Visualising Railway Safety Verification

Filippos Pantekis$^{[0000-0001-7817-6450]}$, Phillip James$^{[0000-0002-4307-649X]}$, Liam O'Reilly$^{[0000-0002-4894-2158]}$,
Daniel Archambault$^{[0000-0003-4978-8479]}$, and Faron Moller$^{[0000-0001-9535-8053]}$

Department of Computer Science, Swansea University, Wales, UK
{`filippos.pantekis, p.d.james, l.p.oreilly, d.w.archambault, f.g.moller`}`@swansea.ac.uk`

Abstract. The application of formal methods to the railway domain has a long-standing history within the academic community. Many approaches can provide both successful proofs of safety and, in the case of failure, traces explaining the failure. However, if a given model does produce a failure, it is difficult to understand the conditions that led to the issue. We present a method to visualise railway safety issues to help engineers and researchers explore the problem so that they can adjust their designs accordingly. We evaluate our approach through qualitative real-world case studies with researchers and railway engineers.

1 Introduction

Railway signalling represents an example of safety critical control systems. As such, the use of rigorous development processes using formal methods has been extensively studied by the academic community [15, 6, 14, 11, 24, 19, 13, 16, 20, 27, 1]. Such approaches involve automatically producing a mathematical proof that the control system under consideration obeys certain rules regarding safety. However, uptake of such methods by industry has been hindered by the challenges of: *scalability* (the proposed mathematical proof techniques do not scale to large industrial examples); *faithfulness* (the models created fail to capture the intricacies of modern railway signalling, which are often supplier dependent); and *usability* (existing tools for formal analysis are not necessarily accessible to signalling engineers). In recent years, the formal methods community has proposed solutions to scalability [25] and faithfulness [21]. However accessibility remains an open challenge.

In this paper, we present a visualisation system for understanding safety issues in scheme plans, specifically (1) a method to draw scheme plans that is useful for railway engineers, supporting interoperability between toolsets; and (2) a dynamic visualisation technique to view key frames pertaining to safety issues in context. We evaluate these approaches with railway engineers from Siemens Rail UK and academics working in formal methods. The feedback provides evidence that our algorithm for track layout is a useful way to improve tool interoperability, whilst the evaluation of our visualisation approach for counterexamples suggests that experienced users can quickly identify issues with designs.

2 Related Work

In this section, we give a brief review of the field of formal methods and its application to railways, before considering approaches to railway graph layout and how they relate to our approach.

2.1 Railway Verification

Formal verification aims to provide a rigorous mathematical argument to show that a system or design meets a given requirement. A typical application area for formal methods is safety critical systems, of which railway control systems are a clear example. Many approaches apply formal methods to railway safety verification [15, 6, 14, 11, 24, 19, 13, 12, 16, 20, 27, 1], with much of this work focusing on the scientific development and application of results to examples in industry. However, uptake of these results by industry is impeded by complex notations and the heavy mathematical constructions that are involved [21].

Recently, there have been advances focusing on the accessibility of these approaches. Specifically, toolsets that support domain specific languages [21, 17] and graphical specification development environments [22, 18] have allowed railway engineers to model and verify systems in notations that are natural. However, when a verification attempt fails, methods for presenting the reasons for failure are lacking.

Another limitation with existing toolsets is that users are often required to re-draw and re-enter railway layouts directly into the verification toolset when geospatial information for track plans is unavailable. Re-entering data is clearly cumbersome and time consuming, whilst importation of verification data tends to be hard as geospatial information is often missing from the data.

In this paper, we address these points by adapting visualisation research results to this area. In particular, we apply energy-based graph layout approaches to automatically import and derive geospatial information for track plans. We then utilize approaches to key frame visualisation in order to provide feedback on failed verification attempts. We have incorporated these into the OnTrack railway verification toolset [22] and have evaluated the work with end users.

2.2 Graph Layout for Railways

Railway track plans illustrate how various railway lines are connected at stations and junctions, and can be interpreted (and drawn) as graphs. Such depictions are natural for engineers working within the railway domain and can be of benefit in visualising points of failure. However, they are less relevant with respect to the correct functioning of the railway.

Existing approaches to drawing metro maps and network layouts [31, 30, 26, 7] provide possible methods for visualising track plans. Here, stations are placed in the plane, with their spread-out geographic locations taken into account alongside desirable æsthetic properties. For track plans, however, we deal with small geospatial areas with complex network topologies; geospatial considerations are far less important to us, and in any case are typically unavailable (particularly if the railway system has not been built).

In order to draw the track plan automatically, we use energy-based methods for graph drawing [23, 8, 4]. These methods modify the graph locally using a scoring function to determine if the layout has improved based on the selected æsthetic criteria. Such methods have been used for general graph drawing but have not been adapted for track layout. We create a method that optimises for the desired properties of track plans, making drawings useful for domain scientists and railway engineers.

2.3 Dynamic Data Visualisation

Visualising a railway safety issue – such as how two trains might collide – requires a visualisation of the track plan and the trains that are moving on it. This is a dynamic multivariate graph visualisation problem [5] where the attributes (trains and point/signal states) are dynamic but the network topology remains the same.

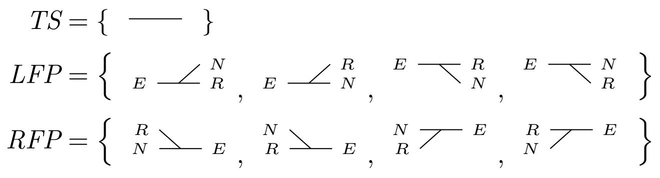
There has been significant work in the area of dynamic data visualisation. In much of this work, animation is of benefit if it is a short animated transition around a key event [29, 2]. Experimental results [3] have found that a "small multiples" representation (visualising dynamic attributes as colour on a static graph) can provide lower user response times with no significant difference in error when compared to animation.

In the railway verification community, signalling engineers often step through safety failures like mathematicians step through the lines of a proof. Our visualisation must not only be perceptually effective, it must also support the cognitive map with which railway engineers and formal methods researchers approach the problem. We thus provide an interactive step-through approach, with support for small multiples around key events.

3 Railway Visualisation Methods

In this section, we present our simulated annealing algorithm [8] for computing a railway layout, followed by details concerning our counterexample visualisation using key frames.

For our purposes, railway track plans are comprised of: track segments (TS); and points, which may be left-facing (LFP) or right-facing (RFP), and whose straight and offshoot tracks are designated as normal (N) or reverse (R), one of each. (The specific purposes of these distinctions is unimportant for this paper.)


TS = { }
LFP = { E N R , E R N , E R N , E N R }
RFP = { R N E , N R E , N R E , R N E }

A *railway graph* $RG = (V, E)$ is an undirected graph where the vertices are either track segments or points: $V = TS \cup LFP \cup RFP$. A track plan layout is an assignment of two-dimensional positions (x, y) for all vertices in the railway graph.

3.1 The Simulated Annealing Layout Algorithm

To establish a good layout for a track plan (i.e., such that it conforms with validity criteria and is therefore understood by railway engineers), we employ a simulated annealing algorithm [8]. This algorithm is given as follows:

```
$\ell \leftarrow$ initial (random) layout;
temp $\leftarrow$ nodeCount($\ell$);   -- *initial temperature*
best $\leftarrow \ell$;
iter $\leftarrow$ 0;
while temp $> 0$ do:
    iter $\leftarrow$ iter+1;
    $\ell \leftarrow$ tweak($\ell$, temp);
    if $\nu(\ell) > \nu(\text{best})$ or rand(0,1) $< \exp\left(\frac{\nu(\ell) - \nu(best)}{temp}\right)$
            then best $\leftarrow \ell$;
    if iter mod $\lfloor$temp$*$c$\rfloor = 0$
            then temp $\leftarrow \lfloor$temp$*$d$\rfloor$
return best
```

Each point is initialised with a random type from the sets LFP and RFP (as determined by the given data) to provide an initial layout. This layout is then repeatedly tweaked in an effort to discover an optimal (best) layout.

There are three essential components to our algorithm: a *temperature (temp)*; a *valuation* function ν for rating layouts; and the *tweak* function.

- From an initial value (equal to the size of the graph), the *temperature* parameter is periodically reduced by a preset constant factor d $\in (0, 1)$, and the algorithm iterates until this temperature reaches zero.
- The valuation of a layout is penalised if:
  - *Node overlap:* the distance between two distinct nodes is zero;
  - *Lack of gap:* the x-coordinates x_1 and x_2 of two unconnected nodes are too close, i.e., $|x_1 - x_2| < 1$;
  - *Long edges:* the distance between two edges is greater than an ideal.
- The *tweak* function takes a layout and a temperature and produces a new layout by making a series of random changes; each point in the graph may be changed to another point of the same type (left/right-facing). The number of such changes is dependent on the temperature, with higher temperatures giving rise to more changes. Hence, tweaking becomes more subtle as the algorithm progresses.

There are two features of the algorithm worthy of comment:

1. The temperature is kept fixed for a number of iterations which is some preset constant $c > 0$ times the temperature before being reduced. Thus, the number of iterations carried out at a given temperature decreases exponentially with the temperature.
2. There is randomness incorporated into the algorithm in that the layout may be randomly replaced by a less-optimal layout; however, the likelihood of this diminishes exponentially with the temperature and the poorness of the layout compared to the currently-identified best.

Fig. 1 shows the results of applying our simulated annealing. The first layout in the figure is the ideal layout, whilst the following three illustrate progressive results. The unreadable labels are immaterial; all that is of interest is the layout.

For this run, we set the temperature decay d=0.75 and the iteration constant c=3; and used the following penalties in scoring: each node overlap and lack of gap is penalised -1; and each edge greater than 1 is penalised -10. As is apparent, the algorithm effectively works from a poor layout towards ones close to the ideal (though flipped vertically).

3.2 Verification: Insights from Failure

When verification tools discover a problem (such as the possibility of a crash), they can evidence the problem by providing a sequence of events leading from the initial configuration to the problematic state. However, being derived from a proof tool, this sequence is often provided in a mathematical language that is unnatural for signal engineers.

To overcome this, we have implemented an approach to visualising these traces in the OnTrack toolset [22]. The last image in Fig. 2 shows one way to depict a possible error state. Each step in the mathematical trace (i.e., each event causing a system state change) is shown through highlighting the state of the track plan elements. Users then have the option to step through each system state leading to the error.

For short traces, this approach can be sufficient. However, counterexample traces can easily become thousands of steps long with many of the steps being superfluous to what is actually causing the problem. We have thus provided users with a simple drop-down filter that allows them to select which types of key frames to present, specifically frames that correspond to particular events in the trace. For selection criteria, we include events from the generated trace. These include events like “route set” or “point switched position”. Fig. 2 shows an example of applying a filter that only shows “route set” events.

4 Expert Feedback

Four experts evaluated our tool and provided feedback (via interviews of approximately 30-45 minutes). Participants consisted of railway engineers working

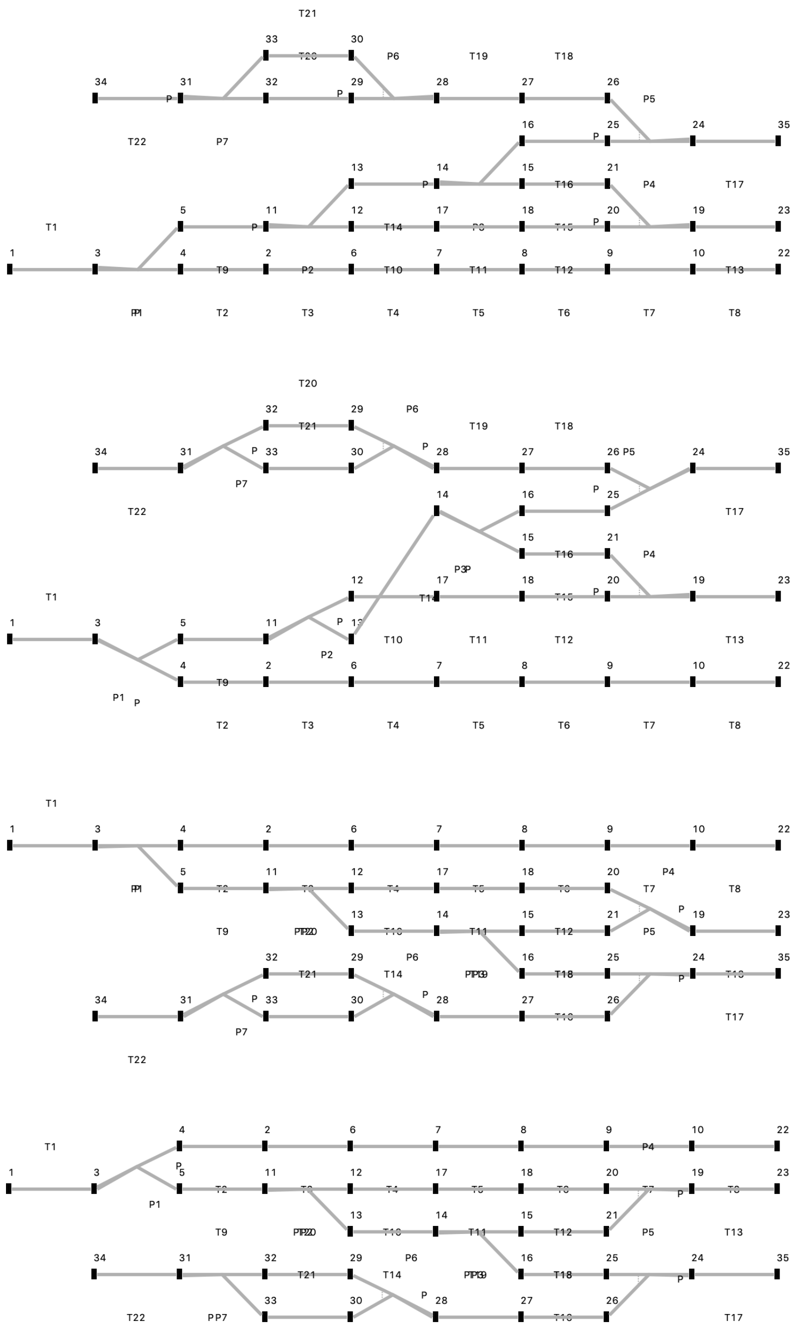


Fig. 1. Sample results from applying simulated annealing.

as safety test engineers in industry (*P1* and *P4*), and academics working on applying formal methods to railways (*P2* and *P3*). The participants were asked to provide feedback on:

- The usefulness of the automatic layout when importing existing railway data. Participants were given a demonstration of our simulated annealing approach and example layouts. They were then asked to compare the automatic layouts to existing practice and to rate the usefulness from 1 (not useful) to 5 (highly useful) as a step towards the end goal of formal verification.
- The usefulness of the visualisation of counterexamples. Participants were presented with a counterexample trace and a demonstration of key event selection. They discussed the key events they would like to see and how useful an approach it would be.

4.1 Importation of Data and Automated Layout

The participants working in academia were keen on the approach, with average ratings: *General usefulness* 4; *Usefulness as a starting point for re-drawing* 3; and *Usefulness for verification* 5.

Clear layouts take precedence over geography. *P2* provides statements to support this idea: *"When verifying, you do not care too much about locations; but having a clear representation helps a lot in identifying errors"*. Similarly *P3* noted: *"I don't really care about the physical reality of the situation as long as I have the logic in place, that is perfect for me."* There is evidence that the automatic layout would have an impact on work practices, with *P2* noting the approach would *"save a lot of manual work"* and *P3* stating it is a *"good way to share benchmarks for verification without spending time encoding"*.

However, *P3* cautioned using automatic layout as a starting point for editing as it may lead to human errors: *"Human error may be a problem if the plan is laid out automatically and doesn't match the real-life model"*. *P2* noted that it would be useful to *"set a region as a 'correct' part of the plan before re-applying, so that you eventually get a plan that corresponds to the real plan"*. This indicates that we should use actual geographic information when available.

The participants working within the railway industry on average rated automatic layout as follows: *General usefulness* 3; *Usefulness as a starting point for re-drawing* 3.5; and *Usefulness for verification* 4. These participants noted that the usefulness depends on company specific formats versus shared data. *P4* noted: *"It could be very useful for some things but not for others; If you don't have the original scheme plan, it would be very useful."*

P1 stated that the approach would be more useful if it provided affordances for user steering or manipulation of the layout, particularly for point directions. From these participants, it is clear that if we have existing track layout information we should use it, but that the automatic layout tool can be useful when this information is not present.

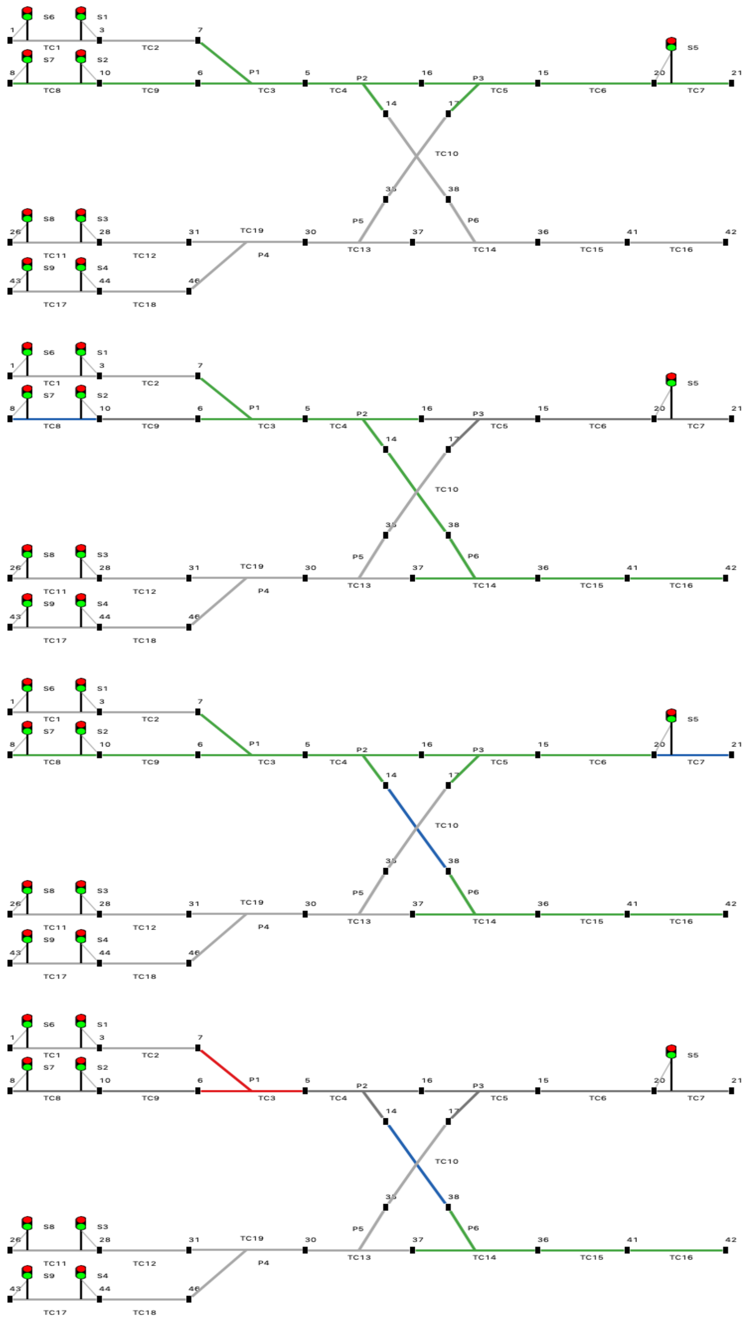


Fig. 2. Presentation of an error trace using the "route set" filter. Green indicates a set route, blue indicates occupation by a train, red an issue, here a "run through".

4.2 Counterexample Visualisation

With respect to visualising counterexample traces, the feedback was positive. Academics gave the following average ratings: *General usefulness of step function* 4.5; and *Dynamic selection of key frames* 4.5. Industry engineers gave the following average ratings: *General usefulness of step function* 4.5; and *Dynamic selection of key frames* 4.5.

P1 noted that the implemented visual approach was in line with their mental model when performing a trace and would save time: *"This is what I do now but without the visual assistance, which would make it quicker"*. All bar one participant explicitly stated that they would like to have all counterexample steps available as well as key frame selection. *P3* stated: *"I think you need both the full trace and be able to jump between states; over simplification doesn't always make things easier"*. Interestingly, participants agreed that key frames would be very useful for experienced users, but a full trace would help for novice users. For example, *P2* noted: *"I think it depends on experience: senior verification engineers may identify problems using only a few key frames, but younger people may like to see the full trace to help understanding"*. *P3* noted: *"Advanced verification experts could look at brief traces and likely detect problems"*. All participants also agreed that the most vital key frame would be "route setting" as described by *P4*: *"Route setting will highlight where the error is in the control table."*

Participants suggested improvements, with three participants saying that viewing detail in time around a particular key frame would be useful. *P1* stated: *"It may be an option to have few of them, maybe 4-5 before and after an event"*. Similarly, *P2* and *P3* would like to see events within an area of a scheme plan, with *P3* stating: *"I might like to see all steps within a particular section."*

5 Conclusion

We have presented a technique that increases the accessibility and usability of formal methods within the railway verification community. Our solution consists of two parts. Firstly, we apply simulated annealing to automatically lay out railway graphs when no geographic information is available, improving interoperability between railway data sets. Secondly, we present key frame visualisations to support the understanding of counterexamples as presented in the language of the domain. Both approaches have been evaluated by expert users.

In future work, we would like to follow up on feedback from expert users and use small multiples [28] to visualise details (i.e., nearby frames) around key frames of interest. Similarly, we would like to explore the application of simulated annealing within subgraphs of a railway graph. To this end, constraint-based methods [9, 10] could be useful. Finally, we would like to perform more formal evaluations of the railway layout algorithm through metric experiments as well as user studies on realistic tasks that railway engineers are required to perform on a regular basis.

Acknowledgments: The authors wish to thank Siemens Rail Automation UK, in particular Simon Chadwick, Mark Thomas and Thomas Werner for their support in undertaking this work.

References

1. N. Aber, B. Blanc, N. Ferkane, M. Meziani, and J. Ordioni. RBS2HLL. In *RSSR'19*. Springer, 2019.
2. D. Archambault and H. C. Purchase. Can animation support the visualization of dynamic graphs? *Information Sciences*, 330, 2016.
3. D. Archambault and H. C. Purchase. On the effective visualisation of dynamic attribute cascades. *Information Visualization*, 15(1), 2016.
4. A. Barsky, T. Munzner, J. Gardy, and R. Kincaid. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 2008.
5. F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1), 2017.
6. C. Bernardeschi, A. Fantechi, S. Gnesi, and G. Mongardi. Proving safety properties for embedded control systems. In *Dependable Computing — EDCC-2*, pages 321–332. Springer, 1996.
7. U. Brandes and D. Wagner. Using graph layout to visualize train interconnection data. In *Graph Drawing*. Springer, 1998.
8. R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM Trans. Graph.*, 15(4), 1996.
9. T. Dwyer. Scalable, versatile and simple constrained graph layout. *Computer Graphics Forum*, 28(3), 2009.
10. T. Dwyer, Y. Koren, and K. Marriott. Ipsep-cola: An incremental procedure for separation constraint layout of graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5), 2006.
11. C. Eisner. Using symbolic model checking to verify the railway stations of hoorn-kersenboogerd and heerhugowaard. In *Correct Hardware Design and Verification Methods*, pages 99–109. Springer, 1999.
12. A. Ferrari, A. Fantechi, S. Gnesi, and G. Magnani. Model-based development and formal methods in the railway industry. *IEEE Software*, 30(3), May 2013.
13. A. Ferrari, G. Magnani, D. Grasso, and A. Fantechi. Model checking interlocking control tables. In *FORMS/FORMAT 2010*. Springer, 2011.
14. W. Fokkink and P. Hollingshead. Verification of interlockings: from control tables to ladder logic diagrams. In *FMICS'98*. CWI, 1998.
15. J. F. Groote, S. van Vlijmen, and J. Koorn. The safety guaranteeing system at station hoorn-kersenboogerd. Technical report, Utrecht University, 1995.
16. A. E. Haxthausen, J. Peleska, and R. Pinger. Applied bounded model checking for interlocking system designs. In *Proceedings of SEFM 2013*. Springer, 2014.
17. A. Idani, Y. Ledru, A. Ait Wakrime, R. Ben Ayed, and P. Bon. Towards a tool-based domain specific approach for railway systems modeling and validation. In *RSSR'19*. Springer, 2019.
18. A. Iliasov, D. Taylor, L. Laibinis, and A. Romanovsky. *SAFECOMP 2018*. 2018.
19. P. James. Sat-based model checking and its applications to train control software. Master's thesis, Swansea University, 2010.

20. P. James, A. Lawrence, F. Moller, M. Roggenbach, M. Seisenberger, A. Setzer, K. Kanso, and S. Chadwick. Verification of Solid State Interlocking Programs. In *Software Engineering and Formal Methods*, Lecture Notes in Computer Science. Springer, 2014.
21. P. James and M. Roggenbach. Encapsulating Formal Methods within Domain Specific Languages: A Solution for Verifying Railway Scheme Plans. *Mathematics in Computer Science*, 8(1), 2014.
22. P. James, M. Trumble, H. Treharne, M. Roggenbach, and S. Schneider. OnTrack: An open tooling environment for railway verification. In *NFM'13*, 2013.
23. T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1), 1989.
24. K. Kanso, F. Moller, and A. Setzer. Verification of safety properties in railway interlocking systems defined with ladder logic. In *AVOCS08*. Glasgow University, 2008.
25. H. D. Macedo, A. Fantechi, and A. E. Haxthausen. Compositional model checking of interlocking systems for lines with multiple stations. In *NFM'17*, 2017.
26. M. Nöllenburg. A survey on automated metro map layout methods. In *Schematic Mapping Workshop*, 2014.
27. C. Parillaud, Y. Fonteneau, and F. Belmonte. Interlocking formal verification at alstom signalling. In *Proceedings of RSSR'19*. Springer, 2019.
28. E. Tufte. *Envisioning Information*. Graphics Press, 1990.
29. B. Tversky, J. Morrison, and M. Betrancourt. Animation: Can it facilitate? *Int. Journal of Human-Computer Studies*, 57(4), 2002.
30. A. Wolff. Drawing subway maps: A survey. *Informatik - Forschung und Entwicklung*, 22(1), 2007.
31. H.-Y. Wu, B. Niedermann, S. Takahashi, and M. Nöllenburg. A survey on computing schematic network maps: The challenge to interactivity. In *The 2nd Schematic Mapping Workshop*, Vienna, Austria, 2018.