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

Franco Mazzanti
ISTI CNR Pisa Italy
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
the context

Timed/Probabilistic Processes \((Uppaal)\)

Event B \((ProB)\)

Process Algebras \((CADP)\)

Statecharts \((Simulink)\)

UML based design \((UMC)\)
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 received in the same OBU cycle.

\[ \text{EF } \langle \text{RECEIVE\_MA} \rangle \; \text{E} \; \{ \text{not BEGIN\_OBU\_CYCLE} \} \; \cup \; \{ \text{RECEIVE\_MA} \} \]

OM8: It cannot happen that three MA are received in the same OBU cycle.

\[ \text{not EF } \langle \text{RECEIVE\_MA} \rangle \]
\[ \text{E} \; \{ \text{not BEGIN\_OBU\_CYCLE} \} \; \cup \; \{ \text{RECEIVE\_MA} \} \]
\[ \text{E} \; \{ \text{not BEGIN\_OBU\_CYCLE} \} \; \cup \; \{ \text{RECEIVE\_MA} \} \; \cup \; \{ \text{RECEIVE\_MA} \} \]
OBU statechart

\[ t1 \]

RBC statechart

\[ t2 \]

MB machine

\[ \text{obuqueue, rbcqueue} \]

\[ t1 = \text{PRE} \]
\[ \ldots \]
\[ \text{END} \]

\[ t2 = \text{PRE} \]
\[ \ldots \]
\[ \text{END} \]

642_863 states

642_865 states
DiscoRail 2019

**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 PRE**conditions & Statements
- Simplification of UML features
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
}

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

...

DiscoRail 2019
642_863 states
DiscoRail 2019  

**OBU process**  
including obuqueue  

**RBC process**  
including rbcqueue  

**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*
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;
<table>
<thead>
<tr>
<th><strong>UMC</strong></th>
<th><strong>LNT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Good for fast prototyping</td>
<td>Nice Imperative style syntax</td>
</tr>
<tr>
<td>Good for debugging early designs</td>
<td>Good compositional verification</td>
</tr>
<tr>
<td>(Almost) Good for graphical visualization</td>
<td>Good for advanced static analysis</td>
</tr>
<tr>
<td>Powerful state/event based branching time logics</td>
<td>Powerful event based branching time logics</td>
</tr>
<tr>
<td></td>
<td>Supported by rich toolset</td>
</tr>
</tbody>
</table>
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.
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