Safety & Capacity of ERTMS - modelled and analyzed in Real-Time Maude

Monika Seisenberger

Joint work with Andrew Lawrence, Ulrich Berger, Phil James, Markus Roggenbach

1st Meeting of the European Technical Working Group on Formal Methods in Railway Control, 28 January 2015

Overview: Safety & Capacity of ERTMS

To investigate
how Real-Time Maude can be used to
verify and simulate the European Rail Traffic Management System

Overview:

- Part I: ERTMS – what it is
- Part II: ERTMS – how it works
- Part II: Generic Modelling: ERTMS as a hybrid automaton
- Part I: Encoding in Real-Time Maude
- Part V: Verification & simulation results

European Rail Traffic Management System (ERTMS) I

What it is:

- European standard of signalling, control and train protection
- to replace the many incompatible safety systems currently used by European railways
- offers possibility for traffic management

What it shall achieve:

- interoperability
- ease of maintenance (less track equipment)
- high capacity

European Rail Traffic Management System (ERTMS) II

Formal methods and ERTMS:

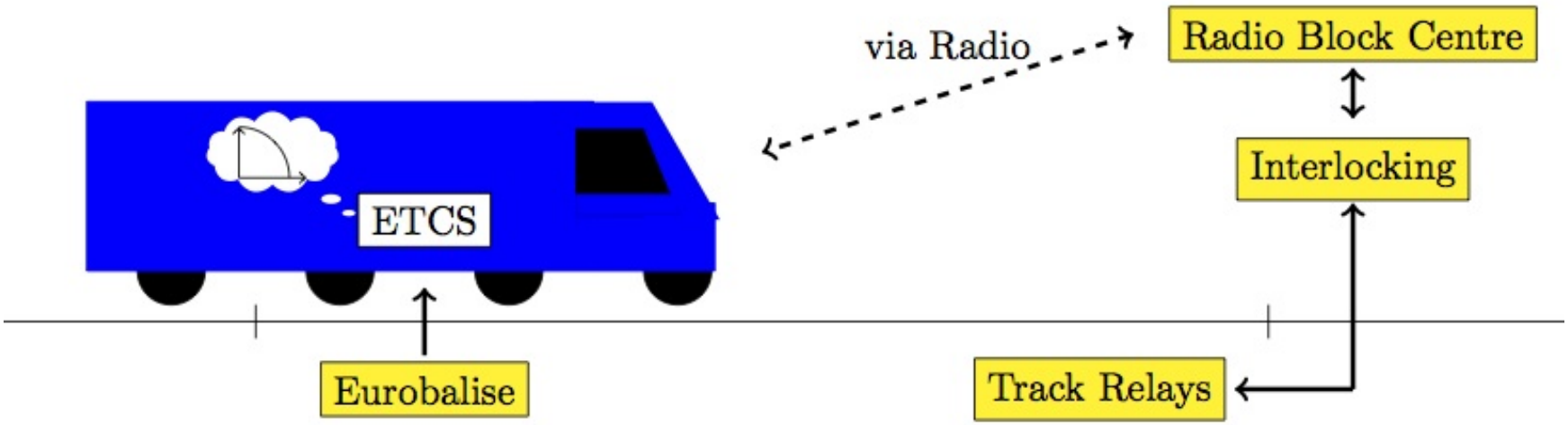
- EuRailCheck (2009): suggested methodology and tools for formalization and validation of the standard.
- Open ETCS Project (ongoing): works towards an integrated framework for modelling, development, implementation and testing of the standard.

Open research questions include:

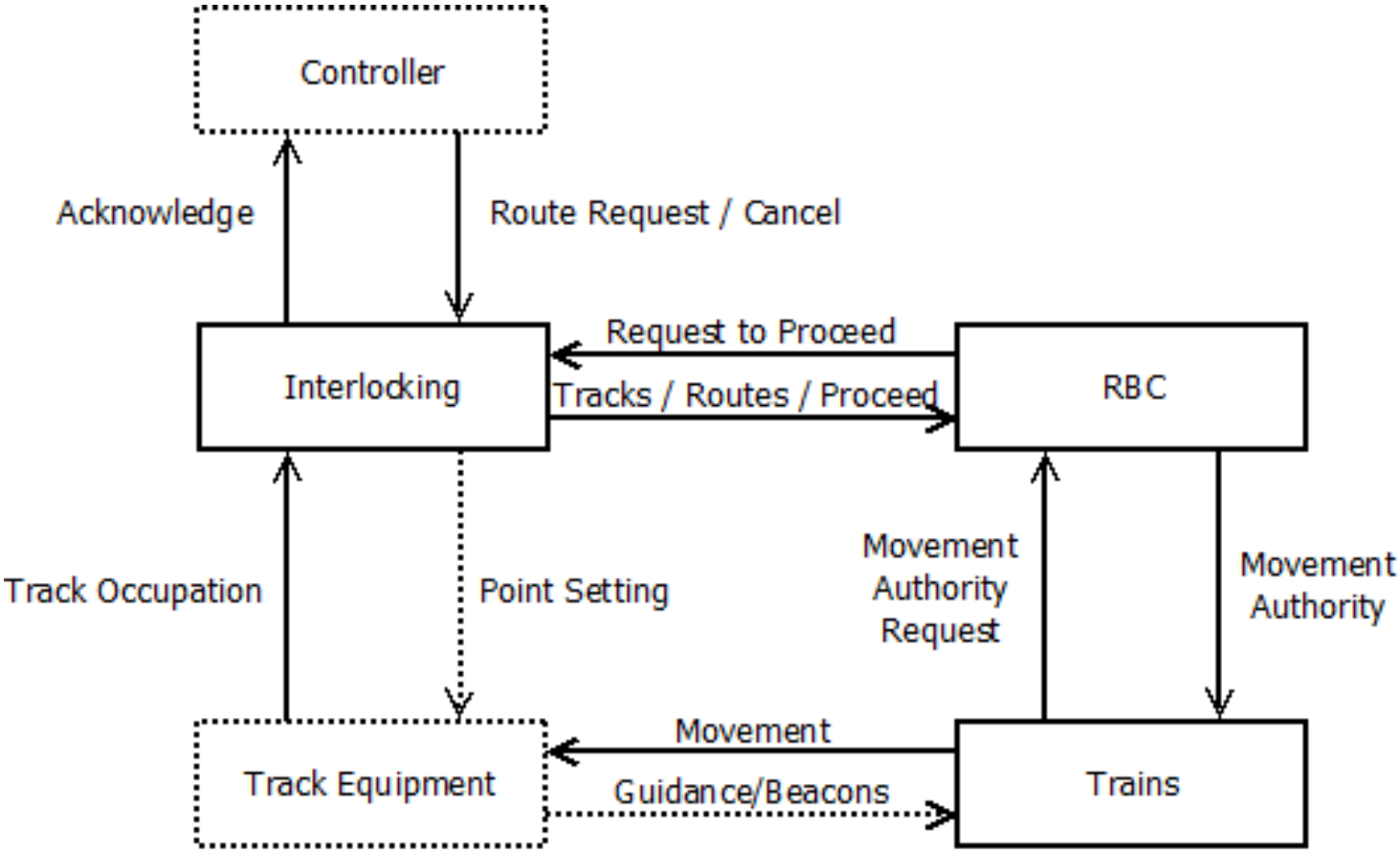
1. How can safety be verified?
2. How can capacity be measured and improved?
3. How can reliability be measured and estimated?

Here: 1 and, partially, 2.

System components of ERTMS, level 2



Information flow in ERTMS, level 2



Data in the ERTMS, level 2

The radio block processor (RBC) deals with continuous data such as the speed and position of the train and grants a movement authority (MA).

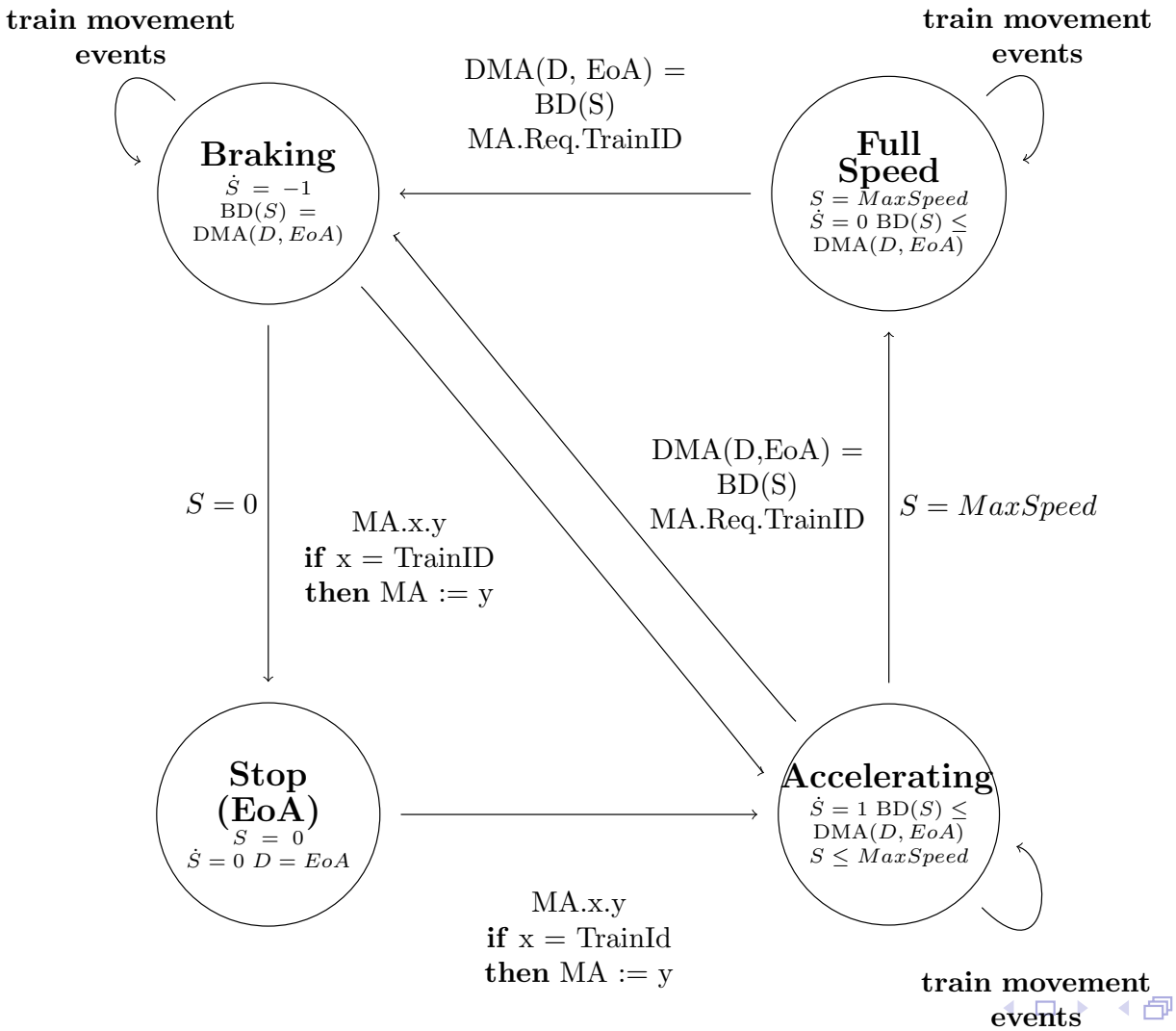
The MA consists of a position on the track past which the train cannot proceed called the end of authority (EoA) and a static speed profile.

Trains move on the track according to the physical laws of movement, their breaking is controlled against a differential equation, their acceleration follows a differential equation.

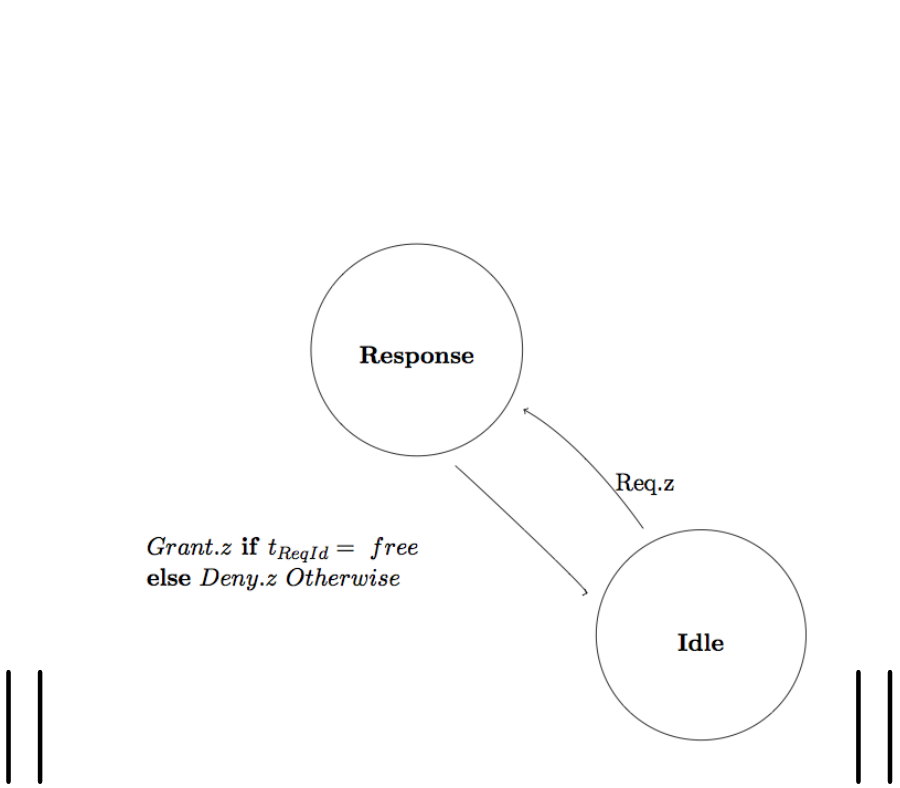
Together the RBC, trains and the interlocking form a hybrid system.

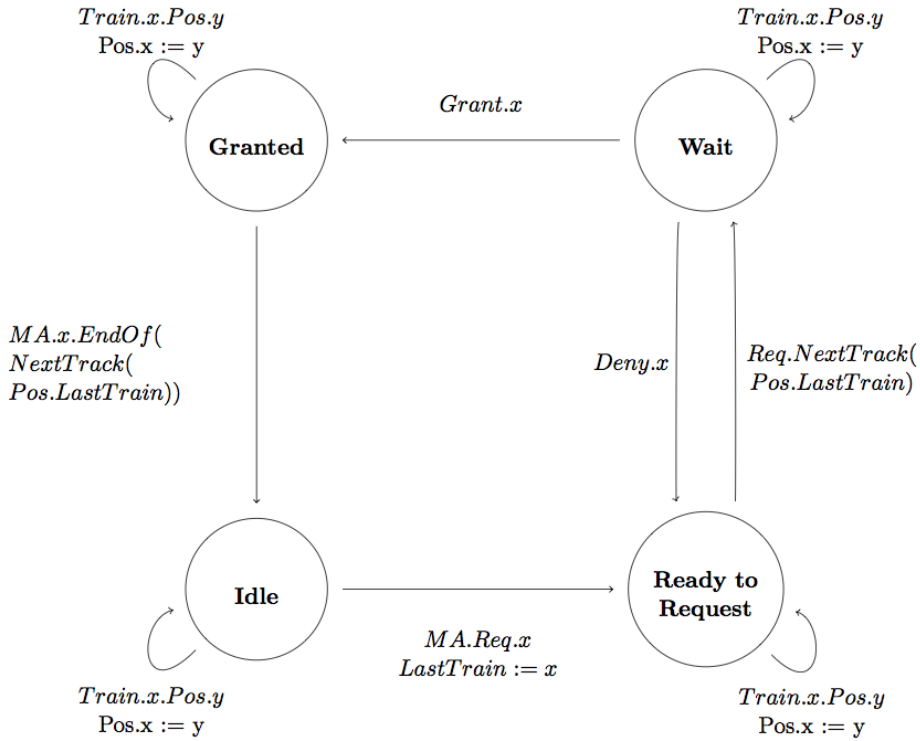
Generic modelling: ERTMS as a hybrid automaton

A model of a train with ERTMS equipment

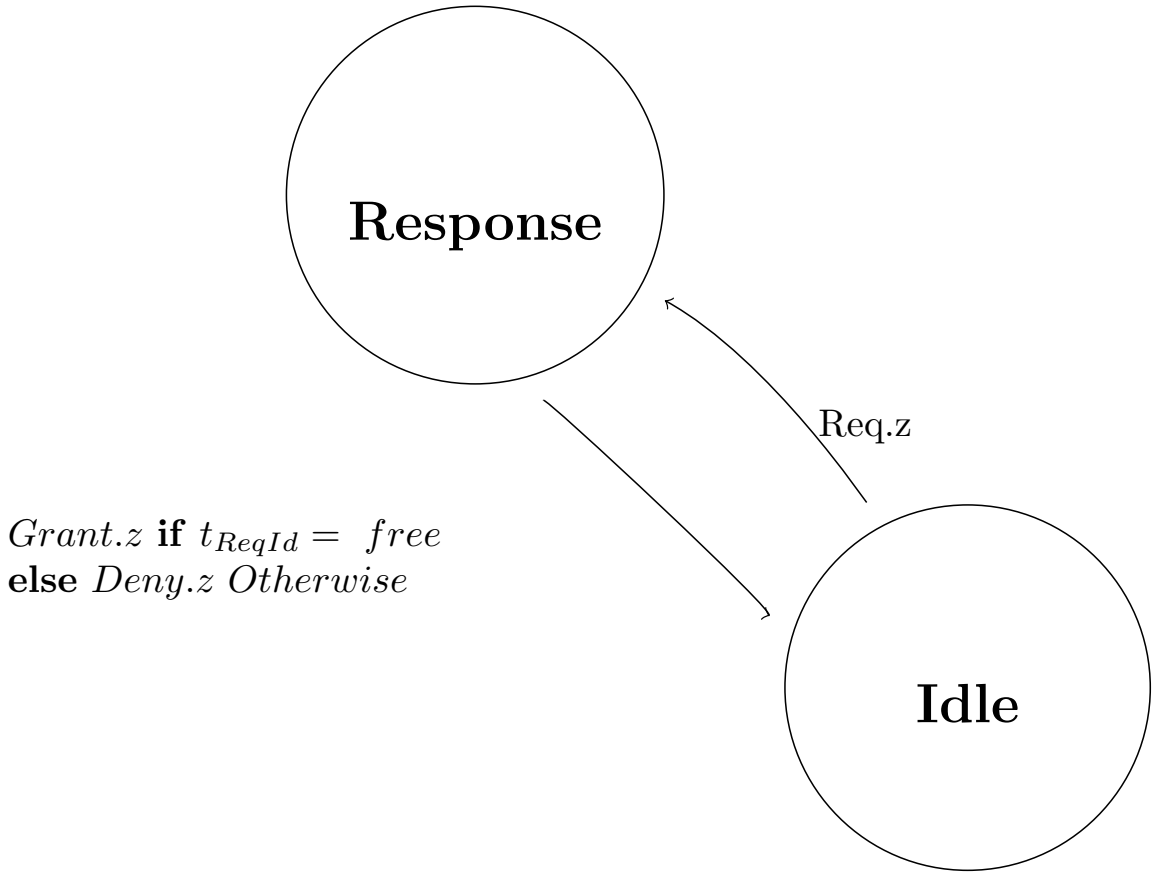


ERTMS as a hybrid automaton

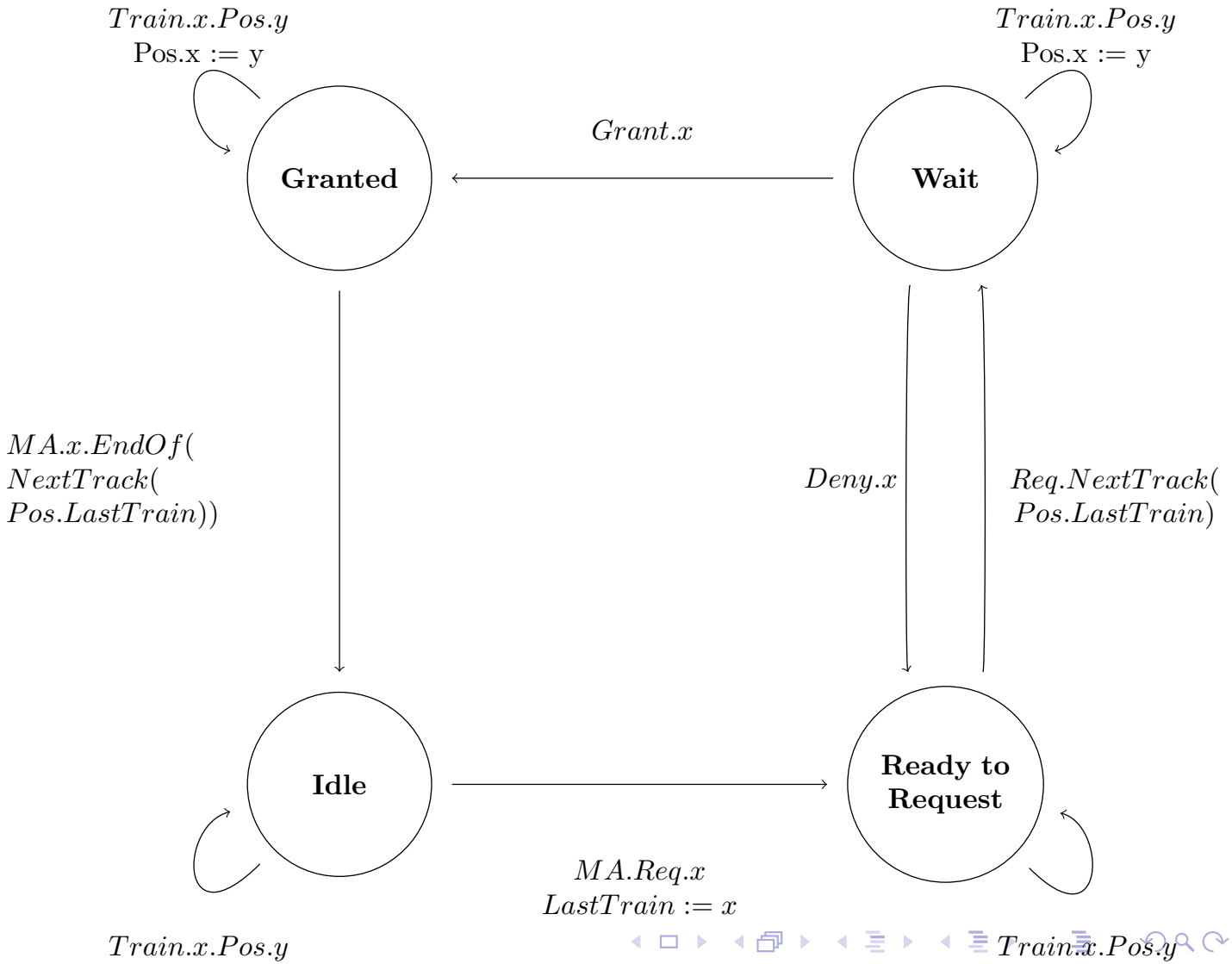




Modelling the Interlocking



Modelling the RBC



Well-formedness of the hybrid automaton

Modelling based on Henzinger, 2000.

Theorem This hybrid automaton is non-Zeno,
i.e. every finite initialised trajectory in the labelled transition system defined by the automaton has an infinite trajectory of which it is a prefix in set of divergent initialized trajectories in the automaton.

Encoding in Real-Time Maude

Real-Time Maude (Peter C. Ölveczky and José Meseguer 2002) is a language and tool extending Maude, that allows for simulation and formal analysis of real-time and hybrid systems.

We make use of especially of its

- OO features
- Discrete time rewriting rules:

```
rl [label] :  {Sys} => {Sys} in time t .
crl [label] :  {Sys} => {Sys} in time t if C .
```

Object Oriented Modelling

```
mod CONFIGURATION is
    *** basic object system sorts
    sorts Object Msg Configuration .

    *** construction of configurations
    subsort Object Msg < Configuration .
    op none : -> Configuration [ctor] .
    op __ : Configuration Configuration -> Configuration
         [ctor config assoc comm id: none] .
```

Timed Progress of a train in state "accelerating"

```
crl [incspeed] :
  {delta(< O : Train | state : acc, dist : DT, speed : S,
                       ac : A, ma : MA, maxspeed : MAX >
         )
   REST
  }
=>
  {pos(O,DT)
   < O : Train |  dist : (DT + S), speed : S + A >
   REST
  }
if S + A < MAX and ... .
```

Message consumption for the RBC in state "wait"

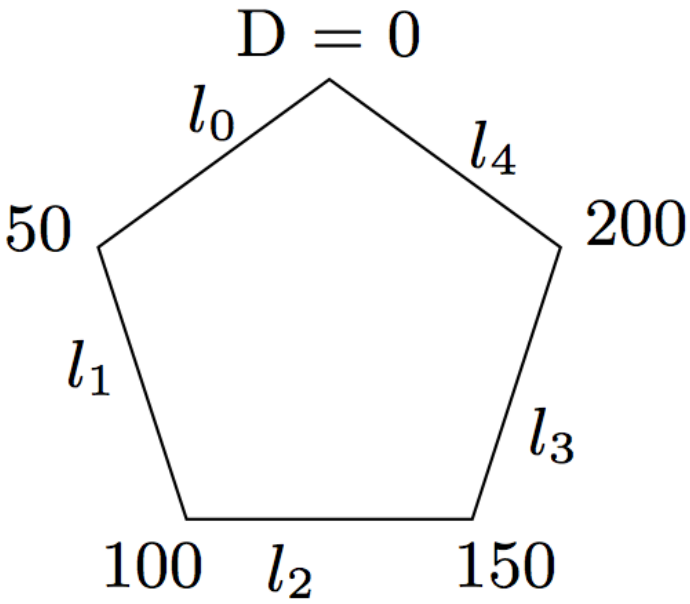
```
rl [rbcgrant]:
  { grant(N)
    <O : RBC | state : wait, lasttrain : T1, ma : MAP1 >
    REST
  }

=>

  { <O : RBC | state : rbcidle,
               ma : insert(T1,endoftrack(N),MAP1) >
    grantma(T1, endoftrack(N))
    REST
  } .
```

δ operator – by Ölveczky and Thorvaldsen, 2007

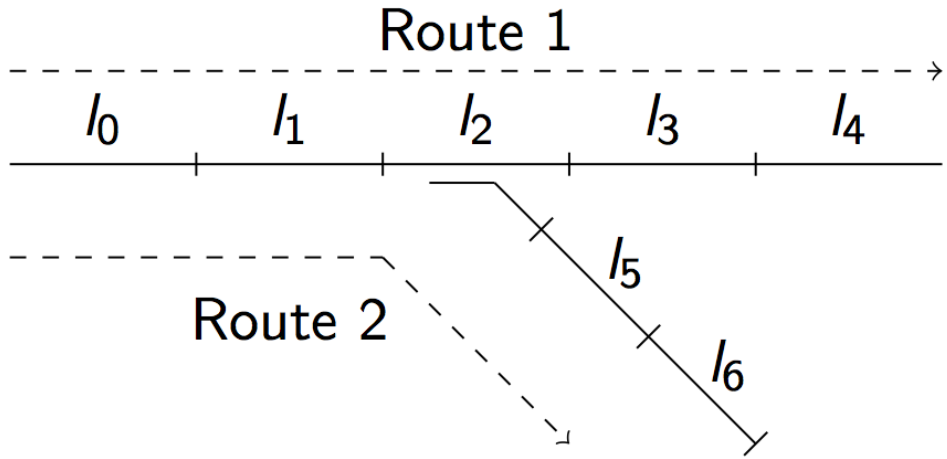
```
op delta : Configuration -> Configuration .
vars CON CON1 CON2 : Configuration .
var OCREST : ObjectConfiguration .


rl [timetrans] : {OCREST} => {delta(OCREST)} in time 1 .
rl [delta] : delta(CON1 CON2) => delta(CON1) delta(CON2) .

--- mte = maximal time elapse
rl [mte] : {CON} => {delta(CON)} in time 1 if mte (CON) >= 1 .

eq mte(none) = INF
eq mte(CON1 CON2) = min(mte(CON1), mte(CON2)).
eg mte (MSG) = 0
eg mte (<    >) = ....
```

Verification & simulation results

First examples with 2 trains in each scenario

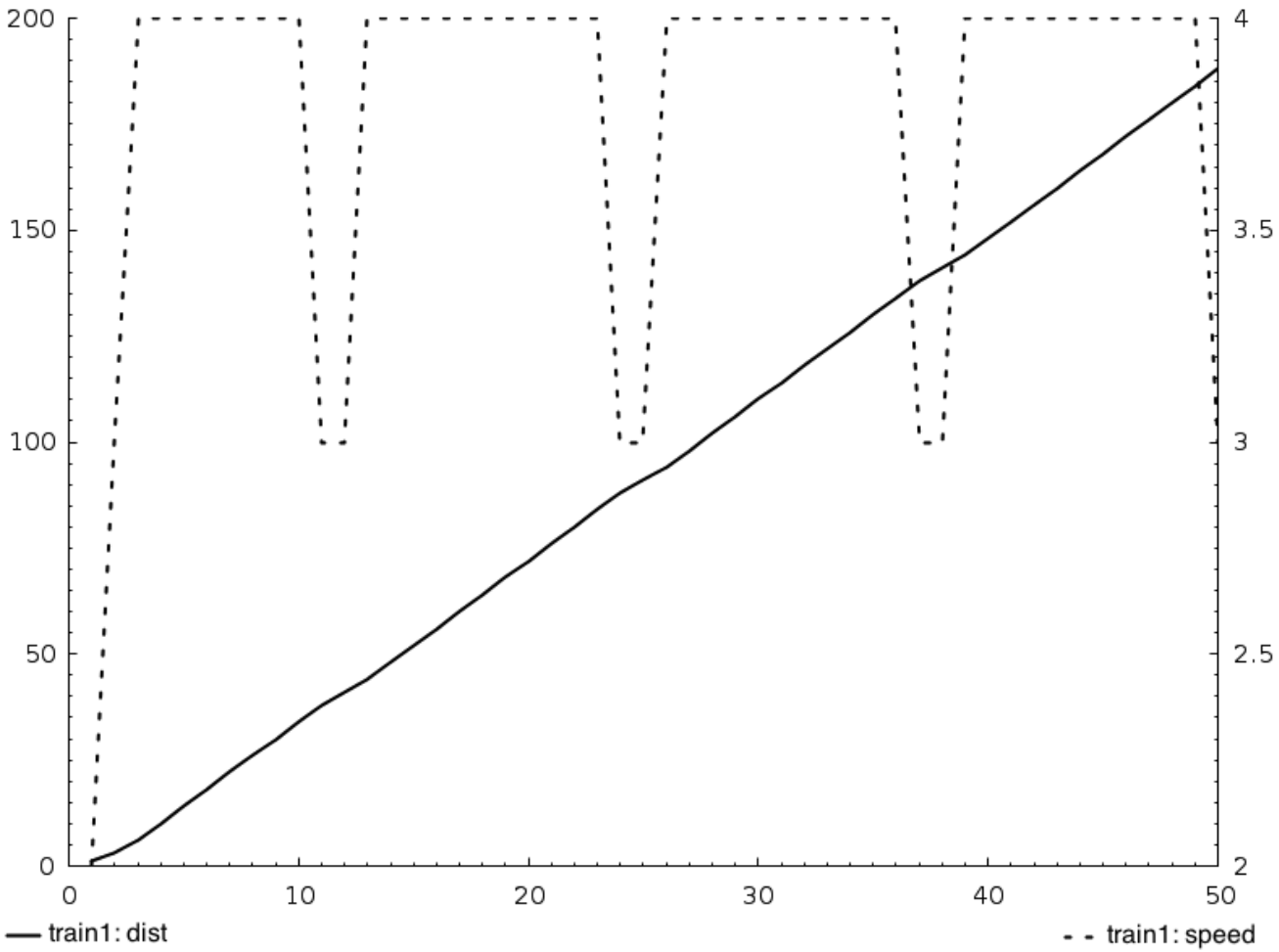


Pentagon-Example

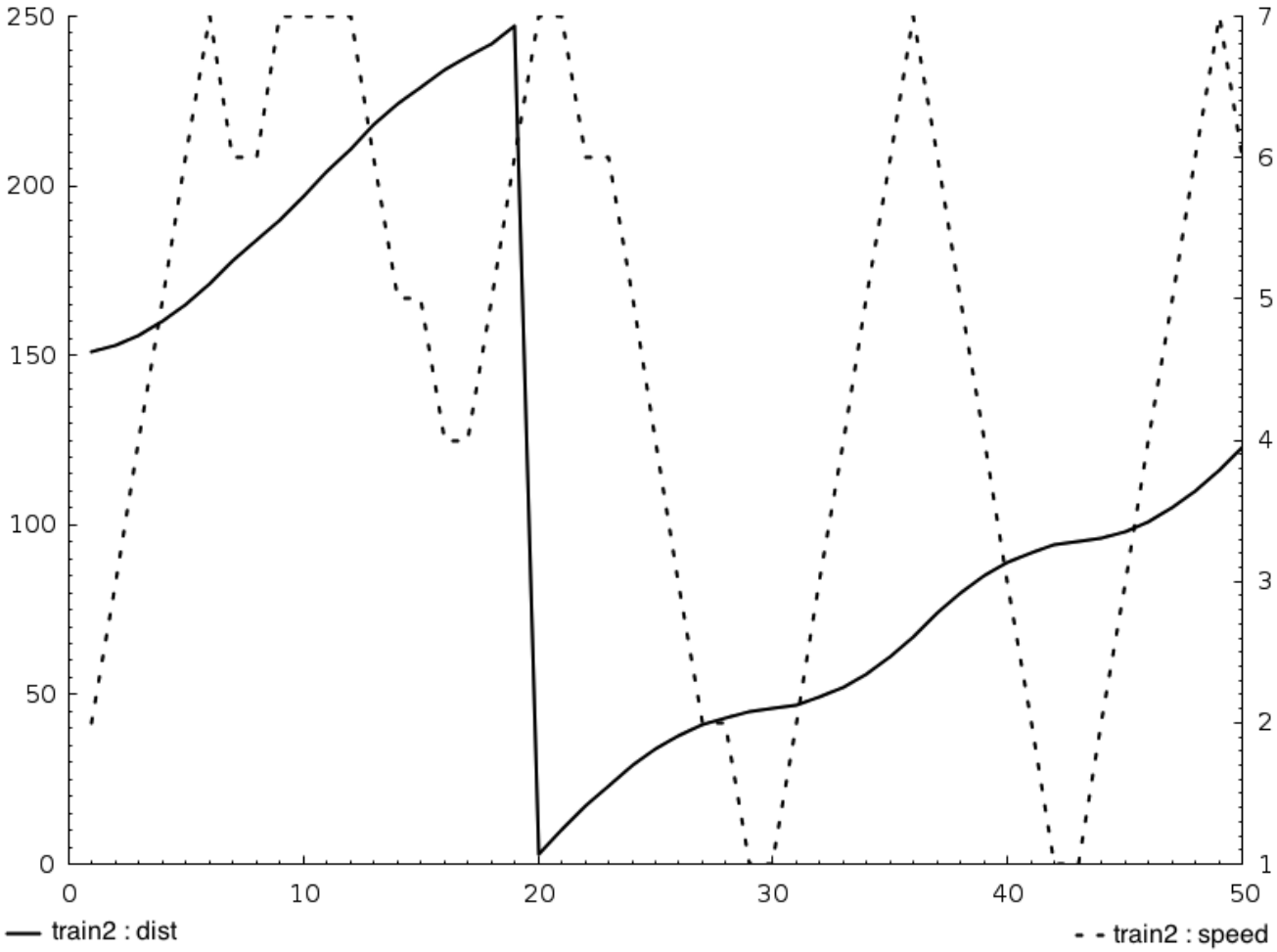


Junction

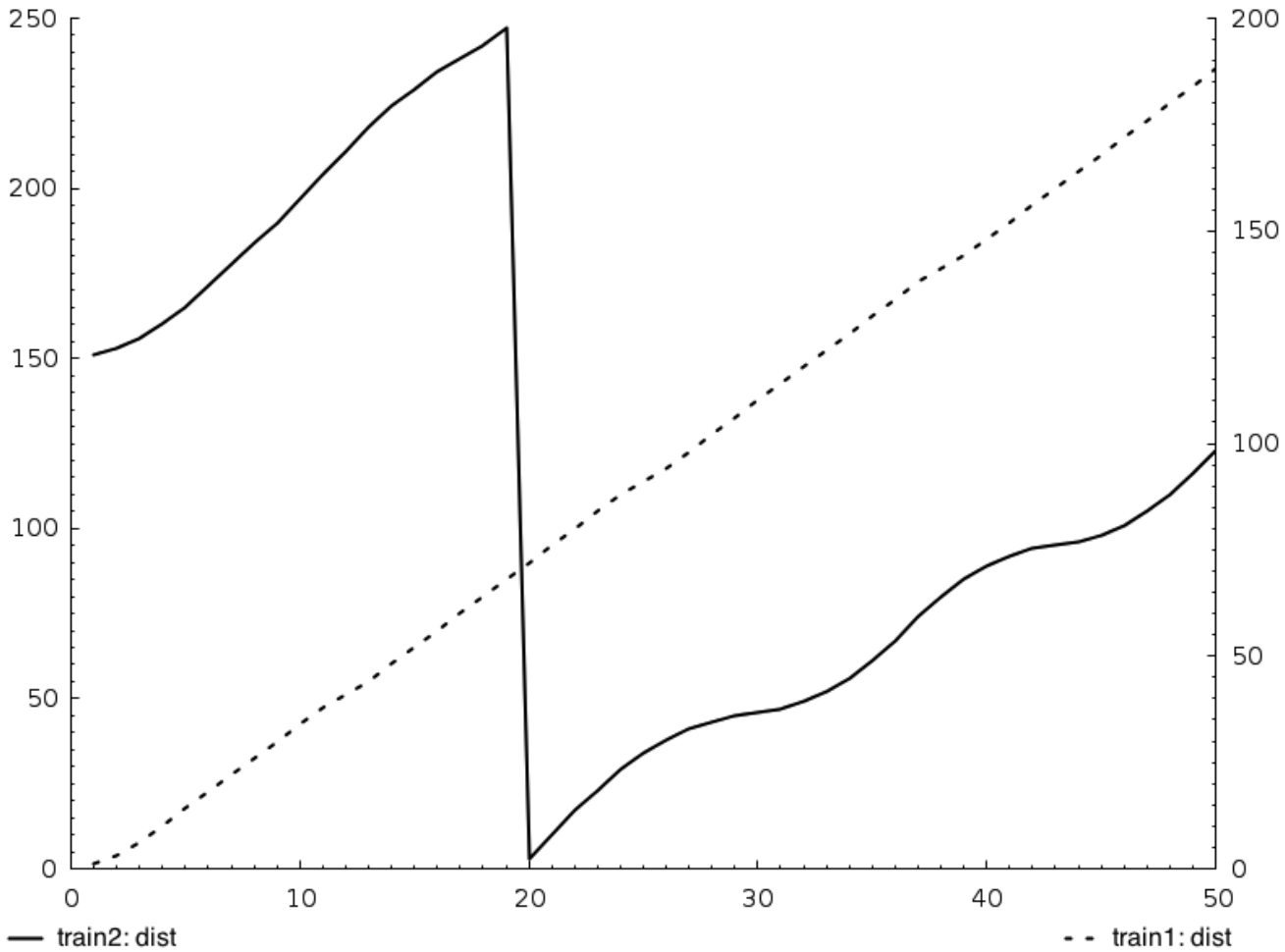
Train speed / distance over time – Pentagon example



Train speed / distance over time – Pentagon, 2nd train



Comparison of both trains



Safety-Verification by applying Real-Time Maude Model-checking for Linear Temporal Logic

Pentagon Example - 2 trains:

- no overlapping MA in the first 100 time units – takes 91 secs

Junction - 2 trains:

- no overlapping MA – takes 7 secs
- no derailment – takes 6 secs

Safety-Verification by applying Real-Time Maude Model-checking for Linear Temporal Logic

Pentagon Example - 2 trains:

- no overlapping MA in the first 100 time units – takes 91 secs

Junction - 2 trains:

- no overlapping MA – takes 7 secs
- no derailment – takes 6 secs

We also compared with realistic tracklengths of 5000 - speed 120/70

Pentagon Example - 2 trains:

- no overlapping MA in the first 70 time units – takes 79 secs

Ongoing work

- Modelling of the controller:
  currently acts as a scheduler in the second example:
  Knows of trains routes, max speed of each train, initializes trains.
  $\rightarrow$ Route setting, interaction with interlocking.
  $\rightarrow$ Cancellation of routes.

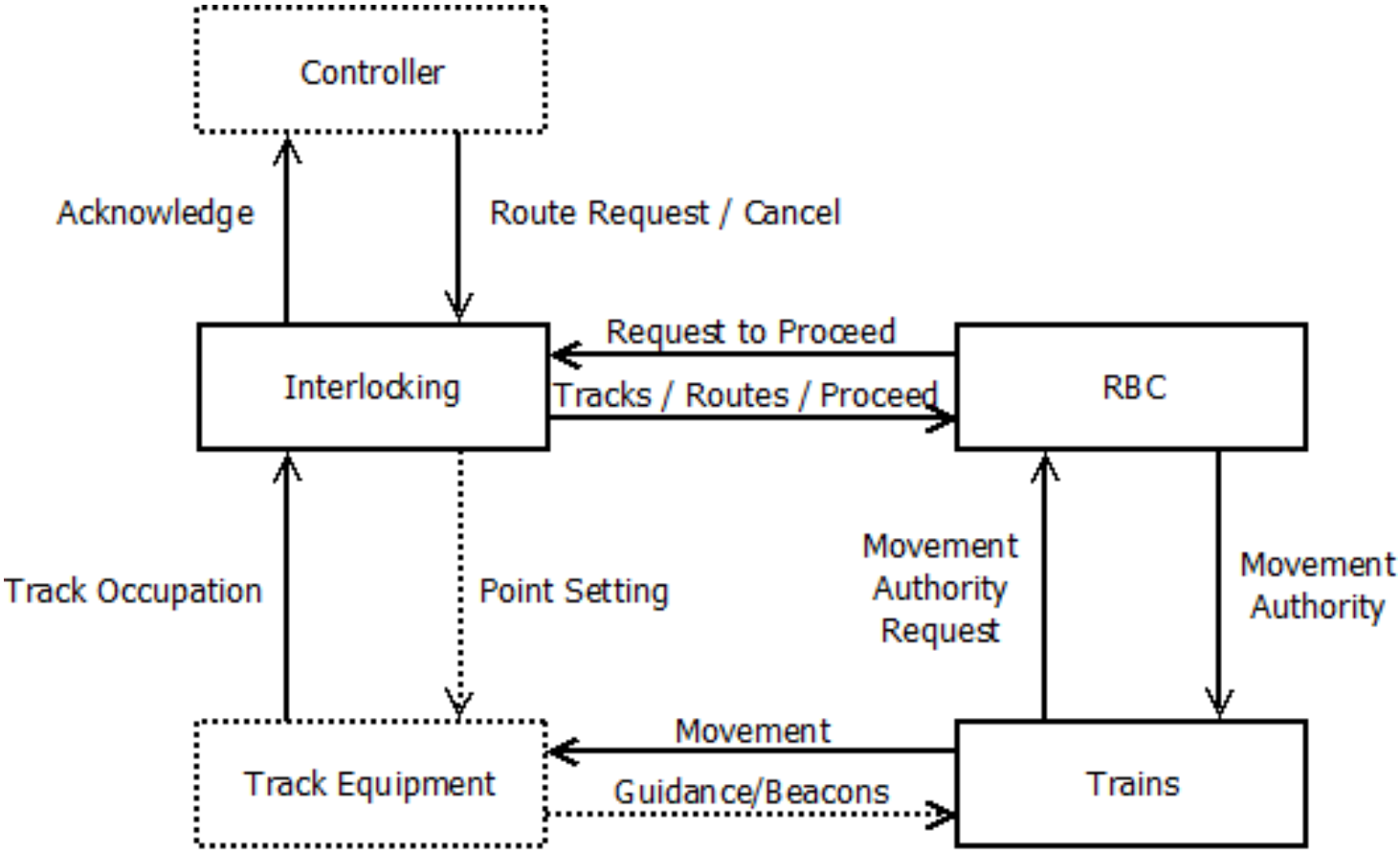
- Sequential release.

- Review of information flow.

- Generic track plans.

- Review of assumptions.

Information flow in ERTMS as implemented by Siemens



Conclusion & Future Work

- Our modelling approach works and is rich enough it to analyze ERTMS for important safety and capacity properties.
- Real-Time Maude offers a competitive framework for modelling & analysing ERTMS.

Future Work:

- Enrich the modelling to capture more aspects of ERTMS.
- Develop the timed simulations towards capacity measures.

- Aside: Final year student - Automatic translation of Ladderlogic into Real-Time Maude