Design and Modeling of Level Crossing Control System with Formal Methods

(Authors：Xia Wang, Keming Wang, Peng Cheng, Ning liu)

Reporter | Xia Wang

Southwest Jiaotong University

2019.08.01 on KMOTS 2019 in Guilin

Outline

- Motivation
- The new level crossing control system (NLCCS)
- Introduction to event-B
- Modelling and refinements
- Verification
- Conclusion

（1） Motivation

The level crossing

Accidents at European LCs account for about one-third of the entire railway accidents and result in more than 300 deaths every year in Europe. It accounts for 24% in all the railway accidents, contributing to injuries of 28.7% and casualties of 30.4% in 2017.

Why level crossing (LC)?

- The LC is not clear when a train is approaching the intersection block.
- The opening time is not enough to pass through.
- The equipment is work abnormally.
- ……

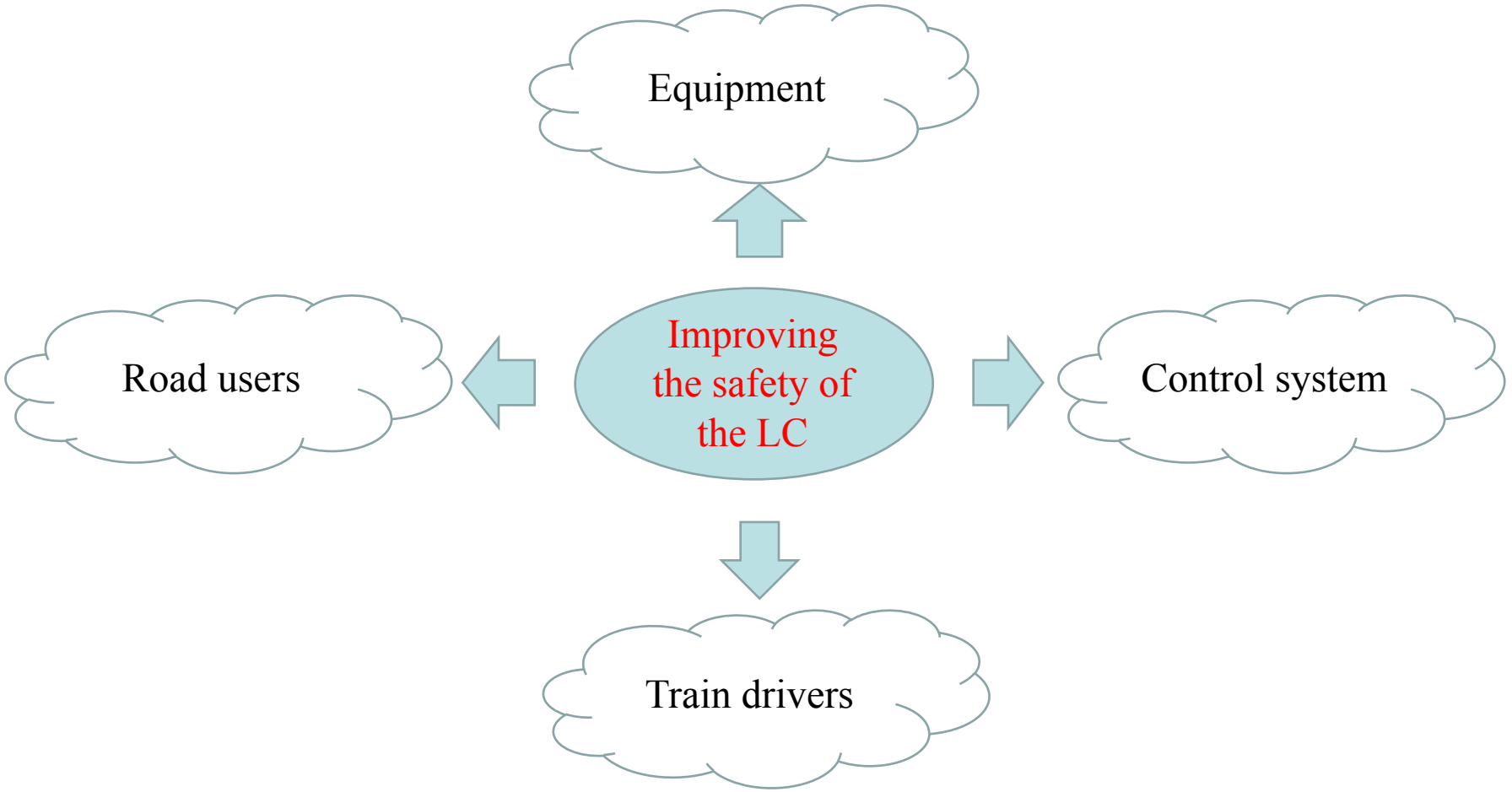
Advantages:

The LC control system is a good example which enough to demonstrate the characteristics of a control system.

Research Review

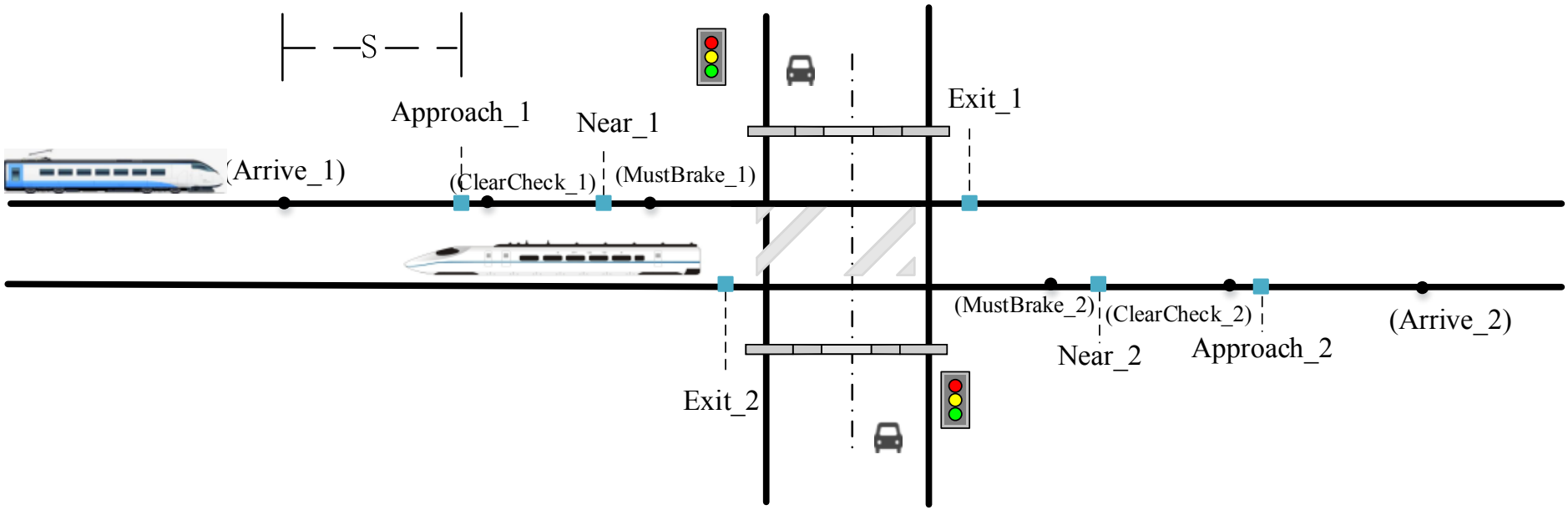
- 2009 Towards safer level crossings Existing recommendations, new applicable technoligies a...
- 2009 Using context descriptions and property definition patterns for software formal verific...
- 2009 Using stochastic Petri nets for level-crossing collision risk
- 2009 Verification of temporal requirements of complex systems using UML patterns, applica...
- 2010 Critical scenarios and their identification in parallel railroad level crossing traffic contr...
- 2010 PANsafer Project —Towards a safer level crossing
- 2010 Patterns for Temporal Requirements Engineering - A Level Crossing Case Study.
- 2010 time-constrained systems validation using mda model transformation ...
- 2012 Adding Technological Solutions for Safety Improvement at Level Crossings A Function...
- 2012 Validation of a new functional design of automatic protection systems at level crossing...
- 2013 Decision support model for prioritizing railway level crossings for safety improvement...
- 2014 Two-Half-Barrier Level Crossings Versus Four-Half-Barrier Level Crossings— A Compar...
- 2015 A Video-Analysis-Based Railway–Road Safety System for Detecting Hazard Situations ...
- 2016 Implementation of ERTMS—A Methodology Based on Formal Methods and Simulatio...
- 2016 Model-Based Diagnosis of Multi-Track Level Crossing Plants
- 2017 A new insight on the risky behavior of motorists at railway level crossings— An observ...
- 2017 Bayesian Network Modeling Applied on Railway Level Crossing Safety
- 2018 Developing accident prediction model for railway level crossings

Review analysis



（2）The NLCCS

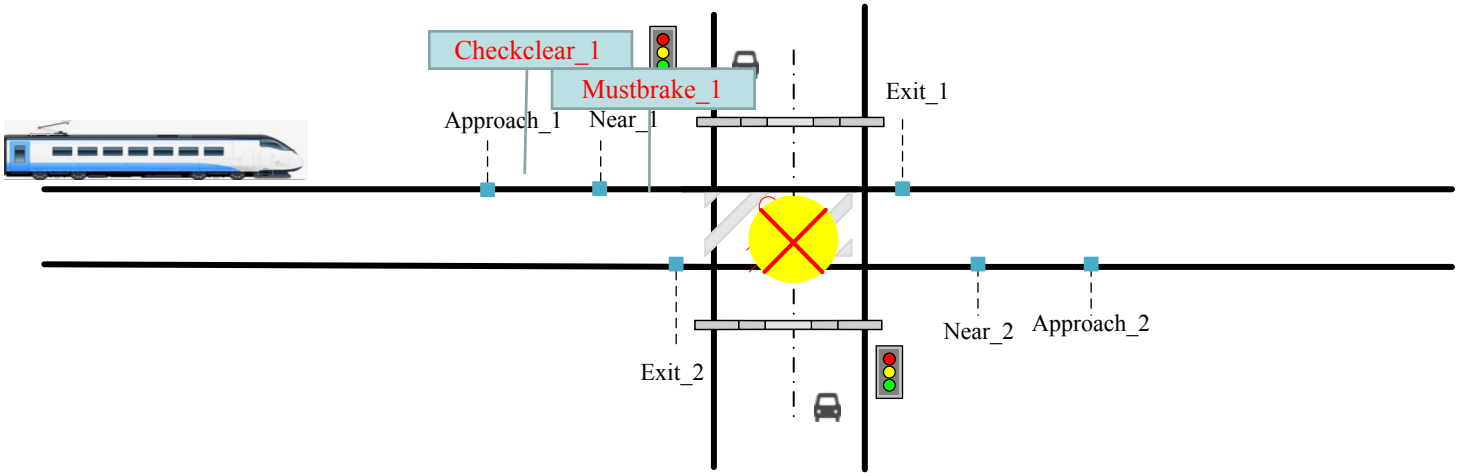
The new LC control system



- **Clearance**

When a train activates the sensor of the approaching site, workers should clear road users.

The train receives the message when arrives at the point of check clearance.
If it is normal, the train can keep going on the railway and pass through the LC.
If abnormal happens, the train should brake. After the abnormal situation is cleared, the train can resume running.



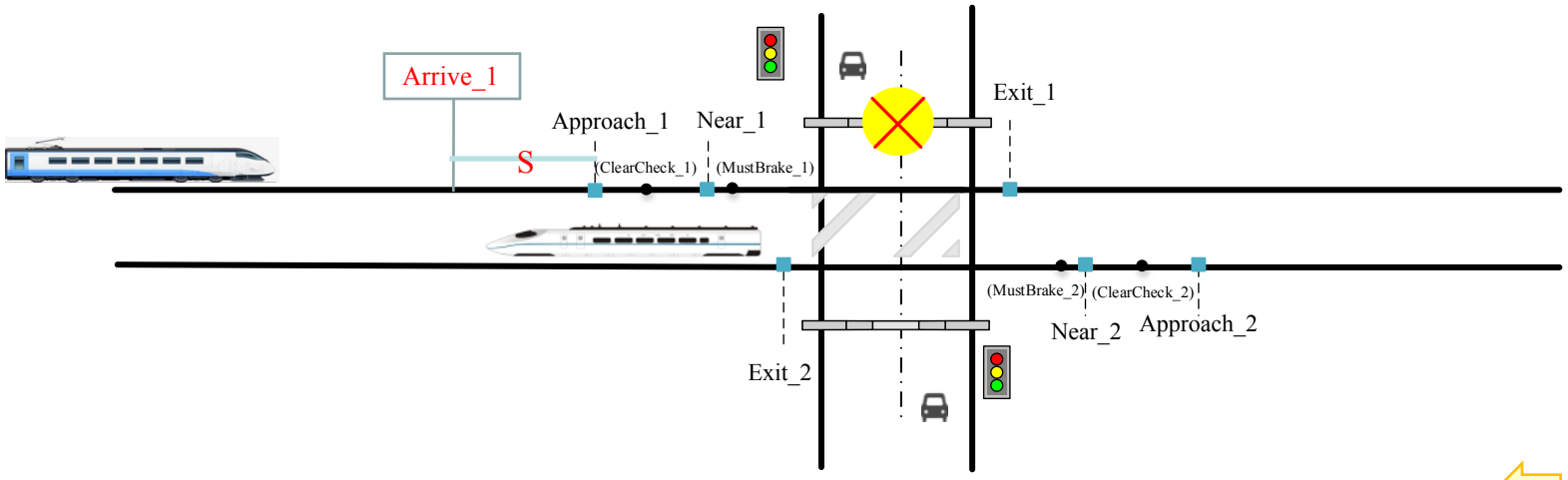
- **Short opened duration (SOD)**

It is important to ensure that the time of gate opened is enough for pedestrians and vehicles to pass through.

The arriving point is set in front of the approach sensor, which is used to make the decision.

$$S = t \cdot v,$$

- t is the minimum time of gate should keep opened
- v is the train speed which is assumed a fixed value
- S is the distance between the arriving point and the approach sensor



（3）Introduction to event-B

Formal methods have a precise mathematical logic, which help developers finding potential errors in the safety-critical system, and the system can be modified in time.

Event-B is a formal method for system-level modelling and analysis, which derived from the B method to model reactive systems.

Features of the event-B

- the use of set theory as a modelling notation
- the use of refinement to represent systems at different abstraction levels
- the use of mathematical proof to verify consistency between refinement levels.
- ……

Event-B model is defined by a tuple (C, S, A, υ, I, Σ, E, Init),
where

- C and S are the model constants and sets (types) respectively.
- A (c, s) is a collection of model axioms.
- υ is the set of system variables.
- I (c, s, υ) is the model invariant limiting the possible states of υ,
- I is a set of invariant properties over υ, c, and s.
- Σ is a model state space defined by all possible values of the vector υ.
- E is a model event set. Init is a predicate defining a non-empty set of model initial states.
- Init is a predicate defining a non-empty set of model initial states.

An event has the following form,

$$e \Rightarrow \text{any } \alpha \text{ where } G \text{ then } A \text{ end}$$

where,

- e is the events name
- a is a list of parameters
- the guard G is a predicate over the model variables
- the A is actions

Rodin

Context structure

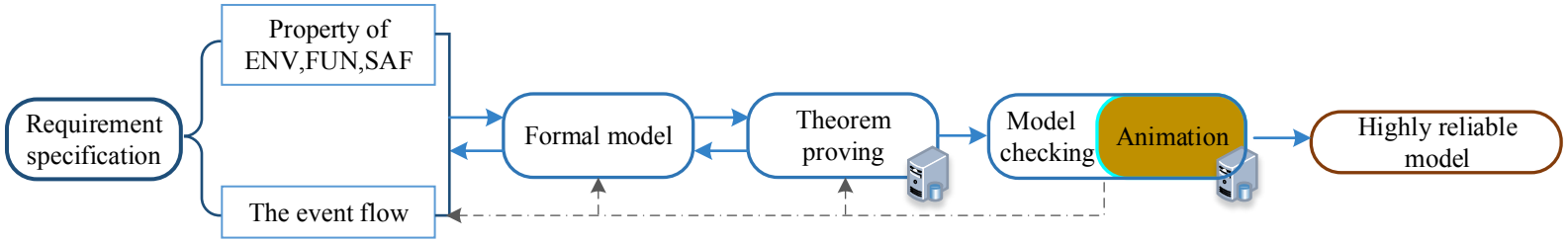
< context_identifier >
extends
< context_identifier_list >
sets
< set_identifier_list >
constants
< constant_identifier_list >
axioms
< label >: < predicate > ...
theorems
< label >: < predicate > …
end

Machine structure

< machine_identifier >
refines
< machine_identifier >
sees
< context_identifier_list >
variables
< variable_identifier_list >
invariants
< label >: < predicate > ...
theorems
< label >: < predicate > ...
variant
< variant > events < event_list >
end

（4）Modelling and refinements

The technical route



EVN	describes the properties of modeling objects in the system
FUN	describes the basic functions of a system
SAF	describes the safe conditions that the control system should have during its operation

Properties analysis

FUN1	When a train activates the approach sensor, the signal is switched to red.
FUN2	When a train activates the near sensor, the signal is switched to red flash.

EVN 1	Gate state includes: opened, opening, closing, closed.
EVN 2	The states of the LC signal lights, red, red flash, white.
EVN 3	There has six alarm sensors of train on the track, AP 1, AP 2, NE 1, NE 2, EX 1, EX 2.
EVN 4	There has six identify points on the railway, AR 1, AR 2, CC 1, CC 2, MB 1, MB 2.
EVN 5	The commands of the controller sends to the gate have three types, goup, keepclosed, go down.

SAF 1	The train should brake when the LC is stay in abnormal situation which caused by unclearing.
SAF 2	The opened gate is satisfy the minimum opened time in the successive closure cycles.

Modelling ---The initial model

Context file

Axiom1: Up_MAXTRAINSinATrack = 1,

Axiom2: Down_MAXTRAINSinATrack = 1,

Axiom3: Up_Maxtrack = 1,

Axiom4: Down_Maxtrack = 1,

Axiom5: Up_TRAINNUM = N,

Axiom6: Down_TRAINNUM = N.

Machine file

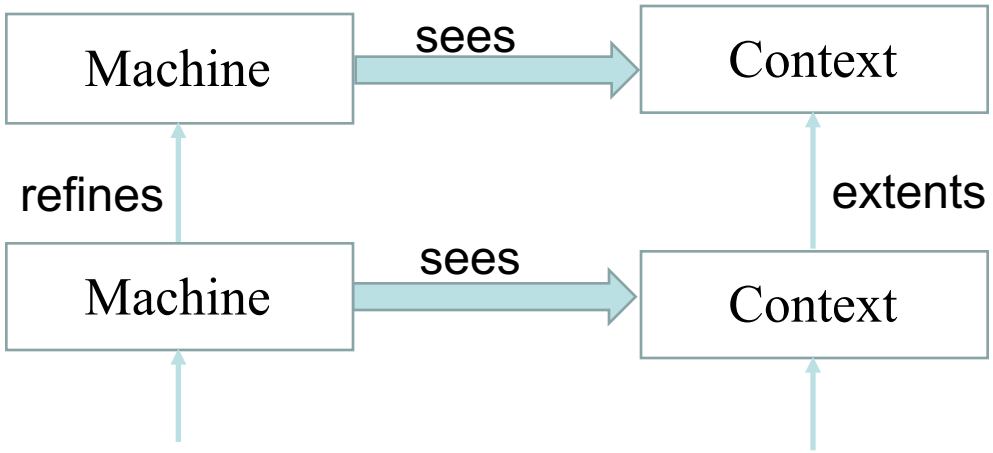
```
Add_Up_track ≙
STATUS
ordinary
ANY
a
WHERE
grd1 : a ∉ Up_track
grd2 : Up_sumtrack < Up_Maxtrack
THEN
act1 : Up_track ≔ Up_track ∪ {a}
act2 : Up_sumtrack ≔ Up_sumtrack +1
act3 : Up_trainintrack (a) ≔ 0
END
```

```
Remove_Up_track ≙
STATUS
ordinary
ANY
a
WHERE
grd1 : a ∈ Up_track
grd2 : Up_sumtrack > 0
grd3 : Up_trainintrack(a) = 0
THEN
act1 : Up_track ≔ Up_track \ {a}
act2 : Up_sumtrack ≔ Up_sumtrack −1
act3 : Up_trainintrack ≔ {a} ⩤ Up_trainintrack
END
```

```
Add_Up_train ≙
STATUS
ordinary
ANY
n
a
WHERE
grd1 : n ∉ Up_train
grd2 : a ∈ Up_track
grd3 : Up_trainintrack (a) <
Up_MAXTRAINSinATrack
grd4 : Up_trainintrack (a) = 0
grd5 : up_train_lc_pos = ∅
THEN
act1 : Up_train ≔ Up_train ∪ {n}
act2 : Up_trainintrack (a) ≔ Up_trainintrack
(a)+1
act3 : up_train_lc_pos(n) ≔ Arrive_1
act4 : up_train_gate_time(n) ≔ TimeofArrive
END
```

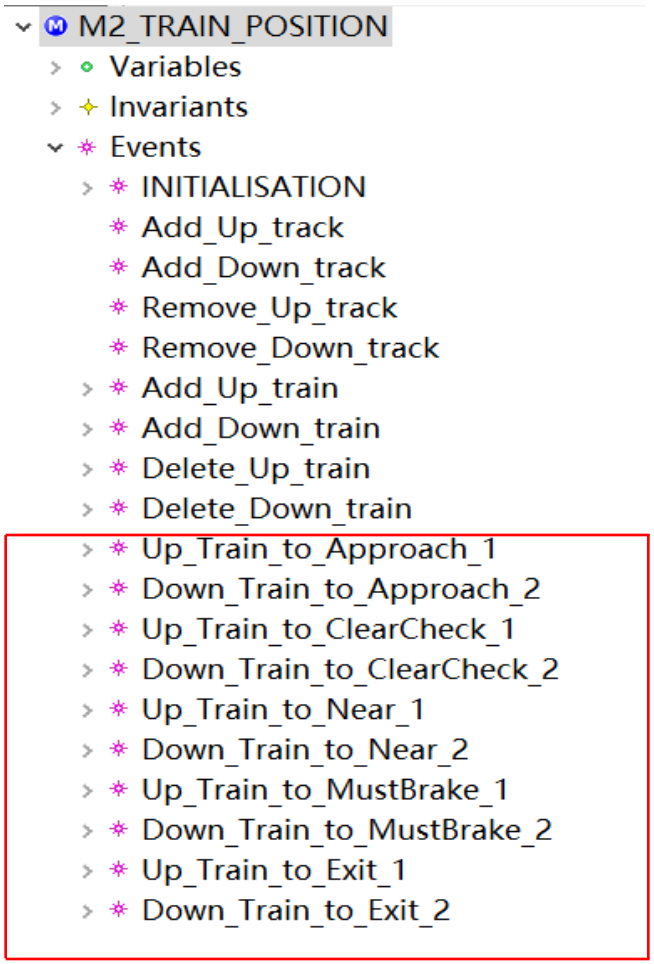
```
Delete_Up_train ≙
STATUS
ordinary
ANY
n
a
WHERE
grd1 : n ∈ Up_train
grd2 : a ∈ Up_track
grd3 : Up_trainintrack (a) > 0
grd4 : up_train_lc_pos (n) = Exit_1
THEN
act1 : Up_train ≔ Up_train \ {n}
act2 : Up_trainintrack(a) ≔ Up_trainintrack(a)
− 1
act3 : up_train_lc_pos ≔ {n} ⩤ up_train_lc_pos
act4 : up_train_gate_time ≔ {n} ⩤
up_train_gate_time
END
```

Refinements



Refinement Steps	Objects and Events
First refinement	Records the behavior of distance update.
Second refinement	Describes the process of check clearance.
Third refinement	Depicts the change of the gate states
Fourth refinement	Appends the change of signal states on the NLCCS.

- **The first refinement**



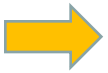
Additional events

- **The second refinement**

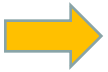
Clearance judgment

SETS	CONSTANTS
Crossing_Check	Unclear
	Clear
	Null
Crossing_State	Normal
	Abnormal
Train_Ctrl	Brake
	KeepGoing
	Normal_Ctrl

Axm1: Partition (Train_Ctrl, {Brake}, {KeepGoing},{Normal_Ctrl})



Axm2: Partition (Crossing_State, {Normal}, {Abnormal})



Axm3: Partition (Crossing_Check, {Unclear}, {Clear}, {Null})

- **The second refinement**

INV1: ((∃ up_n · up_n ∈ Up_train ∧ (up_train_lc_pos (up_n) = MustBrake _1)) ∨ (∃down_m · down_m ∈Down_train ∧(down_train_lc_pos (down _m) = MustBrake_2))) ∧ (Gate_inforClosed∨Crossing_check =Unclear) ∧Sending Infor = Abnormal ⇒ TrainOrder = Brake.

SAF 1	The train should brake when the LC is stay in abnormal situation which caused by unclearing.

clearance

```
SendUnclearMessage:    not extended ordinary ›
WHERE
    grd1:   (∃up_n· up_n ∈ Up_train ∧ (up_train_lc_pos(up_n) = ClearCheck_1))
            ⇒ Crossing_check ≠ Clear
            ∨ (∃down_k· down_k ∈ Down_train ∧ (down_train_lc_pos(down_k) = Near_2))  not theo
    grd2:   (∃down_k· down_k ∈ Down_train ∧ (down_train_lc_pos(down_k) = ClearCheck_2))
            ⇒ Crossing_check ≠ Clear
            ∨ (∃up_m· up_m ∈ Up_train ∧ (up_train_lc_pos(up_m) = Near_1))   not theorem ›
    grd3:   Crossing_check = Clear ⇒
            (∃up_n· up_n ∈ Up_train ∧ (up_train_lc_pos(up_n) = Near_1)) ∨
            (∃down_m· down_m ∈ Down_train ∧ (down_train_lc_pos(down_m) = Near_2))  not theore
    grd4:   (∃up_n· up_n ∈ Up_train ∧ (up_train_lc_pos(up_n) = ClearCheck_1)) ∨
            (∃down_m· down_m ∈ Down_train ∧ (down_train_lc_pos(down_m) = ClearCheck_2))
            ∨
            (∃up_n· up_n ∈ Up_train ∧ (up_train_lc_pos(up_n) = Near_1)) ∨
            (∃down_m· down_m ∈ Down_train ∧ (down_train_lc_pos(down_m) = Near_2))  not theore
    grd5:   (∃up_n· up_n ∈ Up_train ∧ (up_train_lc_pos(up_n) = ClearCheck_1)) ∨
            (∃down_m· down_m ∈ Down_train ∧ (down_train_lc_pos(down_m) = ClearCheck_2))
            ∨
            (∃up_n· up_n ∈ Up_train ∧ (up_train_lc_pos(up_n) = Near_1)) ∨
            (∃down_m· down_m ∈ Down_train ∧ (down_train_lc_pos(down_m) = Near_2))  not theore
THEN
    act1:   Crossing_check ≔ Unclear  ›
END
```

- **The third refinement**

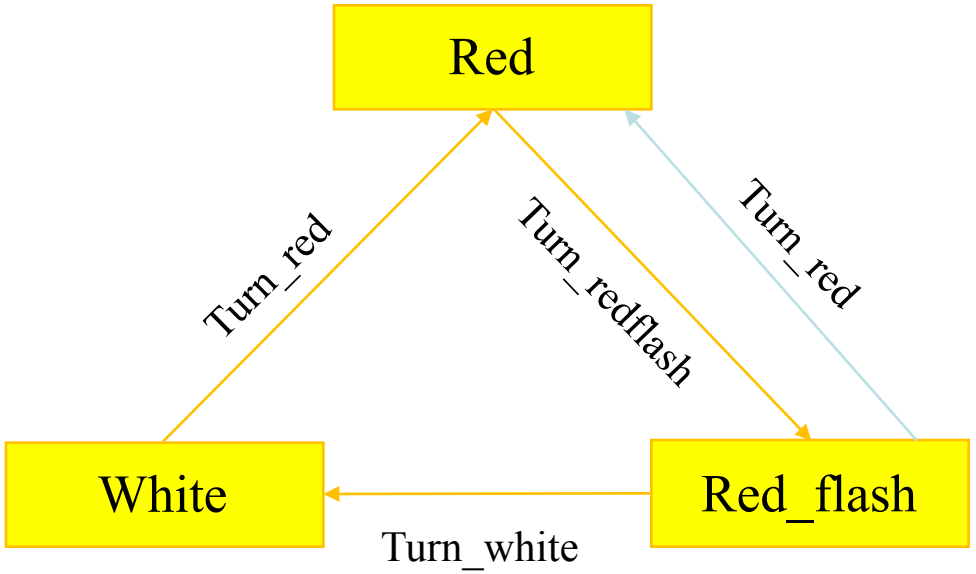
```
inv1:  Gate_infor ∈ Gate_State not theorem ›
inv2:  Gate_ctr ∈ Gate_Order not theorem ›
inv3:  Gate_ctr ∈ Gate_Order not theorem ›
inv4:  T_OpeningStart ∈ N not theorem ›
inv5:  Time ∈ N not theorem ›
inv6:  Gate_infor = Closed ⇒ Time ≥ T_ClosingStart + Time_Closing not theorem
inv7:  Gate_infor = Opened ⇒ Time ≥ T_OpeningStart + Time_Opening not theorem
inv8:  ran(up_train_lc_pos) ⊆ {Arrive_1,Exit_1} ∧ ran(down_train_lc_pos)
       ⊆ {Arrive_2,Exit_2}  ∧ Gate_ctr = GoUp⇒ Gate_infor = Opening  ∨
       Gate_infor = Closed  not theorem ›                          SAF
inv10: ((∃up_n· up_n ∈ Up_train ∧ (up_train_lc_pos(up_n)
       =MustBrake_1)) ∨ (∃down_m· down_m ∈ Down_train ∧
       (down_train_lc_pos(down_m) = MustBr ke_2)))  ∧
       (Gate_infor ≠ Closed ∨ Crossing_che k = Unclear) ∧ Sending_Infor
        = Abnormal ⇒ TrainOrder = Brake   ot theorem ›
inv11: up_train_gate_time ∈ Up_train → DI TANCE not theorem ›
inv12: down_train_gate_time ∈ Down_train →  DISTANCE not theorem ›
inv13: Speedupdate_flag ∈ N not theorem ›
inv14: amountoftrains ∈ N not theorem ›
```

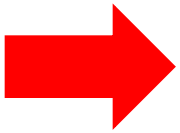
SOD

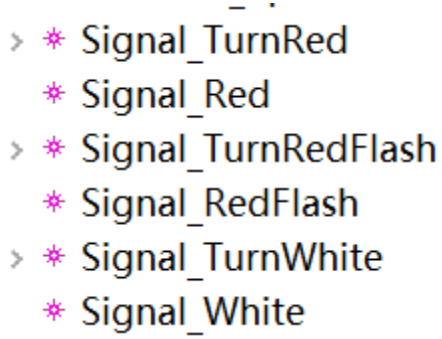
```
Gate_GoUp  ≙
STATUS
 ordinary
WHEN
 grd1   :   Gate_infor = Closed
 grd2   :   ran(up_train_lc_pos) ⊆ {Arrive_1,Exit_1}
 grd3   :   ran(down_train_lc_pos) ⊆ {Arrive_2,Exit_2}
 grd4   :   Sending_Infor ≠ Abnormal
 grd5   :   TrainOrder = Normal_Ctrl
 grd6   :   Crossing_check = Null
THEN
 act1   :   T_OpeningStart ≔ Time
 act2   :   Gate_infor ≔ Opening
 act3   :   Gate_ctr ≔ GoUp
 act4   :   TrainOrder ≔ Normal_Ctrl
 act5   :   Crossing_check ≔ Null
END
```

SAF 2	The opened gate is satisfy the minimum opened time in the successive closure cycles.

- **The fourth refinement**







（5）Verification

Proof obligations are generated from modelling and are input to the proving activity.

Element Name	Total	Auto	Manual	Reviewed	Undischarged
NewProject	**163**	**97**	**66**	**0**	**0**
C1	0	0	0	0	0
C2	0	0	0	0	0
C3	0	0	0	0	0
C4	0	0	0	0	0
C5	0	0	0	0	0
M1_TRAIN	28	25	3	0	0
M2_TRAIN_P...	28	14	14	0	0
M3_clear_che...	11	7	4	0	0
M4_Gate_Stat...	89	50	39	0	0
M5_Signal_St...	7	1	6	0	0

（6）Conclusion

- Analyzing two problems: Clearance + SOD
- Designing a new level crossing control system (NLCCS)
- Modelling the NLCCS
- Verification

Southwest Jiaotong University

THANK YOU

Reporter：Xia Wang