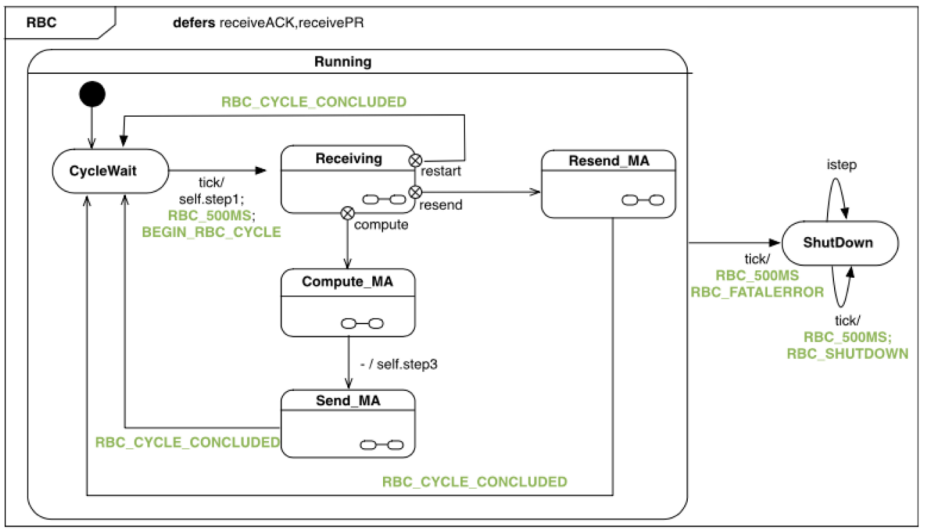
OM7: *It can happen that two MA are recived in the same OBU cycle.*

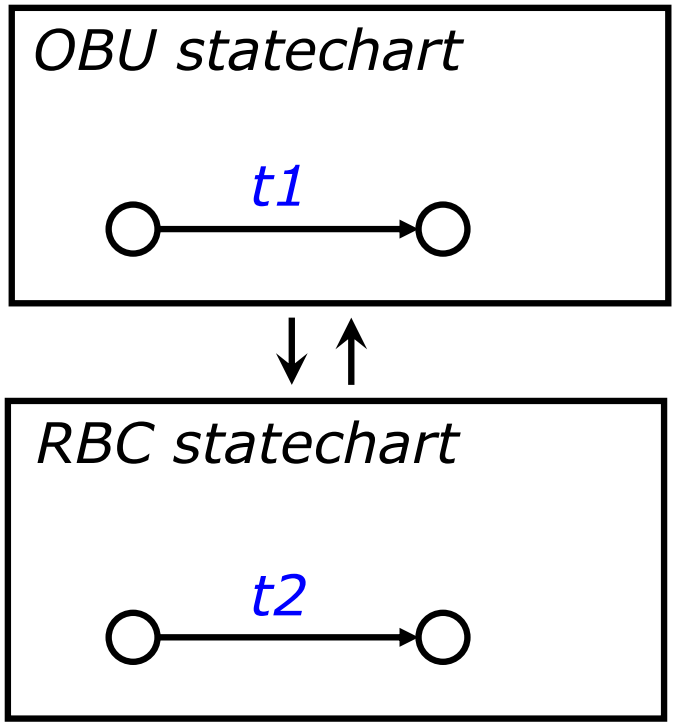
EF <RECEIVE_MA> E [ {**not** BEGIN_OBU_CYCLE } U { RECEIVE_MA } ]

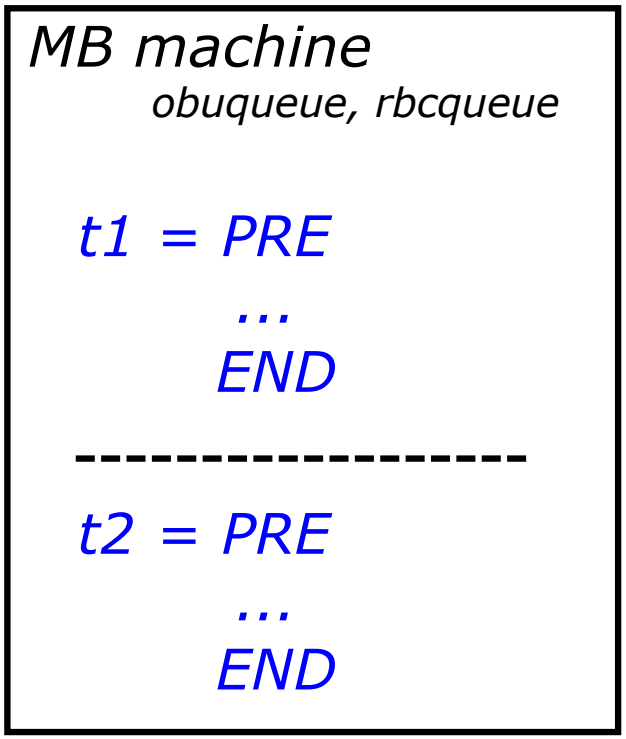
OM8: *It canot happen that three MA are received in the same OBU cycle.*

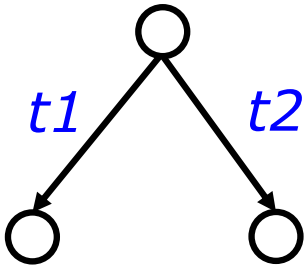
**not** EF <RECEIVE_MA>
E [ {**not** BEGIN_OBU_CYCLE } U { RECEIVE_MA }
E [ {**not** BEGIN_OBU_CYCLE } U { RECEIVE_MA } ]]



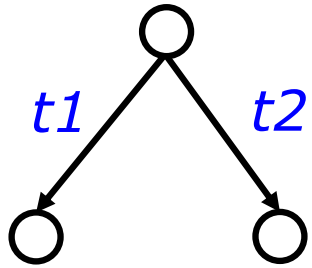






642_863 states




642_865 states

*MB machine*

*obuqueue, rbcqueue*

*t1 = PRE*

*...*

*END*

---

*t2 = PRE*

*...*

*END*

## *Difficulties*

- All components merged in a single machine
- Handling of different UML transition priorities (nesting, completion transitions)
- Handling UML «Deferred» events
- Handling UML «FIFO?» event queues
- Handling UML concurrent regions

## *Solutions*

- *Smart PREconditions & Statements*
- *Simplification of UML features*

## UMC

```
CHECKPR -> CHECKMA
  { istep

   [ LastPR /= null and
     PR_age <= PR_maxage and
     PR_delay >= PR_limit ]
   /
     LABEL.o5_sendpr;
     PR_delay :=0;
     RBC.msgPR;
     LastPR :=null;
     self.istep
  }
```

## ProB

```
o5_sendpr =
PRE
  OBUSTATUS = CHECKPR &
  (#(i).( (i: 0..size(obubuff)) &        // exists i
          (obubuff(i)=istep) &
          !(j).( ( (j: 0..size(obubuff)) &  // forall j
                   (j < i) ) =>
                  (obubuff(j) /= tick )  )
  )     ) &
  LastPR /= null &
  PR_age <= PR_maxage &
  PR_delay >= PR_limit

THEN
  //  skip all deferred items before istep
  VAR  tmp,n,s,done,item  IN
    ...
    obubuff := tmp
  END;
  PR_delay :=0;
  rbcbuff := rbcbuff <- LastPR;
  LastPR := null;
  obubuff := obubuff <- istep;
  OBUSTATUS := CHECKMA
END;
```

# why PROB ?



## UMC

*Good for fast prototyping*

*Good for debugging early designs*

*(Almost) Good for graphical visualization*

*Powerful state/event based branching time logics*

*…*

## ProB

*Good for advanced static analysis*

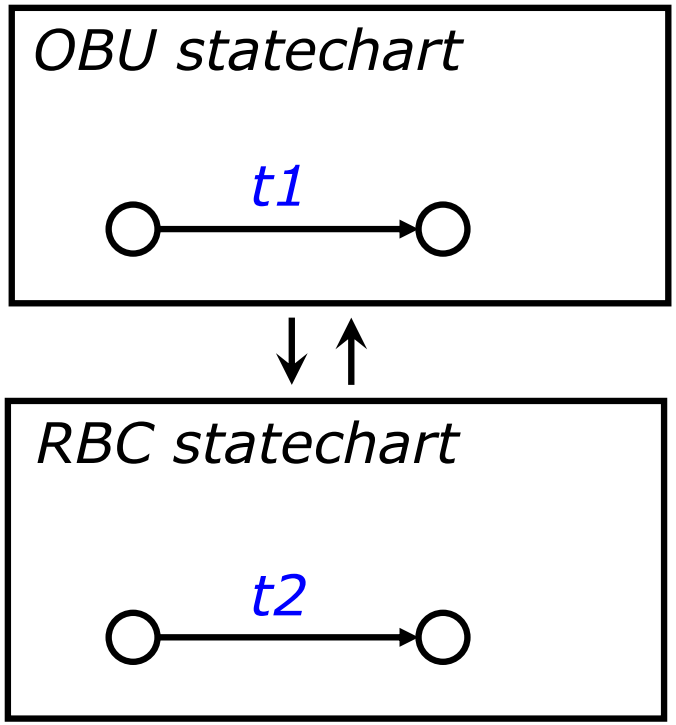
*(Almost) Good for LTL verification*

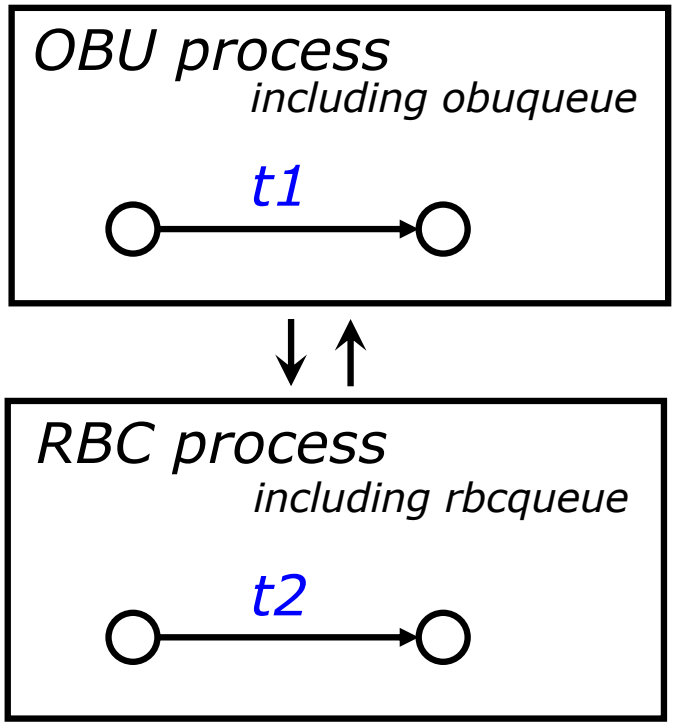
*Powerful state invariants*

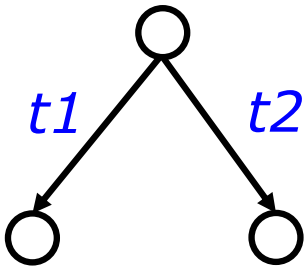
*Allowing controlled refinements into code*

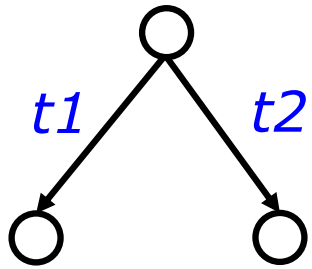
*Good for SMT constraint checking*

*…*

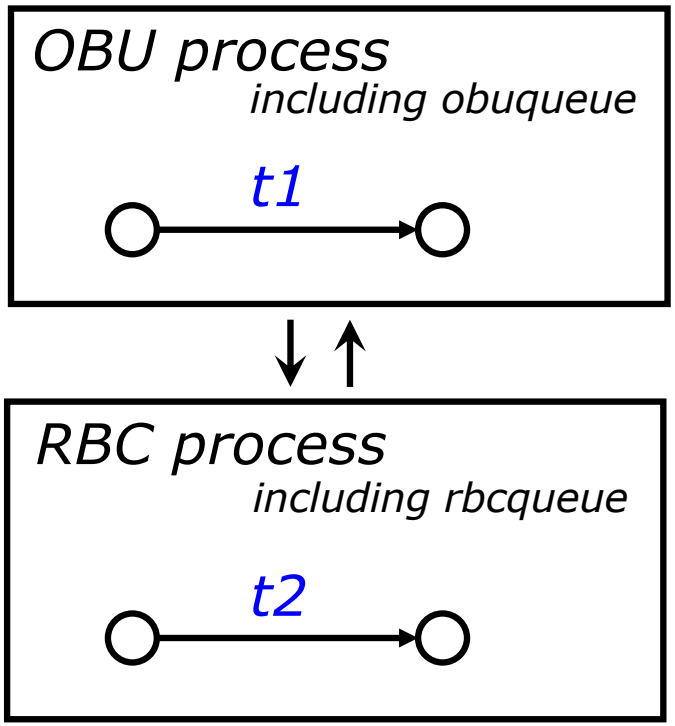



*642_863 states*




## *Difficulties*

- Handling of different UML transition priorities (nesting, completion transitions)
- Handling UML «Deferred» events
- Handling UML «FIFO?» event queues
- Handling UML concurrent regions

## *Solutions*

- *Smart Guards & Statements*
- *Simplification of UML features*

## UMC

```
CHECKPR -> CHECKMA
  { istep

   [ LastPR /= null and
     PR_age <= PR_maxage and
     PR_delay >= PR_limit ]
   /
     LABEL.o5_sendpr;
     RBC.msgPR;
     PR_delay :=0;
     LastPR :=null;
     self.istep
  }
```

## LNT

```
loop CHECKPR in
  select
     <external_sync>;
     obuqueue := append(<eventmsg>,obuqueue)
  []
      …    -- for all incoming external signals
  []
     only if
      istep_ready(obuqueue) and
      (LastPR != UNDEF) and
      (PR_age <= PR_maxage) and
      (PR_delay >= PR_limit) then

      obuqueue := delete(istepmsg, obuqueue);
      o5_sendpr;     -- sync action with RBC
      PR_delay :=0;
      LastPR := UNDEF;
      obuqueue := append (istepmsg, obuqueue);
      break CHECKPR
    end if
  []
    ...
  end select
  end loop;
```

*UMC*

*Good for fast prototyping*

*Good for debugging early designs*

*(Almost) Good for graphical visualization*

*Powerful state/event based branching time logics*

*…*

*LNT*

*Nice Imperative style syntax*

*Good compositional verification*

*Good for advanced static analysis*

*Powerful event based branching time logics*

*Supported by rich toolset*

*…*

## Official Formal Disclaimer:


This work has received funding from the S2RJU under the European Union's Horizon 2020 research and innovation programme under grant agreement No 777561.

The opinions and results discussed in this presentation reflect only the author's view and the Shift2Rail Joint Undertaking is not responsible for any use that may be made of the presented information.

fmt.isti.cnr.it

# ASTRail

SAtellite-based Signalling and Automation SysTems on Railways along with Formal Method and Moving Block validation

## THANK YOU!


*CONTACTS*

Franco Mazzanti
Senior Researcher
ISTI CNR Via Moruzzi 1, Pisa , Italy
http://fmt.isti.cnr.it/~mazzanti



*This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 777561*

*Call identifier: H2020-S2RJU-2017*
*Topic: S2R-OC-IP2-01-2017 – Operational conditions of the signalling and automation systems; signalling system hazard analysis and GNSS SIS characterization along with Formal Method application in railway field*