DITTO Project Deliverable 1.3

Milestone 8

Towards Tool Interaction and Safety Assessment

June 2016

(Revised September 2016)

Prof Faron Moller, Prof Markus Roggenbach, Swansea University (editors)

With contributions by

- Professor John Preston, Dr John Armstrong and Dr Attila Kovacs, University of Southampton.
- Dr Ronghui Liu and Dr Hongbo Ye, University of Leeds.
- Dr Phillip James, Dr Xu Wang, Prof Faron Moller, and Prof Markus Roggenbach, Swansea University
- Dr Hoang Nga Nguyen, Coventry University
- Dr Lei Chen, Dr David Kirkwood and Dr Gemma L Nicholson, University of Birmingham

Table of Contents

List of Figures

List of Tables

Abstract

The development of railway systems is supported by a range of tools, each addressing individual, but overlapping concerns such as, e.g., performance, scheduling, and safety analysis. However, it is a challenge for users to organize work-flows; results are often in different, non-aligning data formats; and tools work on very different levels of abstraction, from macro- to micro-scopic. Thus, tool integration would be beneficial, and also allow for more powerful, experimental prototyping and design.

This report demonstrates how a number of different tools which are being developed within the context of the DITTO project can potentially cooperate. To this end, a common case study area was chosen from the East Coast Mainline and analysed using various tools.

Towards the goal of tool integration, a particular emphasis is placed on safety analysis and how this can be combined with simulation-based approaches as represented, e.g., by the BRaVE tool being developed in Birmingham within the DEDOTS project.

1 Introduction

Models are indispensable in Systems Engineering, e.g., of railway systems. As pre-images of the system under development, they provide purposeful abstractions that can be realized faster and cheaper than the final system. Design processes usually involve several models. In railway design, for instance, there are models for safety analysis of the control system [JMN+14], for capacity analysis [IMNR12], for timetable validation [PGB16], and for capacity utilization [BASP15] to name just a few.

Model representation concerns the question of how models are made accessible to computer programs. They range from what one might call a data format, such as the open source exchange format RailML, to fully fledged Domain Specific Languages (DSL). A data format provides names and typing information for the various model elements. In contrast to this, the purpose of a DSL is to provide a domain specific vocabulary in order to ease the description of domain artefacts. This vocabulary often comes with relations such as “a rail network consists of points, tracks and signals” and adds multiplicity constraints such as “a rail network has at least one station”.

Model analysis allows one to gain insights in the system under development and to make predictions on its properties and behaviour. Often, model analysis is computer-based; typical tools in railway design include BRaVE [BRa, DWK+16], OnTrack [JTT+13] and RailSys [RB01]. Model analysis can not only demonstrate that requirements are met, it can also reveal design faults. Consequently, model analysis is often part of quality assurance, e.g., in safety cases.

Model analysis comprises of a number of techniques, including simulation and model checking. Simulation is the imitation of the system’s operation over time, usually from a given initial situation. It results in a single model run that allows one to make predictions on how the system will operate under the chosen parameters. To obtain just a single run, models need to be made deterministic (e.g., by a progress assumption such as “trains will always proceed at a green signal” or by adding a random generator that takes such choices). The BRaVE tool, for example, includes dynamic multi-train simulation; furthermore, it utilizes simulation for railway operational analysis, system optimization, and system functional testing. In contrast, model checking exhaustively and automatically checks if all possible model runs satisfy a certain property such as “trains will never collide”. Most railway models are inherently non-deterministic, as train drivers can decide if, and at what speed, they want to proceed based on current circumstances (e.g., cattle on the track); also, train controllers can request and cancel routes at will. If model checking for a property is successful, the system will exhibit this

property; if unsuccessful, the model checker will provide a counter-example trace that can be utilized in the error analysis of the design. The OnTrack tool uses model checking to analyze if scheme plans are safe.

Challenges in model-based design include consistency and consolidation of models. Consistency concerns the question of whether or not different models produced for different purposes can be models of the very same system under development; the concern is that they might well contradict each other. Consolidation goes one step further: beyond requiring consistency, it asks if different models share a common basis. When designing railway systems, one often strives to consolidate models that share a track plan as their common basis but differ in the way they enrich track plans with further information such as control tables, timetables, track lengths, gradients and speed profiles. This additional information needs to be consistent; for example, the timetable should not stipulate travel times that are faster than the speed profile would allow for, taking into account the given track lengths.

1.1 Outline of the deliverable

In this report, we outline the functionality of the various tools, essentially following the order of the workflow diagrams presented in [Dit16], describing the use of the various tools as applied to a common case study area. Firstly, in Section 2 the common case study area is motivated and defined. In Section 3 we consider various approaches and tools which we developed and apply to safety analysis applied to the common case study area. In Section 4 we turn our attention to our static timetable optimization tools based on the analysis of capacity utilisation, whilst in Section 5 we outline the use of our timetable optimisation tools. In Section 6 we give an overview of the use of TrackULA, our rail network simulation tool. In Section 7 we describe our practical efforts towards integrating tools between the DITTO project – specifically OnTrack – and the DEDOTS project – specifically BRaVE – and demonstrate how to exploit the synergy created through this integration. Finally, in section 8 we summarise our findings.

2 Common case study area

[Communicated by J Armstrong, University of Southampton]

In developing a common case study area, consideration was given to the relative merits of a hypothetical example and an actual section of the national network, selected to meet all of the modelling requirements. A hypothetical approach has the advantage of allowing the specification of parameters to meet exactly the needs of the users, but requires the generation of hypothetical timetables as well as infrastructure data; and it has a significant disadvantage in that no actual performance data are available for analysis. An actual example tends to have the opposite drawbacks and advantages, in that model parameters are dictated and constrained by the characteristics of the chosen model area, but the example used is by its nature realistic, and actual timetable and performance data are available for analysis and review.

On balance, it was therefore decided to use an actual example based on part of the national network. A sub-section of the initial project study area (as described in more detail later in this document) was chosen, centred on Grantham and extending north and south to Newark North Gate and Highdyke Junction respectively (all of which is included in the initial project study area), but also extended east and west beyond the initial study area, to Ancaster and Bottesford respectively, both of which are located on the line between Nottingham and Sleaford, Boston and Skegness. A simplified representation of the model area, including distances, is shown in Figure 1.

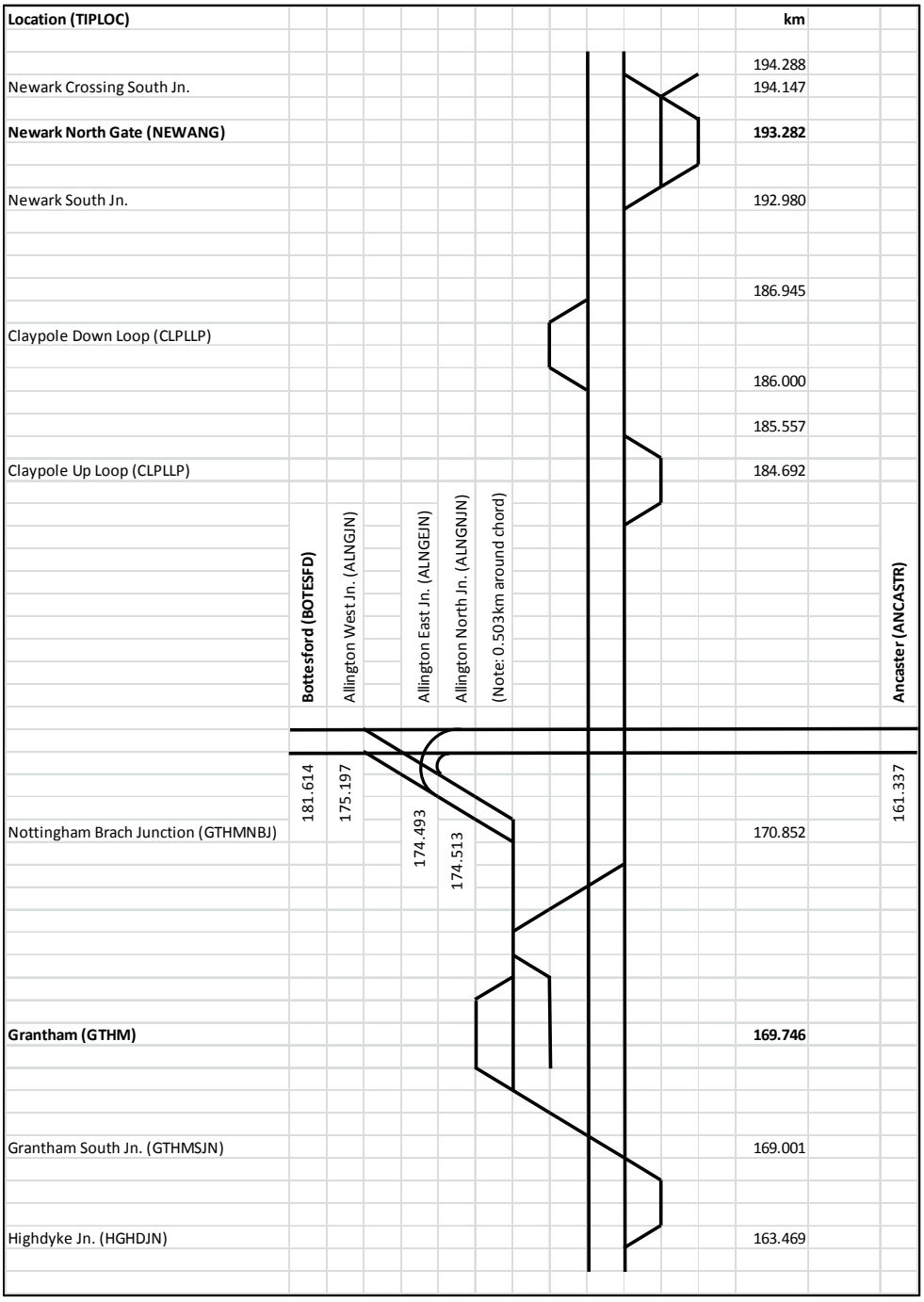

Figure 1: Common case study area

2.1 Case Study Details

Parts of the chosen model area are not included in the BRaVE model data, and have instead been generated by reference to 'Quail' maps and Five-Mile Diagrams; hence, the resulting model is something of a hybrid derived from a range of different data sources.

Reference to the timetable data for the selected area indicates that times are only specified at the eastern, western and southern model area boundaries (Ancaster, Bottesford and Highdyke Junction respectively) only for trains that stop at those locations; the passing times for trains which pass without stopping (which is the majority of services) are not specified. In order to estimate these times for the calculation of capacity utilisation values and investigate capacity utilisation - performance relationships, it was necessary to extend the model to include Sleaford, Bingham and Stoke Junction, for which times for all trains are included in the CIF data.

3 Safety Analysis

[Communicated by P James, X Wang, F Moller and M Roggenbach, Swansea University; and HN Nguyen, Coventry University]

From the common case study area, a number of nodes have been chosen for safety analysis. As a major objective in this project is in integration with the BRaVE tool developed in the DEDOTS project, the main criterion for choosing nodes to analyse was: which of these nodes have been included in the BRaVE model data. These are

- Allington – control table with 113 entries;
- Barkston – control table with 88 entries;
- Claypole – control table with 154 entries;
- Grantham – control table with 502 entries; and
- Newark – control table with 1190 entries.

Figures 2-4 show these track plans as displayed by the BRaVE tool.

We carried out our safety analysis using two different modelling approaches: one based on CSP||B, one based on CSP. Both of these are part of the OnTrack toolset; but having two independent verification approaches increases trust in the correctness results.

In the following, we describe our CSP||B modelling; discuss CSP modelling; and finally give the results of the verifications.

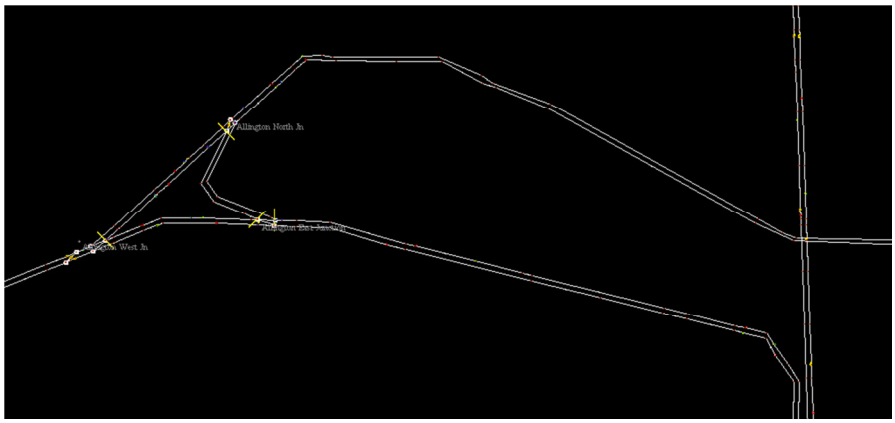

Figure 2: **Allington as displayed by BRaVE**

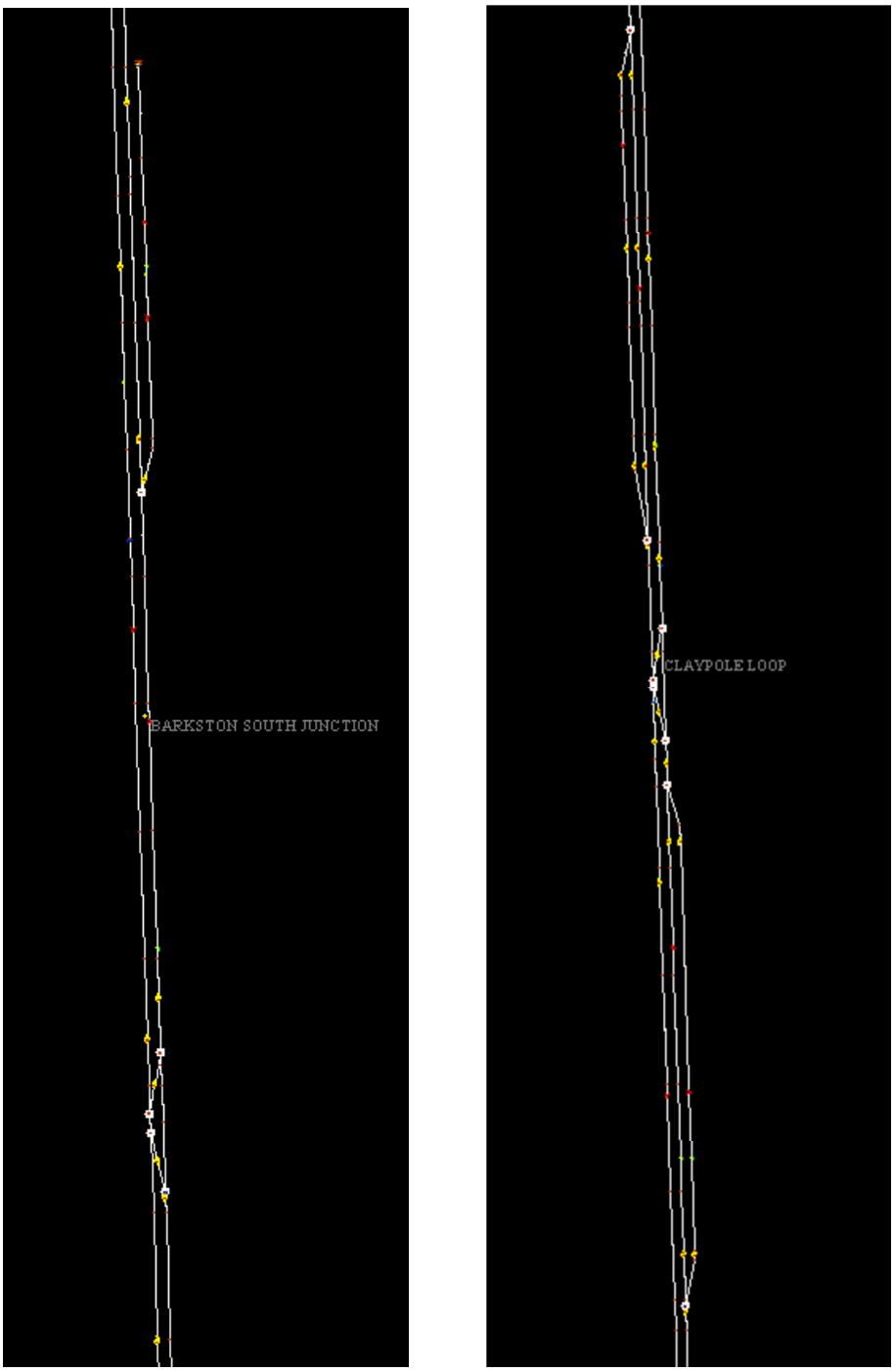

Figure 3: **Barkston and Claypole as displayed in BRaVE.**

Figure 4: **Grantham and Newark as displayed by BRaVe.**

3.1 CSP||B modelling

As we have discussed CSP||B modelling in detail in previous reports, we give here only a brief reminder.

In partnership with Siemens, we have developed a modelling approach which focuses on how information flows through the various elements of the railway. These have been identified to be Controller, Interlocking, Track Equipment and Trains. Each of these elements sends and receives information to the others. The Controller selects and releases routes. The Interlocking serves as a safety layer between Controller and Track Equipment. The Track Equipment consists of elements such as signals, points and track units. Some of these elements have states: Point can be in normal or reverse positions, and Signals can show proceed or halt. Finally, Trains have a driver who determines their behaviour. The data-rich Interlocking component is modelled by a single B-machine, while the Controller and Trains run independently of each other using the CSP interleaving operator. Thanks to having a generic model, one only has to instantiate the model with the location specific data. The purpose of these CSP||B models is to verify the correctness of the Control and Release Tables.

3.2 CSP modelling

Automated tools like FDR3 can assist railway engineers in analysing their signalling designs. The FDR3 tool is based on the CSP formalism. To exploit its strength, we need to model the signalling designs in CSP.

The CSP model consists of two parts: the static part and the dynamic part. The static part encodes the topology of track plans as well as the control data associated with it (i.e. control/release tables). Our modelling focuses on continuously track-circuited bi-directional track plans with only two-aspect (controlled) main signals. For instance, Figure 5 presents such a track plan for a simplified version of Swansea railway station.

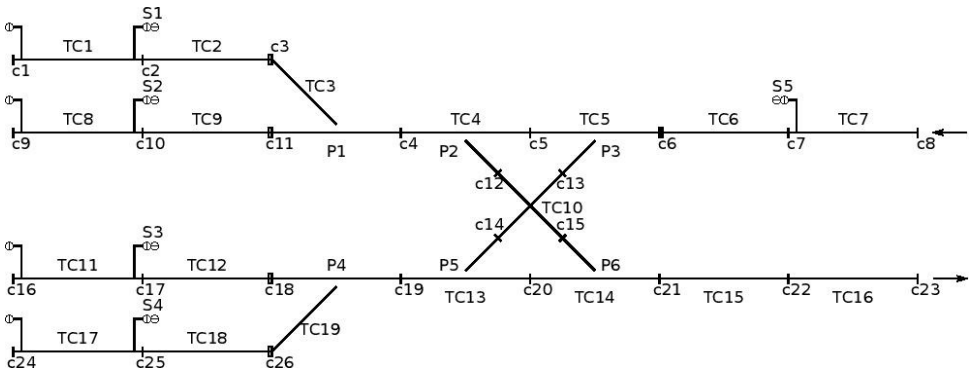

Figure 5: **Swansea train station track plan**

We can encode the track plan as the data structures in CSP code below.

```
datatype UnitID =
          TC1 | TC2 | TC3 | TC4 | TC5 | TC6 | TC7 | TC8 | TC9 | TC10 |
          TC11 | TC12 | TC13 | TC14 | TC15 | TC16 | TC17 | TC18 | TC19 |
          OFFTRACK | C8 | C23

-- points
datatype PointID = P1 | P2 | P3 | P4 | P5 | P6
-- point locations
pointAt(P1) = TC3     pointAt(P2) = TC4   ...

-- signals
datatype SignalID = S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9
-- signal locations
signalAt(S1) = (TC1, TC2)      signalAt(S2) = (TC8, TC9) ...

-- classification of track circuits
BiDirPointTC   =  {TC3, TC4, TC5, TC19, TC13, TC14}
BiDirLineTC    =  {TC1, TC2, TC8, TC9, TC6, TC7, TC11, TC12, TC17, TC18, TC15, TC16}
CrossingID     =  {TC10}

-- configuration according to classification
BiPointConfig  = {(TC3, TC4, TC9, TC2),
                  (TC4, TC3, TC5, TC10),
                  (TC5, TC6, TC4, TC10),
                  (TC19, TC13, TC12, TC18),
                  (TC13, TC19, TC14, TC10),
                  (TC14, TC15, TC13, TC10)}

BiLineConfig = {(TC1, TC2, OFFTRACK), (TC8, TC9, OFFTRACK), (TC11, TC12, OFFTRACK),
                (TC17, TC18, OFFTRACK), (TC2, TC1, TC3), (TC9, TC8, TC3),
                (TC12, TC11, TC19), (TC18, TC17, TC19), (TC6, TC5, TC7),
                (TC15, TC14, TC16), (TC7, TC6, C8), (TC16, TC15, C23) }
```

CSP data structures can also encode the control and release tables.

```
-- Control Table:
ctrlTable(R1)  =   ({P3},          {P1,P2,P6},  {TC2,TC3,TC4,TC10,TC14,TC15,TC16} )
ctrlTable(R2)  =   ({P1,P3},       {P2,P6},     {TC9,TC3,TC4,TC10,TC14,TC15,TC16} )
ctrlTable(R3)  =   ({P4,P5,P6},    {},          {TC12,TC19,TC13,TC14,TC15,TC16}  )
ctrlTable(R4)  =   ({P5,P6},       {P4},        {TC18,TC19,TC13,TC14,TC15,TC16} )
ctrlTable(R5a) =   ({P2,P3},       {P1},        {TC1,TC2,TC3,TC4,TC5,TC6})
ctrlTable(R5b) =   ({P1,P2,P3},    {},          {TC8,TC9,TC3,TC4,TC5,TC6})
ctrlTable(R5c) =   ({P2,P4},       {P3,P5},     {TC11,TC12,TC19,TC13,TC10,TC5,TC6})
ctrlTable(R5d) =   ({P2},          {P3,P4,P5},  {TC17,TC18,TC19,TC13,TC10,TC5,TC6})
ctrlTable(RC1) =   ({},            {},          {TC7})

-- Release Tables:
releaseTable(P1) = {  (R1, TC3, TC4), (R2, TC3, TC4),
                      (R5a,TC3, TC2), (R5b,TC3, TC9)  }

releaseTable(P2) = {  (R1, TC4,TC10), (R2, TC4,TC10),
                      (R5a,TC4,TC3), (R5b,TC4,TC3),
                      (R5c,TC10,TC13), (R5d,TC10,TC13)  }   ....
```

The dynamic part of the CSP code encodes a core subset of *signalling principles* from national or regional authorities [RSSB03, Kerr01] supporting features like: flank protection in route setting; front wheel replacement of signals; train operated route release; sequential release of route lockings; simplified version of comprehensive approach locking; and reversing of trains.

Such signalling principles are formulated as a set of generic rules to prescribe the dependency relation between state changes of different signalling elements (e.g. signal and points). They are modelled as a set of parameterised CSP processes.

As an example, we present here the CSP code modelling the functioning of a signal prescribed by the signalling rules related to bi-directionality, route setting/cancelling, front-wheel replacement, sectional release, etc.

```
SignalBehave(sID, st)=
    let (en,tk) = signalAt(sID) within
        ([] r: routeOpenBy(sID) @ st==Green & cancelRoute.r -> SignalBehave(sID,Red))
      [] ([] r: routeOpenBy(sID) @ st==Red  & setRoute.r -> SignalBehave(sID,Green))
      [] (st==Red & overrun.en.tk -> SignalBehave(sID,st))
      [] (st==Red & stop.en.tk -> SignalBehave(sID,st))
      [] ----------front wheel replacement of signal
         (st == Green & move.en.tk -> SignalBehave(sID,Red))
      [] ----------route release due to sequential release of opposing routes
         (st == Lock & move.tk.en -> SignalBehave(sID,Red))
      [] ----------route release due to cancelment of opposing routes
         ([] r: routeRversGuardBy(sID) @ st==Lock & cancelRoute.r -> SignalBehave(sID,Red))
      []   --- route holding due to the setting of opposing routes
         ([] r: routeRversGuardBy(sID) @ st==Red & setRoute.r -> SignalBehave(sID,Lock))
```

In the CSP code, the process SignalBehave has two state variables: sID representing the signal identifier; and st, which holds the current aspect that the signal is showing (i.e. Green or Red). The cancelRoute operation requires two preconditions: that the route r to be cancelled has sID as its entry signal, i.e. r: routeOpenBy(sID); and that the signal is currently displaying the Green aspect. After the cancelRoute operation, the state of the st variable will change to Red, i.e. SignalBehave(sID, Red).

Similarly the front wheel replacement will be implemented by line 8 of the code:

st == Green & move.en.tk -> SignalBehave(sID, Red)

which says that the train movement operation from track circuit “en” to track circuit “tk” will change the current signal aspect from Green to Red. Here “en” and “tk” are, respectively, the berth and overlap track circuits of signal sID, as defined by line 2 of the code: (en,tk)=signalAt(sID).

The dynamic modelling is the most critical part of the CSP modelling, since the number and size of CSP processes in the model strongly predicts the efficiency of the CSP model checking.

We model all the track circuits, points, signals and trains in a track plan as CSP processes, and our CSP modelling makes an effort to reduce the number and size of CSP processes in the model.

Firstly, we develop a graph-theoretic approach to prove that all three major safety hazards in the railway systems, collision, derailment and run-through, can be captured by a reduced model that does not encode track circuits as CSP processes.

So the FullSystem consists of only Points, Signals, and Trains, which are all drawn together in parallel:

```
TrackSystem = Points [|union(pntSysAlphRoute,inter(sigSysAlphMove,pntSysAlphMove))|] Signals
FullSystem = TrackSystem [|Union({trackSysAlphMove,trackSysAlphRoute})|] Trains
```

Secondly, we use an important principle of CSP model checking which says that a large number of small processes is more efficient than a small number of large processes. Thus, decomposing a large process into a number of smaller processes will significantly improve the performance of CSP model checking.

In our model the decomposition effort focuses on the largest CSP process in the model, i.e. point, which is reduced to two small processes: PointEntry and PointExit.

```
PointBehave(pID) =  PointEntry(pID, Norm) [Union({pntAlphRoute(pID), pntEnAlphMove(pID))
                 || Union({pntAlphRoute(pID), pntExAlphMove(pID))] PointExit(pID, {}, {})
```

By combining the static part with the (reduced) dynamic part, we essentially instantiate the generic signalling principle with specific scheme plans, SigPrinciple(SchemePlan). As a result, we obtain the complete CSP model of control logics for the interlocking under consideration, which is then ready to be fed into our model checking tool, FDR3.

3.3 Verification results

To illustrate the verification capabilities of our modelling, we have verified a section of the East Coast Main Line from Newark to Grantham. In particular, we have considered the regions of Newark, Claypole, Barkston, Allington and Grantham. For this, we have used data imported from the BRaVE toolset, which is based on simulation data from Network Rail. The translation process involved writing the graph traversal algorithm where the following key adjustments were made to the data:

- Overlaps: Firstly, the modelling assumes that all routes are protected by an overlap. However, spurious counter examples in the verification process highlighted that the data that had been extracted from BRaVE does not currently explicitly highlight overlaps. This was not a problem with the BRaVE data, where overlaps are just treated as specialised track circuits, but was a problem due to the lack of their explicit representation. To solve this problem,

the translation process was altered to include the next track circuit after a route ends as an overlap.

- Track Circuits around points: Within the BRaVE data, points are modelled as a particular node with three track circuits being associated to the incoming, normal and reverse branches of the point. However, in our modelling points are assumed to be a single track circuit. This is due to a modelling decision taken within the East Coast Main Line data used within BRaVE. However, to enable a current integration, the translation process was augmented with a manual step to collapse the three track circuits from the BRaVE data into a single track circuit. A similar process was also undertaken for two of the plans (Allington and Newark), where large track circuits containing multiple crossings were split into individual track circuits per crossing.
- Addition of "Entry" and "Exit" tracks: Finally, the last addition that has to be added to the translation is a consideration towards the "open" end points of a particular scheme plan. As the BRaVE data models the whole East Coast Main Line, when particular scheme plans or regions are extracted for verification, there is clearly going to be areas on these scheme plans where trains can enter and exit the plan. As within the BRaVE data these areas do not exist, there is the need to add them for use within our modelling. Such an addition is a fairly straightforward extension of our integration process. It simply involves a post translation step that considers the translated scheme plan and looks for track circuits that do not have a "successor" or "predecessor" track circuit attached to them. At these points, specialised "Entry" and "Exit" track circuits can be added to the model.

3.3.1 Verification in CSP||B

The rail nodes Barkston, Claypole, and Allington can be verified using CSP||B modelling, the principle of covering decomposition advocated by [JMN+14] and other papers and the ProB model-checking tool. The decomposition produces one sub-scheme plan per track circuit.

The verification of the sub-scheme plans of Barkston take less than 5 hours each, for Allington and Claypole they take less than 7 minutes, see Tables 1 to 3 below. We address the question of why these verification times differ in Section 3.3.3.

In these tables, the abbreviations for the columns mean: SP – sub-scheme plan generated by the track circuit named in the column; #TC - number of track circuits in the sub-scheme plan; #Pt: number of points in the sub-scheme plan; #Sn - number of signals in the sub-scheme plan; #Rt: Number of (hidden) routes in the sub-scheme plan. Time - the run-time for the model-checker ProB; #States - number of states inspected; Safety – 'ok' indicates that the considered sub-scheme plan is collision free, has no run-

throughs, and also no derailments. The original scheme plan is safe, if all sub-scheme plans are checked with the result 'ok'.

SP	#TC	#Pt	#Sn	#Rt	Time	#State	Safety
TC3327	13	4	3	5(1)	2h09m19s	28344	ok
TC3343	13	4	3	5(1)	2h09m09s	28344	ok
TC3341	13	4	3	5(1)	2h09m21s	28344	ok
TC3351	2	0	1	1	1s	69	ok
TC3346	3	0	1	1	0s	83	ok
TC3318	2	0	1	2	2s	101	ok
TC3319	13	4	3	5(1)	2h09m12s	28344	ok
TC3323	13	4	3	5(1)	2h09m43s	28344	ok
TCE4	8	4	1	2(4)	4m00s	2396	ok
TC3339	16	4	4	6	3h59m16s	52526	ok
TC3325	13	4	3	5(1)	2h09m51s	28344	ok
TC3337	16	4	4	6	4h41m56s	52526	ok
TC3334	6	2	1	2(1)	24s	521	ok
TC3335	16	4	4	6	3h58m55s	52526	ok
TC3315	7	4	1	2(4)	3m19s	2184	ok
TC3330	16	4	4	6	4h41m28s	52526	ok

Table 1: **Verification of Barkston in CSP||B**

SP	#TC	#Pt	#Sn	#Rt	Time	#State	Safety
N1229	6	1	1	2(2)	9s	242	ok
TC394	3	0	1	2	1s	75	ok
TC395	5	1	1	2(1)	12s	446	ok
TC397	2	0	1	2	0s	59	ok
TC390	8	2	2	2(3)	30s	452	ok
TC392	11	2	2	4(1)	3m14s	2040	ok
TC393	10	2	2	4(1)	2m53s	1872	ok
TC2924	2	0	1	1(2)	0s	39	ok
N1253	5	1	2	3(3)	31s	431	ok
N1252	7	2	2	2(3)	30s	410	ok
TC1499	2	0	1	1(2)	0s	39	ok
N1293	3	1	1	1	1s	52	ok
N1257	12	3	2	4(1)	4m07s	2624	ok
N1259	9	2	2	4(1)	2m31s	1704	ok
TC2862	2	0	1	1(1)	0s	39	ok
TC383	2	0	1	2(1)	1s	59	ok
TC382	13	3	2	4(1)	5m12s	3112	ok
TC381	2	0	1	1(2)	0s	39	ok
TC387	4	1	1	1(1)	1s	55	ok
TC386	3	1	1	1(1)	1s	47	ok
TC385	4	1	1	2(1)	4s	117	ok
TC389	4	1	1	2(1)	14s	378	ok
TC388	7	2	2	3(1)	29s	404	ok
TC371	4	1	1	1	1s	60	ok
N1288	9	2	2	3(4)	2m00s	924	ok
TC3297	2	0	1	1	0s	39	ok
N1241	3	1	1	2(1)	3s	87	ok
N1246	7	2	2	3(1)	29s	404	ok
TC377	14	3	2	4(1)	6m01s	3600	ok
N1245	2	1	1	1(1)	0s	39	ok
TC378	5	1	1	2(2)	12s	206	ok
TC379	2	0	1	1	0s	39	ok
N1265	13	3	2	4(1)	4m50s	3112	ok
TC401	10	2	2	3(4)	2m17s	1020	ok
TC402	2	0	1	1	0s	39	ok

Table 2: **Verification of Allington in CSP||B**

SP	#TC	#Pt	#Sn	#Rt	Time	#State	Safety
TC3437	11	3	1	2(3)	1m10s	713	ok
TC3436	11	3	1	2(3)	1m11s	713	ok
TC3423	9	3	1	2(3)	44s	541	ok
N10967	7	1	2	2(4)	16s	225	ok
N10988	16	5	1	4(2)	3m46s	1526	ok
TC3389	3	0	1	1(2)	1s	47	ok
TC3388	8	1	2	2(4)	21s	249	ok
TC3386	2	0	1	1	0s	39	ok
TC3383	9	1	2	2(4)	27s	273	ok
TC3412	6	2	1	2(3)	15s	257	ok
TC3408	5	1	1	4	16s	271	ok
TC3409	10	3	1	4(2)	2m40s	1426	ok
TC3428	10	3	1	2(3)	59s	627	ok
TC3426	10	3	1	2(3)	1m00s	627	ok
TC3401	17	5	1	4(2)	5m14s	1922	ok
TC3424	9	3	1	2(3)	46s	541	ok
TC3403	17	5	1	4(2)	5m02s	1922	ok
TC3404	15	4	1	4(2)	2m48s	1242	ok
TC3405	14	4	1	4(2)	2m24s	1130	ok
TC3420	2	0	1	4	3s	99	ok
TC3407	11	3	1	4(2)	3m32s	1714	ok
N10993	11	3	2	6(1)	6m42s	2221	ok
N10992	13	4	1	4(2)	2m02s	1018	ok
N10995	4	1	1	4	9s	175	ok
N10994	11	3	2	6(1)	6m43s	2221	ok
TC3390	3	0	1	1(2)	1s	47	ok
TC3392	2	0	1	1(2)	0s	39	ok
TC3394	2	0	1	1(2)	0s	39	ok
TC3395	18	5	1	4(2)	6m13s	2318	ok
TC3396	18	5	1	4(2)	6m23s	2318	ok
TC3399	3	0	1	1	0s	47	ok
TC3415	3	0	1	4	5s	131	ok
TC3414	5	1	1	4	16s	271	ok
TC3413	6	1	1	4	25s	367	ok
N11010	8	3	1	2(3)	34s	455	ok
TC3411	2	0	1	2(1)	1s	59	ok
TC3410	11	3	2	6(1)	6m41s	2221	ok
TC3422	7	2	1	2(3)	20s	313	ok

Table 3: Verification of Claypole in CSP||B

For Grantham (see Table 4 below), we obtain a mixed picture with regards to tool performance. Those sub-scheme plans that can be verified, take under four hours. However, a number of these sub-scheme plans lead to state space explosion, i.e., the tool does not provide any verification result due to lack of resources. A typical instance of this would be the sub-scheme plan for track circuit TC2795, which consists of 23 track circuits, 6 points, 5 signals, and 8 routes.

SP	#TC	#Pt	#Sn	#Rt	Time	#State	Safety
TC2853	22	5	3	5(5)	3h43m55s	19934	ok
TC2795	23	6	5	8(6)	time-out		
TC2831	19	3	4	7(5)	3h21m45s	13381	ok
TC2872	6	1	1	2	4s	155	ok
TC2835	3	0	1	1(1)	0s	43	ok
TC2880	19	3	4	7(5)	3h22m44s	13381	ok
TC2855	17	4	2	4(6)	1h04m08s	9150	ok
TC2793	25	6	6	10(5)	time-out		
TC2846	2	0	1	1(1)	0s	35	ok
TC2823	5	1	2	5(5)	1m41s	699	ok
TC2844	2	0	1	1(1)	0s	35	ok
TC2779	3	0	1	1	0s	5	ok
TC2778	23	6	5	8(8)	time-out		
TC2841	22	5	3	5(5)	3h45m17s	19934	ok
TC2794	7	2	1	3(6)	48s	575	ok
TC2774	21	5	5	8(7)	time-out		
TC2799	9	2	1	3(6)	1m23s	763	ok
TC2800	8	2	1	3(6)	1m05s	669	ok
TC2807	20	6	4	5(9)	time-out		
TC2849	6	1	1	2	6s	155	ok
TC2848	3	0	1	1(1)	0s	43	ok
N9592	25	6	6	9(7)	time-out		
N9593	25	6	6	10(5)	time-out		
N9717	22	5	3	5(5)	3h44m21s	19934	ok
TC2885	17	3	2	3(7)	43m45s	5953	ok
TC2865	17	4	2	4(6)	1h04m42s	9150	ok
TC2864	17	4	2	4(6)	1h04m42s	9150	ok
TC2867	4	0	1	1(1)	1s	51	ok
TC2866	21	4	3	5(6)	2h34m36s	12458	ok
TC2861	2	0	1	1	0s	35	ok
TC2860	8	2	1	2(6)	2m53s	1398	ok
TC2829	16	3	3	6(5)	1h43m10s	10888	ok
TC2777	2	0	1	3	1s	67	ok
N9660	20	3	5	8(4)	3h57m49s	14541	ok
N9609	25	6	6	10(5)	time-out		
TC2857	3	0	1	1	0s	43	ok
N9733	21	4	3	5(6)	2h34m55s	12458	ok
N9653	16	3	3	6(5)	1h43m21s	10888	ok
N9658	19	3	4	7(5)	3h22m14s	13381	ok
TC2883	16	3	2	3(7)	37m23s	5505	ok
TC2840	41	8	8	13(3)	time-out		

TC2818	14	3	2	5(6)	1h19m53s	11265	ok
TC2828	13	3	2	5(5)	54m42s	9671	ok
TC2856	7	2	1	2(6)	2m17s	1246	ok
TC2854	9	2	2	3(5)	2m25s	1021	ok
TC2786	3	0	1	2(1)	2s	67	ok
TC2787	22	6	5	7(8)	time-out		
TC2780	23	6	5	8(8)	time-out		
TC2781	25	6	6	9(7)	time-out		
TC2783	2	0	1	1	0s	5	ok
TC2816	11	2	2	5(6)	19m06s	3637	ok
TC2817	30	8	6	10(13)	time-out		
TC2788	22	6	5	7(8)	time-out		
TC2789	25	6	6	10(5)	time-out		
TC2812	2	0	1	2(1)	1s	51	ok
TC2813	4	0	1	1(1)	1s	51	ok
TC2832	19	3	4	7(5)	3h22m20s	13381	ok
N9583	25	6	6	9(7)	time-out		
N9582	26	6	6	9(7)	time-out		
N9606	25	6	6	10(5)	time-out		
TC2796	21	6	4	7(6)	time-out		
TC2878	2	0	1	2	1s	51	ok
TC2833	20	3	5	8(4)	3h57m21s	14541	ok
TC2830	16	3	3	6(5)	1h43m17s	10888	ok
TC2820	14	3	2	5(6)	1h20m04s	11265	ok
TC2874	3	0	1	2	1s	67	ok
TC2875	5	1	1	2	4s	125	ok
N9743	4	1	1	2	3s	95	ok
N9742	21	4	3	5(6)	2h34m42s	12458	ok
TC2870	5	1	1	2	4s	125	ok
TC2871	21	4	3	5(6)	2h34m27s	12458	ok
TC2798	23	6	5	8(6)	time-out		
TC2790	3	0	1	3	3s	91	ok
TC2802	34	9	6	12(10)	time-out		
TC2826	13	3	2	5(5)	54m38s	9671	ok
TC2825	11	2	2	5(6)	19m02s	3637	ok
N9720	9	2	2	3(5)	2m27s	1021	ok
TC2837	20	3	5	8(4)	3h57m02s	14541	ok

Table 4: **Verification of Grantham in CSP||B**

With regards to Newark (see Table 5 below), the picture gets even worse. Most of the sub-scheme plans fail to verify. Those that can be verified vary from trivial to moderate complexity.

SP	#TC	#Pt	#Sn	#Rt	Time	#State	Safety
TC3558	35	13	3	23(13)	time-out		
TC3559	37	13	4	31(5)	time-out		
TC3555	27	7	6	34(9)	time-out		
TC3553	6	1	2	3(7)	2m15s	1074	ok
TC3497	16	4	4	14(20)	time-out		
TC2626	4	0	1	1(5)	6s	108	ok
TC3494	3	0	1	11	37s	327	ok
TC3492	2	0	1	11	24s	239	ok
TC3529	24	5	5	25(8)	time-out		
TC3528	34	12	4	31(5)	time-out		
TC3523	29	10	4	31(5)	time-out		
TC3522	26	8	4	31(5)	time-out		
TC3521	26	8	4	31(5)	time-out		
TC3520	24	7	4	31(5)	time-out		
TC3527	34	12	4	31(5)	time-out		
TC3526	32	11	4	31(5)	time-out		
TC3525	31	11	4	31(5)	time-out		
TC3524	31	11	4	31(5)	time-out		
TC3548	35	12	4	31(5)	time-out		
TC3545	29	10	4	31(5)	time-out		
TC3506	27	7	5	27(11)	time-out		
TC3542	27	9	4	31(5)	time-out		
TN8885	24	6	4	8(5)	time-out		
TC3579	31	7	7	12(9)	time-out		
TC3571	6	1	2	3(7)	2m19s	1074	ok
TC3572	6	1	2	3(7)	2m19s	1134	ok
TN8912	4	0	2	3(4)	24s	274	ok
TN8913	12	3	3	6(3)	time-out		
TN8910	27	7	5	8(8)	time-out		
TN8911	19	4	4	5(11)	time-out		
TC2627	19	4	3	4(12)	time-out		
TN11229	33	12	4	31(5)	time-out		
TN11228	30	11	4	31(5)	time-out		
TN11225	25	8	4	31(5)	time-out		
TN11222	24	7	4	31(5)	time-out		
TN11221	27	7	5	33(4)	time-out		
TC2622	12	3	2	4(7)	41m10s	7605	ok
TN11298	9	1	3	4(9)	9m26s	1939	ok
TN8930	16	4	3	5(8)	time-out		
TN11201	4	1	1	11	56s	427	ok
TN11202	30	7	6	28(11)	time-out		

TN11205	30	7	6	28(11)	time-out		
TN11206	29	7	6	35(3)	time-out		
TC3563	36	13	3	23(13)	time-out		
TC3562	41	14	4	31(5)	time-out		
TN11286	28	9	4	31(5)	time-out		
TC2616	2	0	1	2	1s	59	ok
TC2617	16	4	3	5(8)	time-out		
TC2614	14	4	3	6(4)	time-out		
TC2615	12	4	2	4(6)	2h04m51s	30717	ok
TC2612	4	0	1	1	1s	55	ok
TC2613	14	4	3	6(4)	time-out		
TC2611	4	0	1	2(4)	10s	154	ok
TC3581	2	0	1	1(3)	1s	39	ok
TC3583	3	0	1	1(3)	1s	47	ok
TC3584	3	0	1	1	1s	47	ok
TC3587	2	0	1	1	0s	39	ok
TC2587	2	0	1	1(1)	0s	39	ok
TC2585	5	1	1	11	1m32s	603	ok
TC2589	24	5	5	25(8)	time-out		
TC3512	20	5	4	31(6)	time-out		
TC3513	27	7	5	33(4)	time-out		
TC3510	29	7	6	35(3)	time-out		
TC3517	24	7	4	31(5)	time-out		
TC3514	24	7	4	31(5)	time-out		
TC3515	23	6	4	31(6)	time-out		
TC2605	5	0	1	2(4)	13s	178	ok
TC2604	3	0	1	1(1)	1s	47	ok
TC2606	11	3	2	3(8)	44m57s	8867	ok
TC2600	17	3	3	5(10)	3h31m21s	13774	ok
TC2608	12	3	2	3(8)	53m38s	9597	ok
TC2597	14	4	3	6(4)	time-out		
TC2596	13	3	2	4(9)	46m29s	6972	ok
TC2591	12	2	2	3(7)	10m32s	2098	ok
TC2593	30	7	7	12(9)	time-out		
TC2599	13	3	2	4(9)	46m34s	6972	ok
TC2598	12	2	2	3(7)	10m28s	2098	ok
TC3501	5	1	1	11	1m54s	747	ok
TC3500	24	6	5	17(21)	time-out		
TC3503	6	1	1	11	2m55s	1067	ok
TC3502	30	7	6	28(11)	time-out		
TC3505	14	3	3	22(4)	time-out		
TN8920	14	4	3	6(4)	time-out		

TC3508	27	7	5	27(11)	time-out		
TN8886	17	3	3	5(10)	time-out		
TN8929	14	4	3	6(4)	time-out		
TN11276	36	13	4	31(5)	time-out		
TN11279	40	14	4	31(5)	time-out		
TN11214	29	7	6	35(3)	time-out		
TC3538	28	10	4	31(5)	time-out		
TC3535	26	7	6	34(9)	time-out		
TN8897	16	4	3	4(9)	4h14m31s	26417	ok
TN8896	17	3	3	5(10)	3h32m00s	13774	ok
TC3532	25	6	6	34(9)	time-out		

Table 5: Verification of Newark in CSP||B

Grantham and Newark are the first instances in our verification practice that the decomposition technique fails to provide sub-scheme plans small enough to be verified with CSP||B modelling and the ProB model-checking tool. This negative result thus demonstrates that further research is necessary in order to overcome this restriction.

3.3.2 Verification in CSP

The rail nodes Barkston, Claypole, and Allington could be directly verified using the model checker FDR3. See below for the runtime of the verification process of these nodes.

Barkston	Points	Signals	Routes	Time	State	Result
Collision free	5	7	11	<1s	94,308	Pass
Derailment free	5	7	11	<1s	92,620	Pass
Runthrough free	5	7	11	<1s	92,620	Pass

Claypool	Points	Signals	Routes	Time	State	Result
Collision free	7	8	11	2.73s	3,359,908	Pass
Derailment free	7	8	11	2.50s	2,714,480	Pass
Runthrough free	7	8	11	2.40s	2,714,480	Pass

Allington	Points	Signals	Routes	Time	State	Result
Collision free	8	8	15	118.48s	119,442,052	Pass
Derailment free	8	8	15	105.11s	103,288,452	Pass
Runthrough free	8	8	15	102.34s	103,288,452	Pass

For Grantham station, we first build the CSP model for the whole station and feed it to our model checking tool, FDR3. This resulted in state explosion. After exhausting 32G of memory space, the tool failed to reach any conclusion.

This is not surprising since according to our experience, our CSP verification approach scales to scheme plans with 10 - 12 signals and 10 - 12 points. However, Grantham station as a whole is way beyond this limit, i.e. with 14 points and 27 signals.

Hence we have to decompose Grantham station. That is, the Scheme Plan of Grantham station (c.f. Figure 4 for the track plan) is decomposed into two parts based on the principle of covering decomposition advocated by [JMN+14] and other papers into a left and a right part (c.f. Figure 6 and Figure 7).

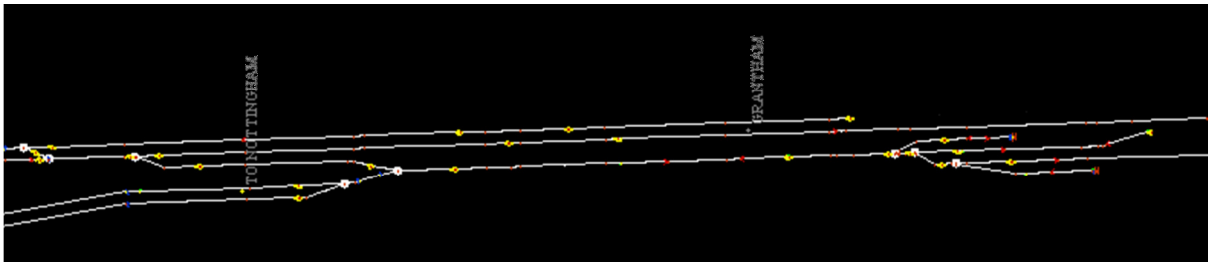

Figure 6: **Grantham station (Left part)**

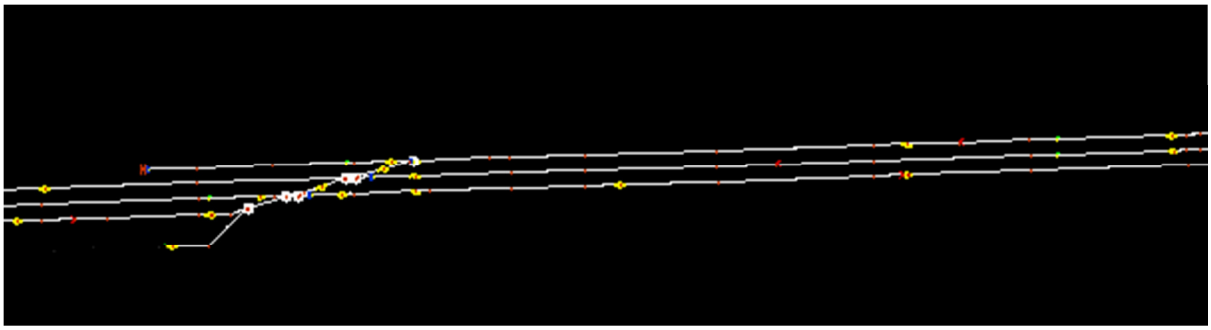

Figure 7: **Grantham station (Right part)**

Our decomposition tries to divide the track plan as evenly as possibly allowed by the covering decomposition (which requires that there is a small overlap between the two parts). The decomposition proves to be successful. FDR3 verifies both within a few minutes c.f. the tables below.

Grantham: left	Points	Signals	Routes	Time	State	Result
Collision free	8	14	22	392.90s	217,592,974	Pass
Derailment free	8	14	22	270.47s	164,584,900	Pass
Runthrough free	8	14	22	265.51s	164,584,900	Pass

Grantham: right	Points	Signals	Routes	Time	State	Result
Collision free	6	13	23	36.36s	29,265,164	Pass
Derailment free	6	13	23	33.00s	24,746,209	Pass
Runthrough free	6	13	23	31.85s	24,746,209	Pass

Newark station is even larger in size. It has 34 signals and 22 points. Not surprisingly, our CSP model of the whole station encountered state explosion when directly fed into FDR3.

Furthermore, with covering decomposition, we can decompose the whole station into two parts: the left part and the right part (c.f. Figure 8 and 9).

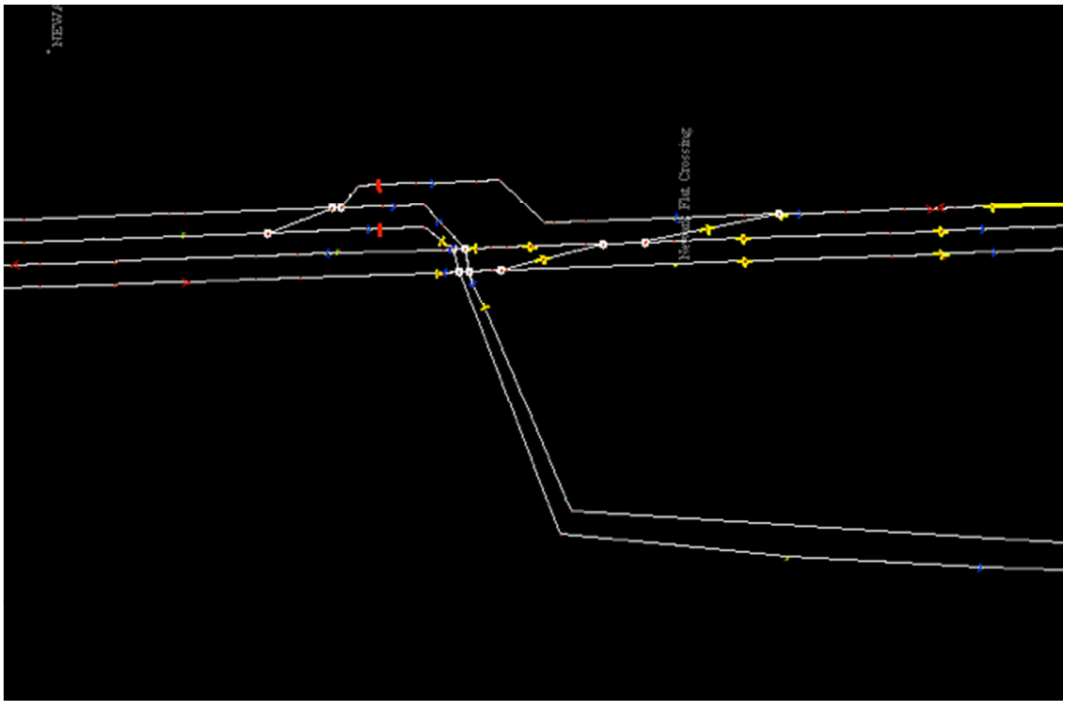

Figure 8: **Newark station (Left part)**

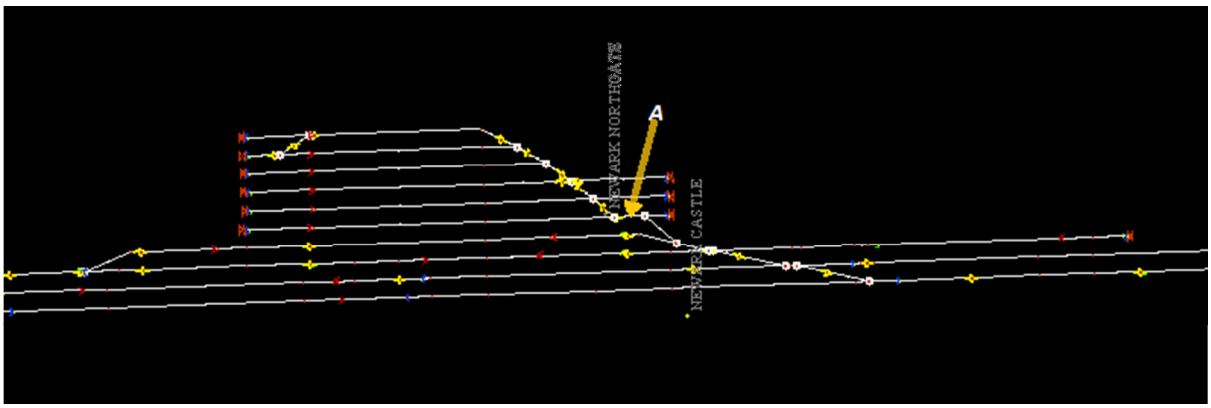

Figure 9: **Newark station (right part)**

The left part went through the FDR3 tool without a problem. But the right part, with 15 points and 23 signals, still causes state explosion. To make the situation worse, it seems that the right part cannot profit from the covering technique. That is, it is hard to be decomposed it into two largely mutually-disjoint parts!

Therefore, we devise a new technique to support further decomposition, which is called **route decomposition**. Basically the route decomposition allows us to add a signal at location **A** in the right part (c.f. Figure 9), which allows all routes passing location A to be decomposed into two routes.

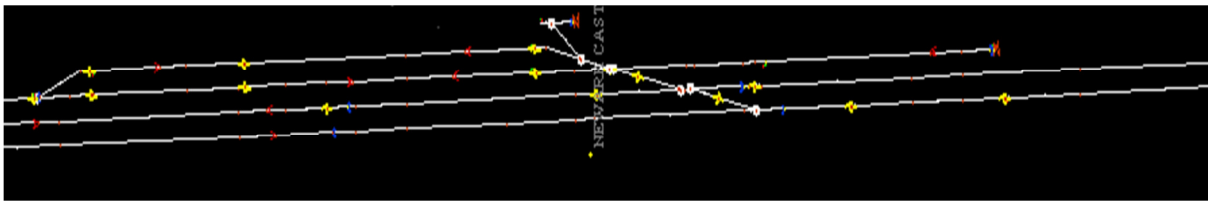

Figure 10: **Newark station (right-bottom part)**

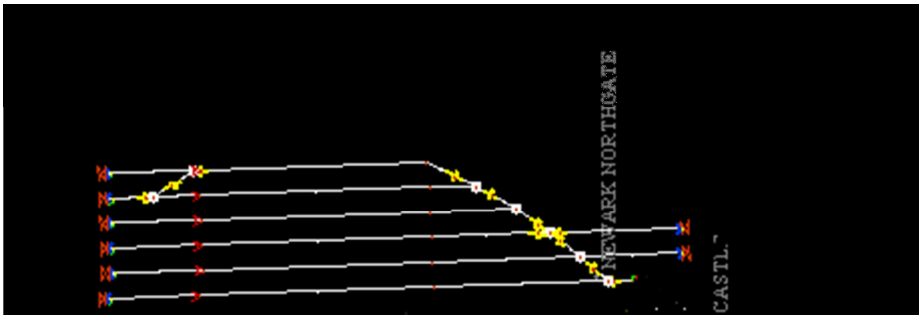

Figure 11: **Newark station (right-top part)**

With the new, virtual, signal installed, we can use the covering technique to further decompose the right part into two subparts: right-top and right-bottom (c.f. Figure 10 and 11), which are both small enough to be verified by FDR3 (c.f. Table below).

Newark:left	Points	Signals	Routes	Time	State	Result
Collision free	7	11	22	53.73s	49,542,286	Pass
Derailment free	7	11	22	49.97s	38,680,050	Pass
Runthrough free	7	11	22	48.62s	38,680,058	Pass

Newark: right-bottom	Points	Signals	Routes	Time	State	Result
Collision free	8	15	32	325.36s	234,981,036	Pass
Derailment free	8	15	32	287.67s	186,312,736	Pass
Runthrough free	8	15	32	278.52s	186,312,736	Pass

Newark: right-top	Points	Signals	Routes	Time	State	Result
Collision free	7	8	9	<1s	3069	Pass
Derailment free	7	8	9	<1s	2813	Pass
Runthrough free	7	8	9	<1s	2813	Pass

The correctness of the route decomposition technique is established by a simulation relation from the new model to the old model since 1) the setting of a long route in the old model can be simulated by the setting of two short routes in the new model and 2) the addition of a green signal in the middle of a long route which is already set will not change the behaviour of trains running on the route.

3.3.3 Comparison of the results

Comparing the verification with CSP||B and the verification with CSP, it is obvious that CSP is much faster, often by two orders of magnitude. Therefore, it is not surprising that CSP verification can deal with larger stations than CSP||B verification. However, CSP verification also requires the application of covering, when it comes to Grantham station. Furthermore, Newark poses a challenge even beyond covering: an additional decomposition technique is needed. Note here, that it is not simply the size of the nodes that determines the size of the sub-scheme plans; rather, the essential criterion is the layout of the routes and how they are entangled with each other.

It comes as a surprise to see CSP outperform CSP||B to such extent. We see three factors contributing to this.

- *Modelling*: based on our experience with CSP||B railway models, in 2015 we made a fresh start with CSP modelling. Here, the long experience with railway modelling allowed us to take a systematic approach, building a minimal model that includes only the really necessary elements, and taking into account a number of well-established modelling ideas.

- *Technology*: The underlying model-checkers ProB (for CSP||B) and FDR3 (for CSP) are tools from different generations. While ProB was built in the early 2000s and has stayed constant ever since with regards to the model checking algorithm, FDR3 is a recent, 2013 re-implementation of the model-checker FDR2. This re-implementation made systematic use of algorithmic advances over the last decade. It is future work to measure the influence of technology by running the ProB model checker on our new CSP models.

- *Granularity*: The CSP||B model is monolithic in the part that represents the interlocking. In contrast to this, the CSP model consists of a large number of small processes. The latter granularity is what makes the CSP model checking faster.

Overall, these verification results demonstrate the need for further experimentation and research in order to come up with a method that can cope with more complex rail nodes.

4 CUI Graphs with suitable timetables

[Communicated by J Armstrong, University of Southampton]

Calculation of Capacity Utilisation Indices (CUIs) for the nodes and links comprising the study area requires the assignment, or mapping, of timetabled trains onto the study area infrastructure, and then 'compressing' the timetable for individual nodes and links so that train movements are separated in time by the minimum applicable headways (for links) or minimum junction margins, platform reoccupation times (or headways) or turnaround times (for nodes, i.e. junctions and station platforms).

The assignment of trains to the study area network is performed by reading electronic timetable in CIF (Common Interface File) or similar format, identifying and extracting trains that serve/use the Timing Point Locations (TIPLOCs) included in the study area during the modelled day(s) and time period(s). The fundamental link between the infrastructure model and the timetable data is thus the set of station and junction TIPLOCs included in the modelled area. Most stations and many junctions have multiple platforms or tracks with distinct identifiers, as used in CIF data, each of which should be included in the infrastructure model for assignment purposes. For example, Peterborough station (TIPLOC: PBRO) includes platforms 1-7 and the non-platform Down Fast (DF) line and Two-way Goods Line (GL), thus requiring nine distinct nodes to which train movements can be assigned. Similarly, at Stoke Junction (TIPLOC: SOKEJN), north of Peterborough, the single southbound line through Stoke Tunnel splits into two lines, slow (SL) and fast (FL), and southbound trains passing this location must be assigned to one line or the other.

We consider network models at various levels of abstraction. The macro-level model comprises the TIPLOCs included in the modelled area and their intermediate links, including information (at most) on the numbers of tracks between them (as used in the MCNDP, for example); the meso-level model includes the TIPLOCs, individual station platforms and numbers, and the IDs of the intermediate tracks (SL, FL for example); and the micro-level model includes all the intermediate switches and crossings and the individual track segments between them, for detailed routeing and capacity utilisation calculation purposes. The macro-level, TIPLOC-based model is summarised for the Retford-Huntingdon section of the ECML in Figure 12 below, with platform numbers and track IDs excluded (note: the platform and track labels used in the infrastructure model must match those used in the CIF data to enable the correct assignment of trains to the infrastructure).

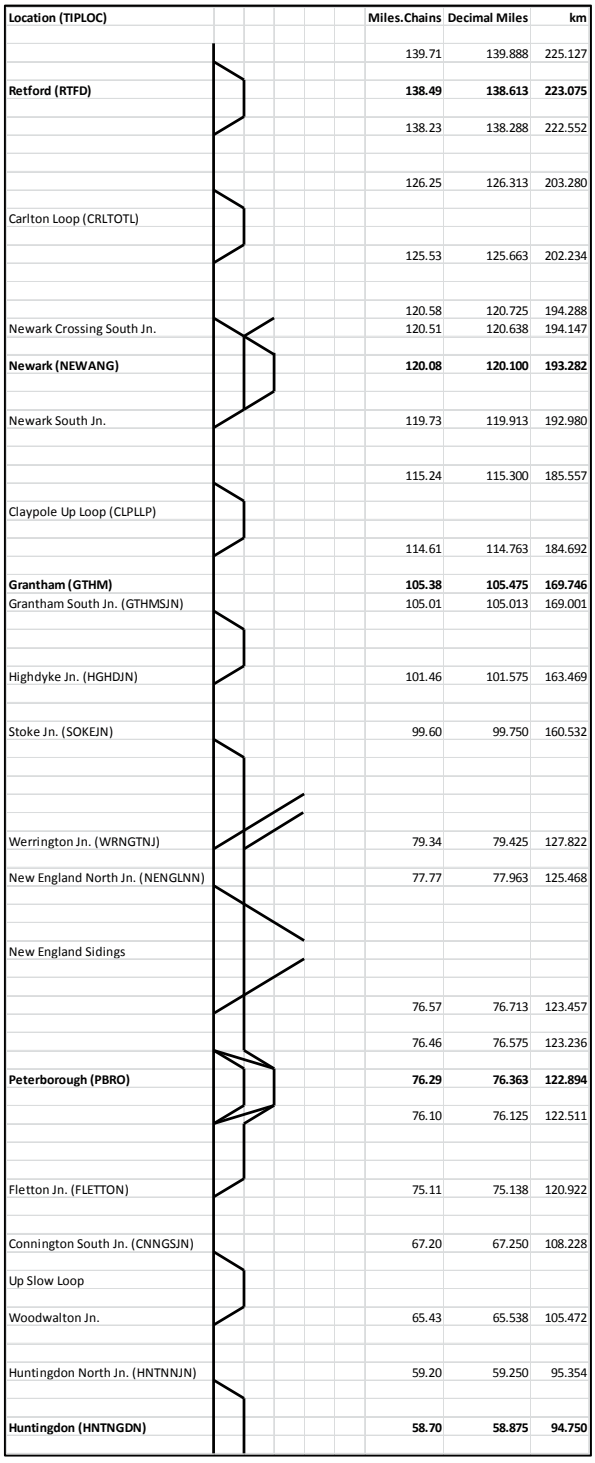

Figure 12: **The Retford-Huntingdon area model**

In addition to the meso-level representation of TIPLOCs and the intermediate tracks between them, the micro-level network model of intermediate switches and crossings (including labels), and the tracks between them (including their lengths), is required. The

detailed routings of trains through these individual nodes and links is determined, and the timings at the intermediate nodes between TIPLOCs are interpolated on the basis of the specified times at TIPLOCs and the intermediate link lengths between them. The switches in the network (but not the crossings) are labelled in the industry-standard Five-Mile Diagrams. These labels were used in the OCCASION project and in the initial DITTO analysis; however, since they are not referred to in the CIF or other timetable data, any agreed set of labels can be used. An extract of the detailed node-link data for the model area is shown in Figure 13.

From	To	Length (km)	CUI Link?
1316	SOKEJN_SL	0.007	TRUE
SOKEJN_SL	1297B	23.629	TRUE
1297B	1297A	0.15	TRUE
1297B	1294A	0.365	TRUE
1294A	TALNGTN_SL	0.166	TRUE
TALNGTN_SL	1273	8.358	TRUE
1273	1272B	0.016	TRUE
1272B	1273	0.016	TRUE
1272B	1274	0.15	TRUE
1274	1272B	0.15	TRUE
1274	WRNGTNJ_SL	0.046	TRUE
WRNGTNJ_SL	1274	0.046	TRUE
WRNGTNJ_SL	1264A	2.358	TRUE
1264A	WRNGTNJ_SL	2.358	TRUE
1264A	1264B	0.073	TRUE
1264A	1262B	0.01	TRUE
1262B	1264A	0.01	TRUE
1262B	NENGLNN_SL	0.006	TRUE
NENGLNN_SL	1262B	0.005	TRUE
1262B	NENGLNN_GL	0.006	TRUE
NENGLNN_GL	1262B	0.005	TRUE
NENGLNN_SL	PBRO78	1.772	TRUE
PBRO78	NENGLNN_SL	1.772	TRUE
PBRO78	1245A	0.2	TRUE
1245A	PBRO78	0.2	TRUE
PBRO78	1262B	1.777	TRUE
1245A	1244B	0.02	TRUE
1244B	1245A	0.02	TRUE
1244B	1244A	0.08	TRUE
1244A	1244B	0.08	TRUE
1244B	1242A	0.382	TRUE
1242A	1244B	0.382	TRUE
1242A	1228A	0.08	TRUE
1228A	1242A	0.08	TRUE
1228A	PBRO_1B	0.021	TRUE
PBRO_1B	1228A	0.021	TRUE
PBRO_1B	PBRO_1	0.08	TRUE
PBRO_1	PBRO_1B	0.08	TRUE
PBRO_1	PBRO_1A	0.08	TRUE
PBRO_1A	PBRO_1	0.08	TRUE
1316	SOKEJN_FL	0.007	TRUE

Figure 13: Example Node-link Data

The data shown represents the southbound Slow Line from the facing switch (1316) at Stoke Junction to Platform 1A at Peterborough. (Platform 1 is divided into areas A and B for short trains, while the whole of the platform is used for longer formations.) It can be seen that the line allows two-way operation between Werrington Junction (TIPLOC: WRNGTNJ) and Peterborough Platform 1. The last line of data shows the link from

switch 1316 to the Fast Line at Stoke Junction, which leads in turn to platforms 2 and 3 at Peterborough.

The headways, junction margins and platform reoccupation and turnaround times used for the timetable compression process are obtained from the Operational Rules of Network Rail's Timetable Planning Rules [NR].

Initial work on the relationship between capacity utilisation and congestion-related reactionary delay (CRRD) for the agreed interface demonstration model area was conducted for southbound operations through Grantham, the central location of the demonstration model. CUI values were calculated for three-hour 'time buckets' for several platform and switch nodes in the Grantham station area, and were plotted against total southbound delay minutes recorded for the same time buckets for the May 2015 timetable (i.e. the timetable in use on Wednesday 4th November 2015, the chosen 'representative operational day' for the exercise). The three-hour time buckets start from 01:00, and thus include the standard morning and evening peak periods of 07:00 – 10:00 and 16:00 – 19:00, respectively. The Grantham station area is shown in Figure 14 below.

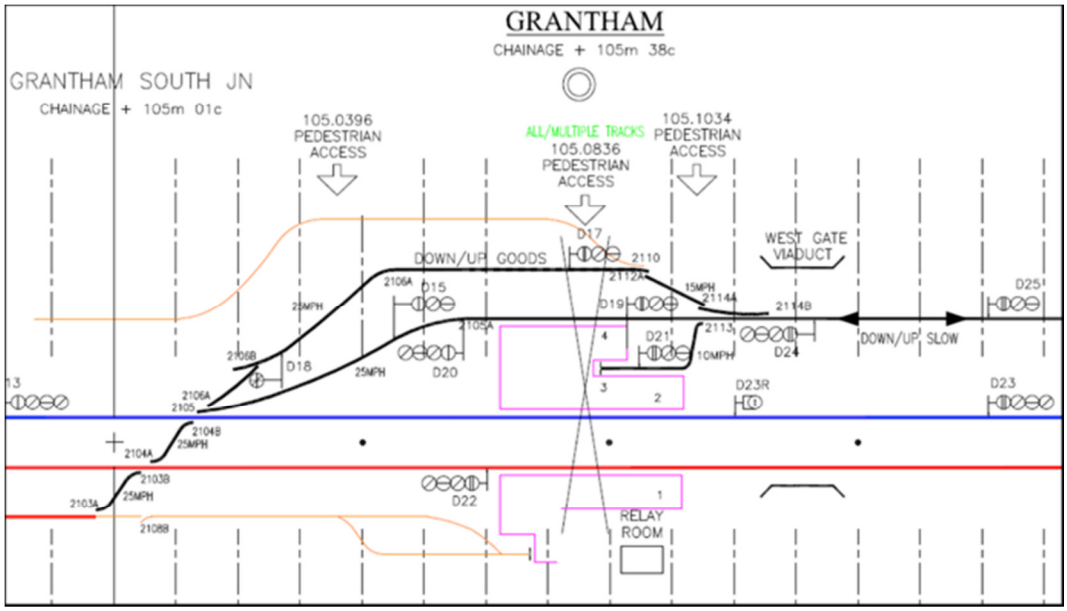

Figure 14: Grantham Station

The relationship between CUI at Platform 1 and total southbound delay minutes is shown in Figure 15 below. The relationship has quite a high level of correlation, with an R^2 value of 82%, and delay minutes increasing sharply as CUI approaches 60%.

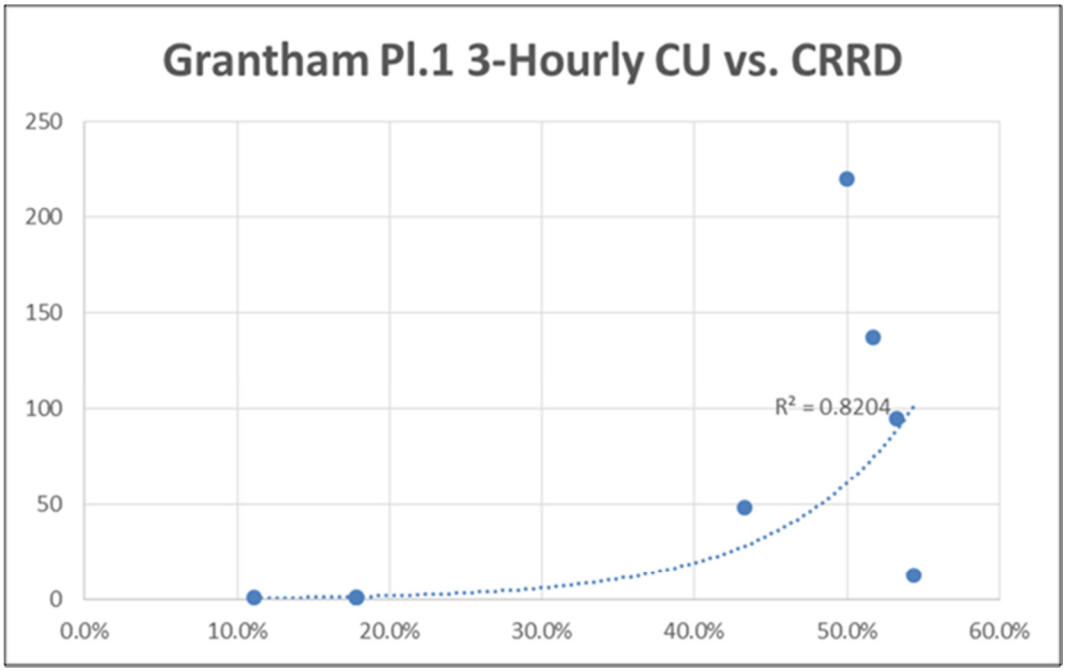

Figure 15: **CUI vs delay for Platform 1**

The equivalent results for Platform 4 are shown in Figure 16 below. This has an improved level of correlation, but this is perhaps somewhat misleading, since it can be seen that the CUI values for Platform 4 are very low, with maximum values of 10%.

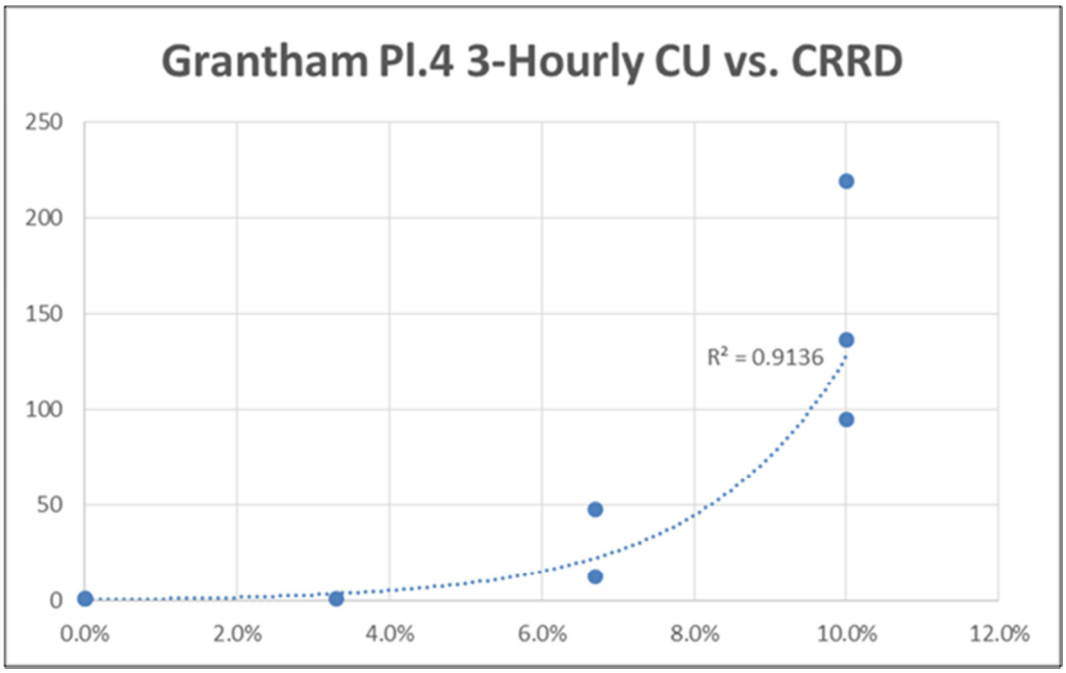

Figure 16: **CUI vs delay for Platform 4**

The low CUI values for Platform 4 are unlikely to be directly governing the overall levels of CRRD at the station; however, it may be that high CUI values for Platform 4 coincide with high values elsewhere in the station, and that delays reflect the combination of these. This hypothesis is supported by the CUI:CRRD results for the converging southbound switch 2104A (see station layout diagram above), which carries traffic from both Platforms 1 and 4, as shown in Figure 17 below.

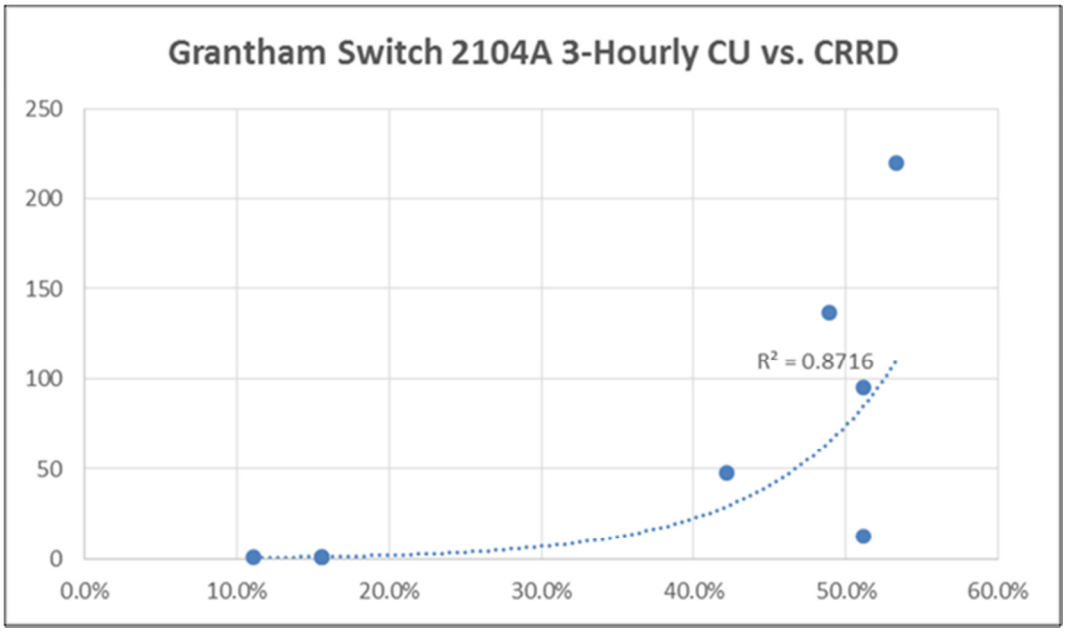

Figure 17: CUI vs delay for Switch 2104A

The R^2 value of 87% is slightly less than that for Platform 4 alone, but considerably higher than that for Platform 1, and the capacity utilisation values for which CRRD begins to increase sharply are similar to those for Platform 1, i.e. 50%-60%. Further investigations are required to examine these relationships and identify suitable upper limits for capacity utilisation; these will include the disaggregation of station delays by individual platform and switch, where possible, and analysis of the trade-off between additional service and additional delays [PKA15].

The relationship between capacity utilisation and delay, and suitable capacity utilisation upper limits are still under investigation. It was originally intended to include the capacity utilisation calculations as part of the iterative timetable optimisation process, and there remains an aspiration to do this. However, the optimisation calculations are already very time-consuming, even on the Southampton supercomputer, and it has been suggested that it may be more practical to apply the capacity utilisation calculations to a range of optimisation outputs to retrospectively identify a 'frontier' where the benefits of adding services outweigh the performance (i.e. delay)-related handicaps. As things stand, the infrastructure model used in the timetable optimisation

process is less detailed than the one used for the capacity utilisation calculations (meso- vs. micro-level), and combining the two processes would need to take account of this.

5 Train timetabling and scheduling under uncertainty

[Communicated by J Preston and A Kovacs, University of Southampton]

A viable approach to keeping up with increasing numbers of railway passengers is to run more services at peak times; that is, add more services to the timetable. However, more traffic means more conflicts amongst trains; the tighter the capacity constraints, the more conflicts. Without sufficient buffer times to absorb uncertain delays, the delay of one train might propagate over the entire network.

Given this, we address a realistic timetabling problem by considering the number of services offered along with their reliability. A two-stage stochastic programming model has been developed for generating timetables with the required number of services at the tactical level. Different recourse actions to recover from delays are taken into account at the operational level (e.g., speeding up trains). The model considers conflicts among different types of trains (e.g., express and freight trains) at different locations (e.g. points, junctions, and platforms).

Small instances can be solved by commercial solvers; however, for solving large instances, we developed a large neighbourhood search algorithm (LNS). In each iteration, the algorithm executes two phases: in the first phase, a feasible order among trains is determined; given this order, the reliability of the timetable is optimised in the second phase.

Train services are scheduled by a recursive algorithm that is guaranteed to insert a service into a given timetable if a feasible insertion position exists. Appropriate buffer times are incorporated into the timetable by a greedy algorithm and linear programming in order to absorb uncertain delays.

More complicated recourse actions have been tested which include changing the platform assignments if a platform is blocked, and allowing trains to overtake if an express train is stuck after a regular train. However, our results suggest that considering complicated recourse actions can be avoided in the timetabling phase. This result remains to be verified on railway systems with large-scale delays.

The LNS has been tested extensively on benchmark instances. The results show that the algorithm is able to generate feasible timetables even when capacity constraints are tight. The solution quality increases with a larger number of iterations. The generated results are on average 6.6% worse than the best known solution; the average computation time is 4.1 hours.

The results of a case study indicate that there is plenty of room for increasing the operational capacity at Peterborough: 40 additional services could be inserted into the timetable. As the availability of rolling stock and staff, as well as shunting movements within the stations, have not been considered here, the results should be interpreted as a best case situation. Nevertheless, they suggest that it is possible to increase the capacity utilisation of the existing infrastructure by using state-of-the-art optimisation techniques, as opposed to alternative strategies that are significantly more expensive and involve reducing headway times (e.g., by updating the signalling system and improving the breaking performance of trains) or laying new tracks. However, each additional service would lead to a decrease in reliability by around 3.6%. Some further details are provided below.

5.1 Timetable optimization graphs with suitable timetables

For our timetable optimization modelling, our model consists of a network layout, a set of trains, and a set of delay scenarios. An example of a network layout is given in Figure 18. We consider several stations with different numbers of platforms, points, junctions, double track lines, quadruple track lines with fast and slow tracks in each direction and single track lines that are traversed in both directions. Grantham is at the northern boundary of the modelled area.

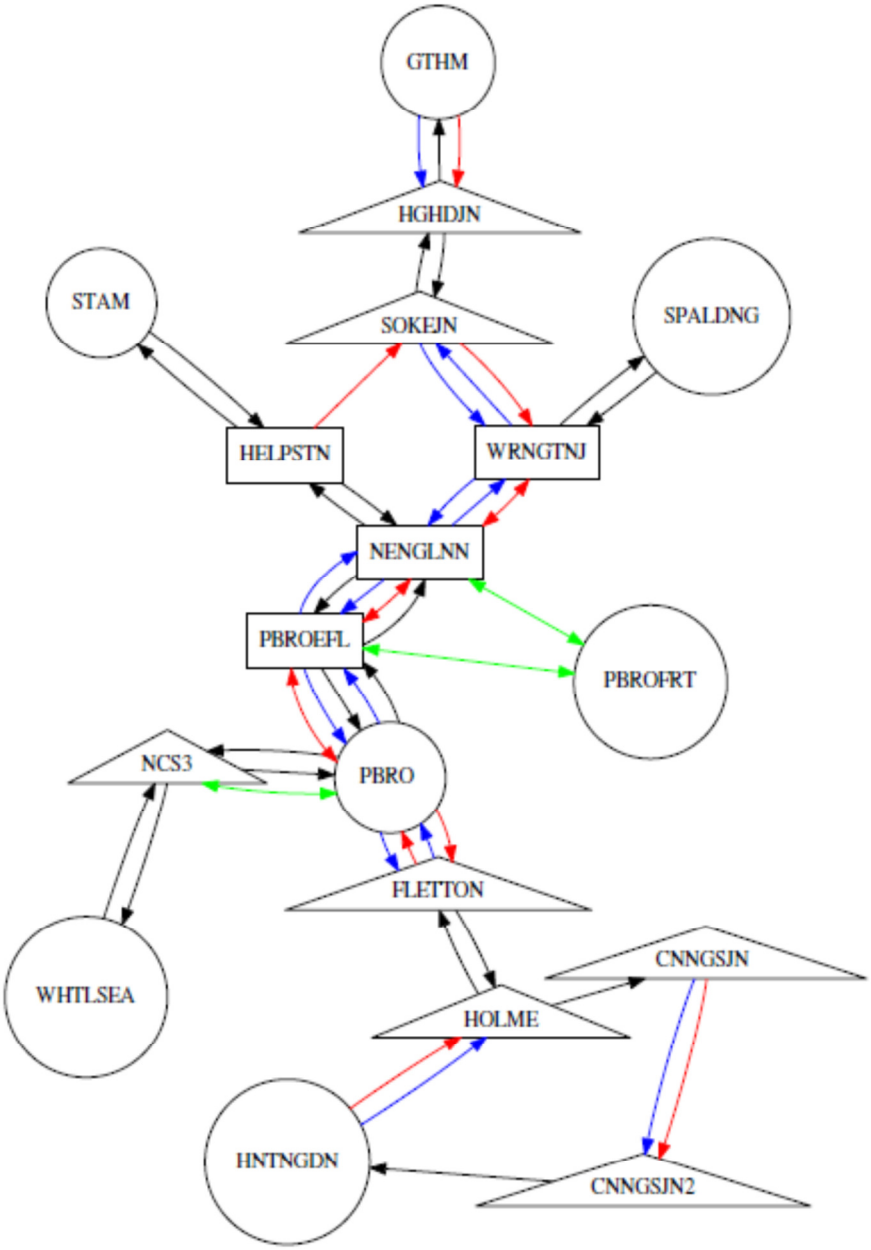

Figure 18: **Example of a network layout.**

In our code, the network layout is presented as follows:

stations: 6
switches: 2
crossings: 1
arcs: 21
PLATFORMS
stationID #platforms platforms&direction(U..up, D..down, B..both)

0	4	1 U 2 U 3 D 4 D
1	4	1 U 2 U 3 D 4 D
2	2	1 U 2 D
3	4	1 U 2 U 3 D 4 D
4	2	1 U 2 D
5	2	1 U 2 D

ARCS

from	to	distance (km)	directions(1,2)	type(F..fast,S..slow,M..main)):
0	1	40	1	F
0	1	40	1	S
1	0	40	1	F
1	0	40	1	S
0	6	52	1	M
6	0	52	1	M
6	2	57	1	F
6	2	57	1	S
2	6	57	1	F
2	6	57	1	S
0	7	46	1	M
7	0	46	1	M
7	3	42	1	F
7	3	42	1	S
3	7	42	1	F
3	7	42	1	S
0	8	13	1	M
8	0	13	1	M
8	5	39	1	M
5	8	39	1	M
4	8	8	2	M

JUNCTIONS

crossingID 8 #linesInvolved 2

line	#conflicts	conflicting lines
0 4	1	5 0
5 0	1	0 4

The set of trains travelling along the network is provided in the form of a timetable. For each train, we are given the route (i.e., a sequence of stations, junction, and points); preferred arrival and departure times at different locations; and the type of train (e.g., freight, express, or regular). Furthermore, a delay scenarios list is provided, where for each train, the duration and location of the delay is specified.

Our case study focuses at the rail network surrounding Peterborough station. The layout involves seven stations, four junctions, and seven points. The network comprises of 47 arcs, each arc representing a track segment that can either be a fast, slow, main, or freight track. Freight trains can be assigned to slow, main, and freight tracks; regular trains to slow and main tracks; and express trains to fast, slow, and main tracks.

The set of trains is selected from a representative weekday (4/11/2015). From the national timetable, we select all passenger and freight trains (including empty locomotives) that visit Peterborough between 7am and 9am. In total, we consider 55 services in the reference timetable. The average speed of express, regular, and freight trains is assumed to be 125, 100, and 75mph, respectively. The time required for acceleration and deceleration is considered by decreasing the average speed by 7% if a given train has to stop once in our model, by 14% with two stops, and by 21% with at least three stops.

Delay information is gathered from historical delay data provided by Network Rail. More than 6 million delays were recorded between 1/12/2013 and 18/04/2015 (i.e., over 503 days). After filtering out irrelevant information[1], almost 800,000 records remain.

In a second step, we match filtered trains (T) with trains in the delay data (D). There is no unique identifier that unambiguously links trains in the two sources of data. Therefore, we apply the following strategy: Take the set of relevant trains, the delay data, and a time margin TM. Match T with D if: (i) T is a passenger train (delays of freight trains are not recorded); (ii) T and D have the same origin and destination; and (iii) T departs within the departure time of D +/- TM.

Delay scenarios are sampled in a Monte-Carlo fashion. In each scenario, and for each train, we decide by Bernoulli trial whether or not it is delayed; if yes, we associate the location and duration of the delay. The length of the delay is modelled by a Gamma distribution.

The results of this work will be reported in detail in milestone 9 (due September 2016) but we have established that an additional 40 trains in the morning peak hour at Peterborough is feasible, although not necessarily desirable. This preliminary result would represent an increase in service of around 73% but an increase in an index of

[1]Primary delays are a model input. The efficiency of the model algorithm is measured by its ability to mitigate delay propagation by incorporating proper buffer times into the timetable. The smaller the secondary delays, the better the objective value, and the better the solution.

delays of around 144%. Of these 40 additional trains, 18 will run to/from Grantham. Grantham is modelled as having 46 trains in the morning peak, so this would represent an increase in service of 30%.

6 Rail network simulation

[Communicated by R Liu and H Ye, University of Leeds]

TrackULA is a rail network simulation model adapted from its sister road simulation model DRACULA (for Dynamic Route Assignment Combining User Learning and microsimulation) developed by the team from University of Leeds. The data format for TrackULA follows that of DRACULA and this format is summarised below. For a full description, please consult the DRACULA User Manual [Dra].

6.1 Network Representation

The railway network in TrackULA is represented as a directed graph. A directed graph is a set of nodes joined by a set of directional links, where:

- A node in our railway network is a station, terminal, junction, signal point, or an external node used to connect to the network outside the study area. A node is specified by its location (in terms its x- and y-coordinates relative to a reference point in a network), the number of lines (tracks) it is connected to, and its type (e.g. station, terminus, junction etc.). For a fixed-block system, a signal point (i.e. the start and end of a block) is represented as a node. In a moving-block system, a node tends to be a station, terminus, or a junction.
- A link is a directional track segment between two nodes. By default, a link is unidirectional, i.e. a track for a single direction use. (There is a possibility to extend a road network feature of DRACULA to model the bidirectional tracks in TrackULA.) A (unidirectional) link is specified by its upstream and downstream nodes, link length, speed limit, and turns permitted to other outbound links from the downstream node.

6.2 Vehicle Characteristics

Trains are individually represented, with each having a set of individual characteristics including:

- train type: e.g. high-speed passenger train, low-speed passenger train, freight train ;
- vehicle length: the physical length of the vehicle (as trains in TrackCULA is considered as a rigid body)
- minimum distance headway: a minimum stopping distance to the train or an obstacle in front

- maximum acceleration
- maximum deceleration
- desired cruise speed (relative to the maximum speed limit on any individual track)

6.3 Train Timetable

The network and vehicle characteristics can be easily translated from a road to a rail setting but a train timetable also needs to be described in terms of routes/paths through the network. In TrackULA, the timetable contains three sections of records:

1. Record type 1 – Service description: the name of the route (e.g. ECML), type of train (e.g. passenger/freight, fast/slow train), the start time of the first train, and the frequency of the service;
2. Record type 2 – Service route: a list of nodes to traverse through the network;
3. Record type 3 – Stops en-route: a list of stations at which to stop, and an average boarding flow or boarding times.

Stations are normally represented as a node.

For schedule-based services (i.e. most of the train services), it is advised to specify the average boarding time per stopping station. The value of the boarding time can be derived from the scheduled arrival and departure times.

For frequency-based services (e.g. metro lines, or bus services), the boarding flow is often specified. In this case, the train/bus dwell time is derived from the following function:

$$T = D + A + b_1(1 - p_s)N + b_2 p_s N$$

where T is the boarding time, D the door opening and closing time, A the average alighting time, N the number of boarding passengers, p_s is a proportion presenting faster boarding passengers (e.g. in the case of buses, the passengers who hold a pre-paid ticket, i.e. Oyster card; while in the case of trains/metro, passengers without luggage/ experienced commuters), and b1 and b2 are the times it takes for the faster and the slower boarding passengers to board.

In this version of TrackULA, we do not model the capacity constraints on train carriages, nor the alighting passenger flows. Thus the time it takes for passenger to alight a train is modelled as an average value A.

6.4 Simulation Outputs

The default outputs are aggregated measures of:

- Total vehicle-hours travelled;
- Total vehicle-kms travelled;
- Average travel time;
- Average speed;
- Fuel consumption; and
- Emissions from pollutants CO, NOx and HC.

The measures are aggregated over a user-specified time period (e.g. every 10minutes). Spatially, the measures are aggregated for each route and road link as well as for the whole network.

At the user's request, the program may also output individual vehicles' second-by-second locations and speeds to provide space-time trajectories of the vehicles. This produces a very large output file. A graphical animation of the vehicles' movements can also be shown in parallel with the simulation, giving the user a direct view of the traffic conditions on the network.

For modelling of public transport (e.g. trains), additional outputs are provided to give measures of:

- service operation: journey time of each rail service, dwell times at each rail station etc.;
- passenger wait times at rail stations;
- reliability of rail services over the period modelled; and
- reliability to passengers.

Passenger wait times at a station are derived from the average boarding flows and the headways between the trains.

There are many different measures of public transport reliability. Liu and Sinha [LS07] and Sorratini et al. [SLS08] provide a comprehensive review of reliability measures for buses, from the view point of both the operators and the passengers. The excess wait

time (the difference between actual wait time and the schedule wait time) has been widely used to measure bus passengers' reliability. Liu and Sinha [LS07] showed that the excess wait times are closely correlated to bus headways, implying that the variability in headways is a good measure for passenger reliability. For rail services, a common adopted reliability measure is the percentage of on-time arrivals (at the terminal station).

6.5 An Example Illustration

An example network and its model description are illustrated in Figure 19. This example network has some similarities with the section of the ECML between Newark and Grantham, including the Claypole loops (see Figure 3).

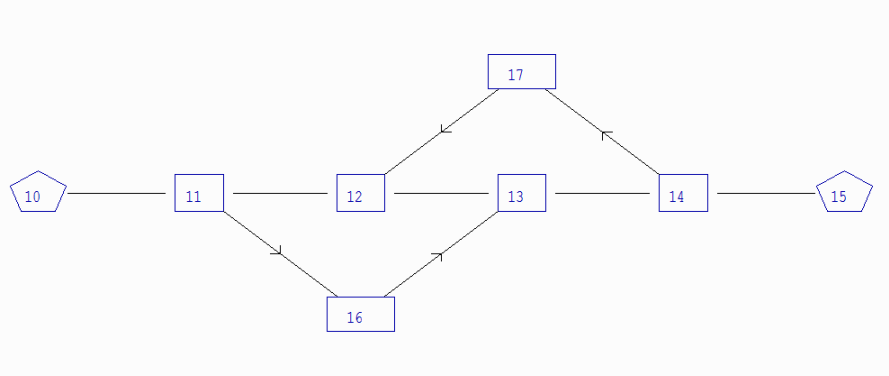

Figure 19: Example network

This network consists of 8 nodes (numbered 10 through 17). The connections along the main corridor (from node 10 to 15) are bi-directional, while the sections through nodes 16 and 17 are one-way.

The coding for one of the nodes, node 12, is described below:

```
12     3     1
       11    1   200 5000     0   0 0 1800   1 1
       17    1   120 4000     0   0 0 1800   1 1
       13    1   200 5000 1800   1 1    0    0 0
```

Line 1: ' 12 3 1' is a node description, specifying that node 12 has three connections, and is of type 1 (a priority junction).

Lines 2-4: is a link description, with each line describing one of the links connected to node 12. For example, line 2 describes that this link comes from node 11, and that it has one single track on the section with speed limit 200 (km/hr), and of length 5000 (m). The following three zeros indicate that this link is prohibited to turn to node 17, while the last three digits indicate the turn from the link to node 13 is permitted.

The full description of the above example network is given below. For further detailed explanation of the network description, please consult the DRACULA User Manual [Dra].

```
10     1     0
      11     1  200 5000
11     3     1
      10     2  200 5000 1800   1 1 1800   2 2
      12     1  200 5000    0   0 0 1800   1 1
      16     0
12     3     1
      11     1  200 5000    0   0 0 1800   1 1
      17     1  120 4000    0   0 0 1800   1 1
      13     1  200 5000 1800   1 1    0   0 0
13     3     1
      12     1  200 5000 1800   1 1    0   0 0
      14     1  200 5000    0   0 0 1800   1 1
      16     1  120 4000    0   0 0 1800   1 1
14     3     1
      13     1  200 5000    0   0 0 1800   1 1
      17     0
      15     2  200 5000 1800   1 1 1800   2 2
15     1     0
      14     1  200 5000
16     2     1
      11     1  200 5000 1800   1 1
      13     0
17     2     1
      14     1  200 4000 1800   1 1
      12     0
```

7 Tool Integration at work: BRaVE and OnTrack

[Communicated by P James, X Wang, F Moller and M Roggenbach, Swansea University;
L Chen, D Kirkwood and G L Nicholson, University of Birmingham;
and HN Nguyen, Coventry University]

As an example of tool integration, we briefly give an account of how the tools BRaVE and OnTrack complement each other.

7.1 An example of the capabilities of BRaVE

As an example of the capabilities of BRaVE, BRaVe has been used to simulate a timetable running on the East Coast Main Line during the morning peak period, 7am – 10am. The section of network used for the simulation consists of the southern part of the ECML between London King's Cross and Doncaster, and the Hertford loop line, the Cambridge line, and the Peterborough to Lincoln line. In this example, a simulation was first conducted with traffic flowing as timetabled, followed by a further simulation in which multiple entrance delays in the range 30 seconds – 300 seconds to all trains entering during the first 15 minutes of the simulation were generated.

This experiment permits the examination of the effects of perturbed running at either or both of a network-wide macroscopic or more detailed microscopic level. The macroscopic approach relies on event-based collection of simulation data: station, timing point and block section arrival and departure times. From this, a view of the severity and propagation of delay can be obtained, alongside a dynamic assessment of the capacity consumed and robustness of a timetable. Measurements of delay per service or per station can be made, and then manipulated to give the PPM value and a record of the delay distributions across a network.

In ideal traffic flow conditions, the macroscopic event-based data allows the quantification of timetabled capacity consumption using, for example, the CUI, UIC405 and UIC406 measures. Further, the energy consumed at the wheel is calculated and recorded.

Figure 20 shows the arrival delays to services arriving at Grantham station during the simulation period. Such analysis may lead to the requirement to examine particular services in more detail at a microscopic level. BRaVE permits the analysis of this kind to any subset of services running during the simulation. Figure 21 shows the gradient profile, speed profile and running diagram for an individual service. BRaVE runs in a time-driven manner and for microscopic analysis, observations are made at fixed time intervals, the standard value being every 1 second, as was the case for this example.

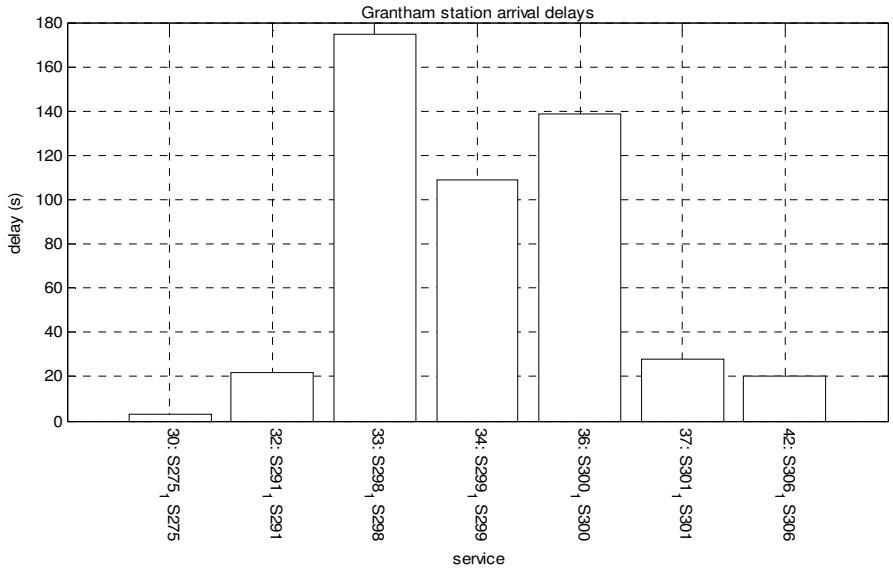

Figure 20: **Arrival delays to services at Grantham station.**

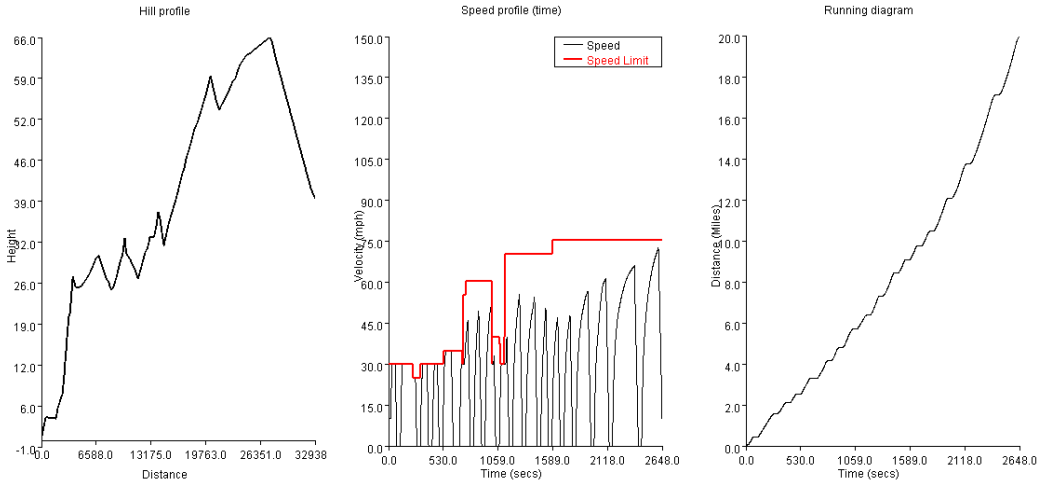

Figure 21: **The gradient profile (left), speed profile (centre) and running diagram (right) for service S291 during perturbed running.**

7.2 Simulation in BRaVE

In order to test the integration of OnTrack and BRaVE, an illustrative simulation is set up in BRaVE in which three trains run along a short section of the East Coast mainline between Barkston South (just north of Grantham) and Werrington Junctions (just north

of Peterborough). Two trains (S1 and S3) travel exclusively along the mainline, while the third train (S2) enters the mainline at Barkston South Junction and travels towards Werrington Junction.

All three trains travel in the same direction, in the order S1, followed by S2, followed by S3. This means that train S2 enters the mainline between trains S1 and S3. A screenshot of the simulation in progress is given in Figure 22. The corresponding block occupation diagram for the mainline is shown in Figure 23. It can be seen that S2 enters part way along the mainline, occupying the ECML from the second block onwards.

The simulation is designed to replicate stressed traffic conditions, equivalent to the case where the first train is running late along the section, impeding the free flow of S2 and S3. In this case, it is the job of the signalling system to safely separate the trains; the correct design of the interlocking rules results in safe separation of the trains, i.e., no blocks overlap in Figure 23. Block section numbers 3 and 4 are the critical sections, in which the blocks of S1, S2 and S3 are contiguous.

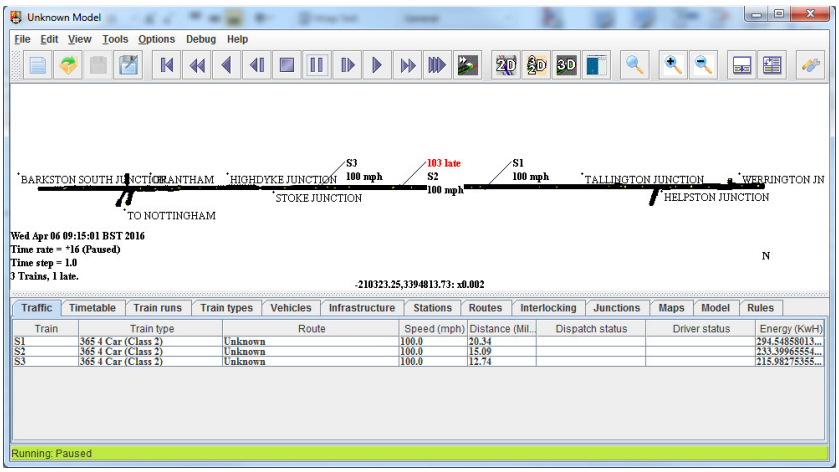

Figure 22: **Running simulation between Barkston South and Werrington Junction**

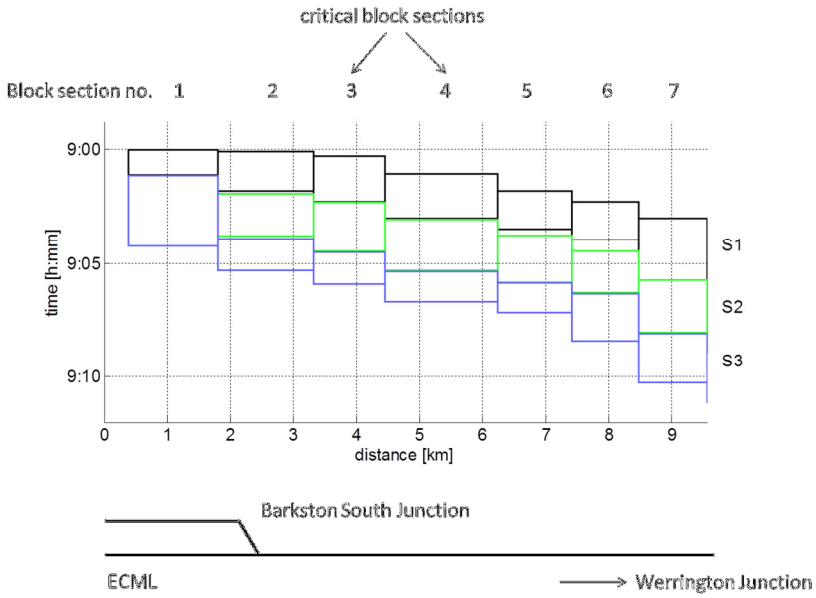

Figure 23: **Blocking model Barkston to Werrington**

In order to demonstrate an unsafe condition, the simulation is repeated but a modification is made to the signal script of the signal protecting Barkston South Junction (see Figure 24). The interlocking rule for this signal was previously set to protect the junction at Barkston South by clearing when a route is set across the junction area. The signal script is modified to protect the track circuit in front of the signal.

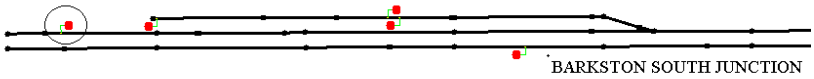

Figure 24: **Corrupted signal**

The simulation is then repeated with otherwise the same conditions for the three trains travelling from Barkston South to Werrington. BRaVE performs error checking on the signal script but as the script is still valid, no error is reported. Its blocking model is shown in Figure 25.

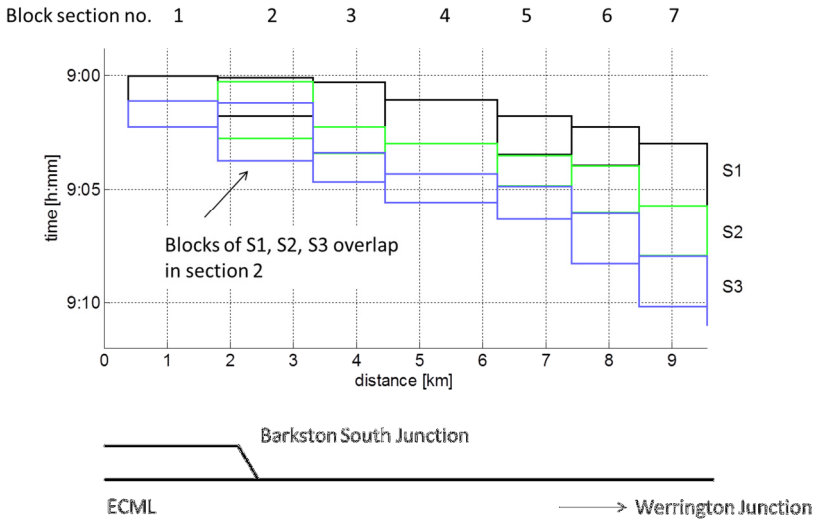

Figure 25: Blocking model of unsafe interlocking rules

In this case there is an overlap of blocks when train S2 enters the ECML at Barkston South Junction (in block section 2). This is confirmed by checking the graphical model in the BRaVE console (Figure 26). Further, the interlocking rule does not prevent S3 from entering block section 2 before that block is released by S2, a second unsafe condition.

Figure 26: Train collision

It is possible in BRaVE to run simulations where trains pass over each other, either in the same direction or in opposite head-on-collision directions. Clearly this is not acceptable in the real world.

7.3 Model checking in OnTrack

After the illustrative models of Barkston South Junction and Werrington have been converted from the BRaVE format into OnTrack DSL format, OnTrack can be used to generate its CSP||B model, ready for automatic verification by ProB.

Table 6 summarises the experimental results. Each row in the table shows the size for a sub-scheme plan in terms of numbers of track circuits (#TC), points (#Pt), signals (#Sn)

and routes (#Rt). It also highlights the model checking result including running time and number of states (#states) of the corresponding CSP||B model, and whether the three safety properties are satisfied. Thanks to covering theory developed in [JMN+14b], safety results for each of these CSP||B sub-models can be combined into the safety result for the whole Barkston South section.

No.	#TC	#Pt	#Sn	#Rt	Time	#State	Safety
1	13	4	3	5	2h09m	28344	true
2	13	4	3	5	2h09m	28344	true
3	13	4	3	5	2h09m	28344	true
4	2	0	1	1	1s	69	true
5	3	0	1	1	1s	83	true
6	2	0	1	2	2s	101	true
7	13	4	3	5	2h09m	28344	true
8	13	4	3	5	2h09m	28344	true
9	8	4	1	2	4m00s	2396	true
10	16	4	4	6	3h59m	52526	true
11	13	4	3	5	2h09m	28344	true
12	16	4	4	6	4h41m	52526	true
13	6	2	1	2	24s	521	true
14	16	4	4	6	3h58m	52526	true
15	7	4	1	2	3m19s	2184	true
16	16	4	4	6	4h41m	52526	true

Table 6: **Verification results**

In addition, model checking the Barkston South example with the corrupted signal (see Figure 24) as mentioned in the above simulation also confirms that the model is unsafe. In particular, ProB provides a counter example when checking the sub- scheme plan 10 responsible for the safety of the point N10807 where a train collision occurs as depicted in Figure 26. Rather than a collision, the first unsafe situation found by the model checker is a run-through (see Figure 27). The trace leading to this error is as follows. First a train passes the corrupted signal. As soon as it arrives at the point N10807, the corrupted signal can be set again, allowing the second train to pass it. However, before the second train can reach N10807, the first train is clear of N10807, thereby releasing the lock on the point. Then, the signal on the side-line can be set, resulting in N10807 to be moved to reverse. Finally, a run-through occurs when the second train arrives at the point N10807.

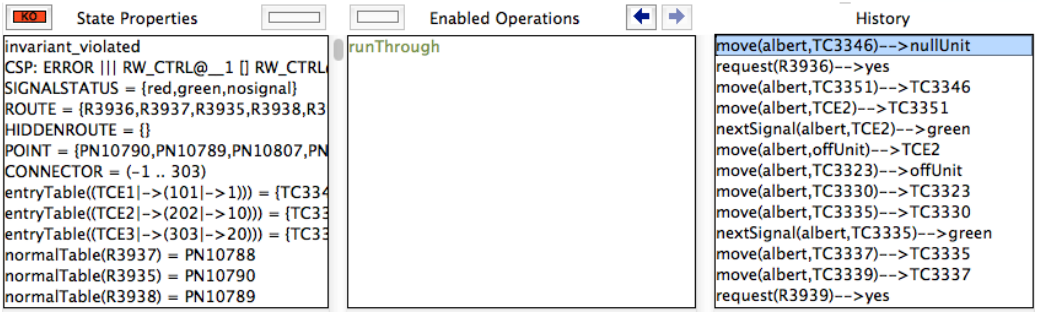

Figure 27: **Counter example trace**

7.4 Reflection

Our integration of BRaVE and OnTrack bridges the gap that occurs from varying details in data sources through automated transformations. This integration provides a first step towards a seamless environment for prototyping, concept development, and safety analysis “under one roof”.

We gave an example where two trains clearly conflict with each other and showed both: simulation data produced by BRaVE and a counter example trace obtained from OnTrack. This demonstrates how the OnTrack tool complements the capability of BRaVE by providing an interlocking design checking function and can be used to give confidence that the model is correct and that the design of the interlocking is valid.

8 Summary

In this report, we presented a number of tools to carry out various analyses on a common case study area chosen from the East Coast Mainline.

The analyses range over a number of different questions:

- Are the rail nodes safe?
- What are the capacity utilisation indices for the nodes and links?
- How can timetabling and scheduling be optimized under uncertainty?
- How can we dynamically optimize a network?

To this end, tools developed in Southampton, Leeds, and Swansea complement each other to provide answers to these questions, where a special focus is on safety assessment.

Additionally, we demonstrate in an inter-project cooperation how simulation and verification can be carried out in an integrated fashion. To this end we report on an experiment carried out on an example from the common case study area which shows how tools from the different projects – namely Brave from Birmingham and OnTrack from Swansea – have been integrated to work with each other.

Overall, we demonstrate how the methods and tools developed within DITTO complement each other and can be applied together for a comprehensive analysis of rail networks.

References

[BASP15] S. P. Blainey, J. Armstrong, A. S. J. Smith and J. M. Preston. New routes on old railways: increasing rail's mode share within the constraints of the existing railway network. Transportation, pages 1–18, 2015.

[Bra] BRaVE. http://www.bravesim.org.

[Dit16] Ditto Project Deliverable 4.1, Milestone 3: Interim Report and Prototype, April 2016.

[Dra] DRACULA User Manual. http://www.its.leeds.ac.uk/software/dracula/

[DWK+16] H. Douglas, P. Weston, D. Kirkwood, S. Hill-Mansen and C. Roberts. Method for validating the train motion equations used for passenger rail vehicle simulation. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 2016.

[IMNR12] Y. Isobe, F. Moller, H. N. Nguyen, and M. Roggenbach. Safety and line capacity in railways – an approach in Timed CSP. In IFM, pages 54–68, 2012.

[JMN+14] P. James, F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider and H. Treharne. Techniques for modelling and verifying railway interlockings. International Journal on Software Tools for Technology Transfer, 2014.

[JTT+13] P. James, M. Trumble, H. Treharne, M. Roggenbach and S. Schneider. OnTrack: An Open Tooling Environment for Railway Verification. In NASA Formal Methods, Lecture Notes in Computer Science. Springer, 2013.

[Kerr01] D. Kerr and T. Rowbotham. Introduction to Railway Signalling. Institute of Railway Signal Engineers, London, 2001.

[LS07] R. Liu and S. Sinha Modelling urban bus service and passenger reliability. Paper presented at the International Symposium on Transportation Network Reliability, The Hague, July 2007.

[NR] Network Rail. Timetable Planning Rules. http://www.networkrail.co.uk/aspx/3741.aspx

[PGB16] D. C. Paraskevopoulos, S. Grel and T. Bekta. The congested multicommodity network design problem. Transportation Research Part E: Logistics and Transportation Review, 85:166 – 187, 2016.

[PKA15] J. Preston, G. Kampantaidis and J. Armstrong The Trade-off between Delays and Capacity Utilisation: Observations from the UK. 6th International Conference on Railway Operations Modelling and Analysis. Tokyo, Japan, 23-26 March, 2015

[RB01] A. Radtke and Bendfeldt, J.-P. Handling of railway operation problems with RailSys. World Congress on Railway Research, Cologne, Germany. http://www.uic.org/cdrom/2001/wcrr2001/pdf/sp/3_4_1/235.pdf

[RML] RML. https://www.railml.org/en/.

[RSSB03] Rail Safety and Standards Board. Railway Group Standard GK/RT0060 (issue four): Interlocking Principles. Rail Safety and Standards Board, London, June 2003.

[SLS08] Sorratini, J., Liu, R. and Sinha, S (2008) Assessing bus transport reliability using micro-simulation. Transport Planning & Technology, 31(3), 303-324.