Modelling Railway Networks with Bigraphs: Electrification, Failures, and Optimisation

Ricardo Almeida[1], Susmoy Das[1], Blair Archibald[1], Muffy Calder[1], and Michele Sevegnani[1]

School of Computing Science
University of Glasgow, Glasgow, United Kingdom
firstname.lastname@glasgow.ac.uk

Abstract. Upgrading rail network infrastructure involves complex system design decisions that can be informed by suitably abstract models. Ideally, the modelling techniques support formal analysis e.g. of safety, security, resilience, and performance, the models are extensible, and they are accessible to both railway engineers and policy makers.
We propose using bigraphs—a diagrammatic formal model with user-defined entities and rewrite rules—as a visual and intuitive approach to extensible railway system modelling. An example is presented: electrification rollout and the adoption of battery-powered trains that includes the impact of (probabilistic) power blackouts and supports system-level optimisation e.g. selecting which track segments to electrify, without resorting to constraint programming languages. Formal analysis via model checking is used throughout.
This work represents a shift in the use of formal methods in railway engineering: from verifying isolated components such as signalling, to supporting formal, system-level design and decision-making.

Keywords: Model checking · Bigraphs · Markov Decision Processes · Discrete-Time Markov Chains · Rail Networks · Electrification.

1 Introduction

Railway systems are inherently spatial, involving both static infrastructure and dynamic behaviour. We propose a diagrammatic approach to modelling based on Milner's bigraphs [13], a formalism capable of representing both spatial hierarchy and non-local connectivity as well as dynamic evolution through user-defined rewrite rules. The visual representations are also user-defined, so they can be intuitive and tailored to specific domains. Here, we employ a rich theory of bigraphs, employing conditional, probabilistic, and action-based extensions [3,4].

Bigraphs have previously been applied to verify properties such as safety, reliability, and predictability in location-aware, event-driven systems [12]—including applications in wireless sensor networks [16] and transportation [7]. Despite their visual nature, bigraphs retain formal rigour and support advanced reasoning techniques, such as (probabilistic) model checking. This makes them a strong

complement to traditional formal methods in rail engineering, which have typically focused on low-level aspects like verifying segment locking [9,14]. In contrast, our work applies formal methods to higher-level system design.

An example bigraph of a simple rail network decomposed into track *segments* that can be considered collectively as *routes* is shown in Fig. 1. The train icon represents a train and a basic track segment is represented by a rectangle (it is common to mix meaningful diagrams with standard geometric shapes). The green links indicate contiguous track segments (within a route). The dashed rectangles (without shading) represent (possibly) disjoint parts of the system. Fig. 1a is a bigraph representing two routes, the left route has two track segments, the first of which contains a train; the right route has only one segment, which does not contain a train. The gray filled rectangles in Fig. 1b abstract away an unspecified bigraph that might exist/connect there. The rule in Fig. 1b indicates that a train may move between the two connected segments. This simple model is developed further in Section 3, including a much richer definition of Segment.

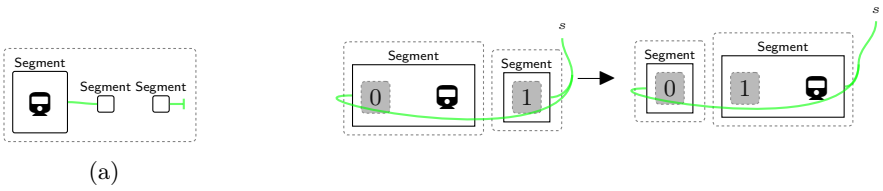


Fig. 1: (a) Bigraph model of a basic rail network composed of two routes, each of which is composed of track segments, which may or may not contain a train. Segments are connected by green links, which are ordered from left to right. (b) A rewrite rule that shows how trains can move between two connected segments. Everything else in the segments remains the same, and this is abstracted as simply 0 and 1.

We make the following research contributions:

- **Bigraphs as an intuitive, diagrammatic modelling technique for rail networks.** We present a bigraph model of rail networks that includes partially electrified infrastructure and an energy mechanism. The bigraph model is highly parameterised and extensible and the underlying models are MDPs (Markov Decision Process). Formal analysis is via model checking and reward structures, using the BigraphER [15] tool.
- **Analysis of electricity failures.** We show how to quantify and analyse the impact of electricity failures on battery levels of the trains.
- **Optimal electrification strategies.** We develop an extension to the model that initially contains no electrified segments, and then explore all possible electrification combinations to find the ones that guarantee all trains reach their destination with a safe amount of charge left.

2 Bigraphs and Bigraphical Reactive Systems (BRSs)

Bigraphs specify models based on both spatial relationships—e.g. in Fig. 1 train entities (we denote either 🚆 or Train) are *nested* (i.e. placed inside) in rail Segments—, and non-local linking—e.g. a Segment has a single green link[1] determining the next segment to move to in a route. Nesting/Linking can be between arbitrary numbers of entities, e.g. an entity can have any number (including 0) children and a link can have any number of entities on it (these are hyperedges, not classic binary links). Links can either be open or closed, the former allowing the possibility for the link to be established/terminated dynamically as the system evolves. The entities in the model, and how they should nest/link, are fully user-specified and this makes them flexible to a wide variety of scenarios.

We allow parameterised entities [5], e.g. Charge(80), Charge(65), ..., that represent entities carrying some value[2].

Rather than needing to specify the entire bigraph of a complete system each time, special structures allow some information hiding. The dashed *unfilled* rectangles are *regions* that denote these two elements of the system might be[3] in two different spatial locations, the dashed *filled* rectangles denote there may be additional nested entities (including none), while the names above the bigraph denote that there may be other entities on this link.

Bigraphs represent the state of a system at a single point. To allow systems to evolve over time a *user can define* a set of rewrite rules (sometimes called reaction rules) that specify system behaviour. Rewrite rules have the form $L \longrightarrow\!\!\blacktriangleright R$, where L and R are bigraphs. This means that a match of L (within a larger) bigraph can be replaced by R to evolve to a new state. For example, Fig. 1b shows a rule that moves trains between segments. Here we have an L consisting of a 🚆 on a Segment linked to another Segment to its right, while R has the train on the right segment implying the train has moved to the next segment. We abstract away from entities using *sites*, which are like free variables, and we draw them as dashed gray filled rectangles. This allows the movement rule to be applied in general situations: regardless of what else is in the segments, the rule gets applied so long as there is at least a train while everything else in the segments (represented by sites 0 and 1) remains the same. We use parameterised rewrite rules to define families of rules for different parameter values.

We use Conditional Bigraphs [3] that allow constraints to be placed on elements inside sites (and in general wider context; but we only constrain sites here). For example, when defining a train-movement rewrite rule such as the one illustrated in Fig. 1b), we can guard it with if $\langle -, \overset{\text{Charge}(0)}{\square}, \downarrow \rangle$ to ensure trains have charge in order to move. Here, $-$ means should not exist, Charge(0) is the bigraph we want to disallow, and $\downarrow$ means we disallow in the sites.

[1] In practice, we nest an additional entity rather than linking segments directly. See Section 3.

[2] In theory, this corresponds to defining distinct entities for each value.

[3] They may be either completely disjoint or siblings, but never nested below one another.

An initial bigraph, with a set of rewrite rules, is a bigraphical reactive system (BRS). To analyse a BRS, we start from the initial state and apply all possible rules at each step. When multiple rewrites are possible for a state, we apply all rules resulting in state-branching (looking at all futures). This generates a transition system representing how the system can evolve. We can later ask questions such as "Can my system reach this failure state?".

Sometimes, applying all rules is too strong. To gain more control, rewrite rules can be organised into *priority classes*. Rules with the highest priority are applied first, and only when there are no applicable rules do we try the next class (and so on). If several rules within a class are applicable we still get the branching behaviour.

The BigraphER tool we use supports instantaneous rules [15,5], that allow multiple rewrites to apply atomically without introducing intermediate states. These rules can enhance efficiency by removing redundant interleavings and intermediate states, yielding a smaller, more semantically meaningful transition system—beneficial for model analysis and verification.

We employ two extensions to conditional BRS. Reaction rules may include weights, resulting in *probabilistic bigraphical systems* (PBRS) and then further extended with non-deterministic actions, resulting in *action bigraphical reactive systems* (ABRS) [4], in which case the underlying model is an MDP (Markov Decision Process).

3 Diagrammatic Model of A Railway

We consider a simplified semi-electrified railway scenario as a proof of concept for our modelling approach. The rail network is divided into track segments of equal length (10 miles each), and each segment may be electrified or not. We assume identical battery-powered trains, but support 3 different routes. Routes may include stops at stations.

Time in the system advances in discrete units, or clock ticks, corresponding to 10 minutes each. At any given moment, each active train is either moving at a constant speed of 60 miles per hour or stopped at a station for exactly 10 minutes. This ensures a headway of 10 minutes between successive trains (i.e. the minimum time interval maintained between consecutive trains on the same route to ensure safety). These parameters could be adjusted by, e.g. increasing the granularity of how time increases to support trains moving at different speeds and other variants.

The model incorporates a simple energy mechanism: a train gains 10% charge for every electrified track segment it traverses or every 10-minute stop at an electrified station. If the segment/station is not electrified, the train loses 5% charge to traverse it.

In the next section we describe the bigraphical states of the model. The dynamic behaviour of the system, i.e. the transitions that define how to move between these states, is defined via rewrite rules in Section 3.2.

3.1 A Diagrammatic Model of Rail Network, Routes and Trains

Railways are spatial in nature and this is reflected in our bigraph model. Key entities are Segment and Train representing where trains are in the network and where they can move to. We first describe the basic entities and then how we model specific routes. An example model state is in Fig. 2 and we describe the key elements below. Links through waypoints WP are the routes the trains take, including their designated stops. For visual clarity, some of these links have been omitted as have links for activities such as timekeeping.

Each Segment has a name SName($name$), e.g. SName($T1$), a flag E determining if it is electrified: E.Yes or E.No. The segment might also contain a *single* Train[4]. As our Trains are battery-powered, they nest a Charge(n) that describes the percentage charge left.

The previous entities were largely physical in nature, e.g. the train on a track. We can also utilise additional entities within the same model to represent non-physical constructs such as routes.

We overlay routes onto segments by nesting waypoint entities (WP). Each waypoint contains a Next pointer that determines where a trains should go next on a route (or empty if the train is at a terminus). A series of linked waypoints then describes a single route. Each train contains a RouteN entity that points to the waypoint it should head to next on the route. The entity RStops denotes segments where a train might wait, e.g. a station. A segment may be a stop for some routes but not for others.

To allow discussions about specific routes, we store all route information in a R entity (within a separate Routes entity). Each R($name$) contains a Start linked to the first waypoint in this route; Trains, linking to all active trains on this route; and Stops, linking to any tracks where this route stops.

Trains also maintain some non-physical information such as an internal state machine that determine the next Action to perform: Wait to stay in the same segment, or Move to move to the next segment on the route (if one exists).

3.2 Rail Model Dynamics

So far we have considered only the static aspects of the system: describing trains, rail segments, routes, etc. We now define system dynamics as rewrite rules.

All trains move synchronously over these steps, i.e. there is no interleaving of steps, and we enforce this using a global clock entity (updates only happen while a train's local clock is not at the most recent time) and instantaneous rules. This is similar in style to discrete event simulation. The mechanism of using a local clock (LC(t)) per agent alongside a global clock (Clock(t)) is inspired by Albalwe et al.'s work on modelling real-time systems using bigraphs [1].

In detail, the steps are:

1. **Train Departures.** Timetabled departures are modelled by a parameterised rule, `trains_starts`($t, route$)(Fig. 3) that is only applicable when the global

[4] We are not modelling segment locking/signalling directly, but this could be added.

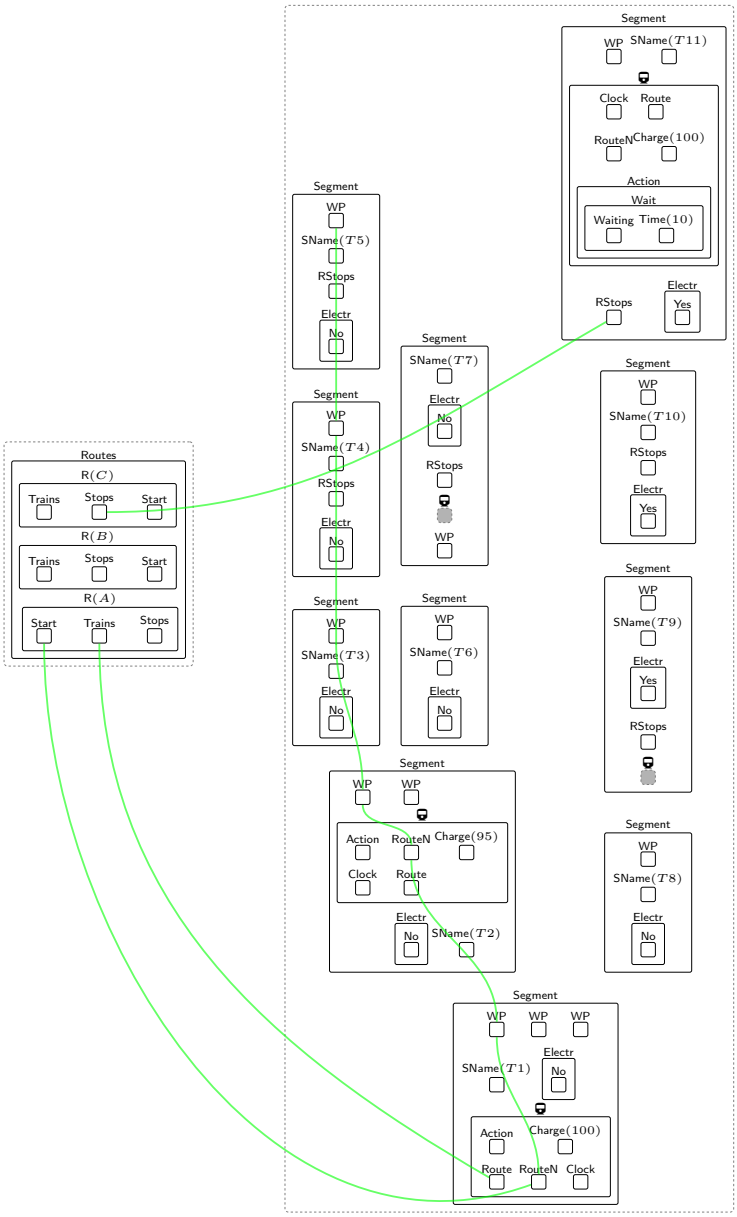


Fig. 2: (Partial) Example state of the Railway Bigraphical Reactive System, depicting five trains travelling along their respective routes (for simplicity, only a subset of links is shown). All trains depart with a full (100%) charge from $T1$ (e.g. note the link from Start in R(A) to RouteN). Trains lose charge when passing through non-electrified tracks (e.g. the train on $T2$), and regain it on electrified tracks such as $T11$. On $T11$, a train completing route C has recharged and, because this track corresponds to a designated stop (see the link from Stops in R(C) to RStops), it is currently waiting for 10 minutes before completing its route. Waypoints (WP) indicate the paths each route takes through the network, and the links collectively visualize the full structure of each route.

clock matches the departure time t for a route. It creates a new Train in the initial segment for that route (always $T1$ in our examples) given by Start. The Train is given 100% Charge, a clock set to the current global clock time, an empty Action, and the RouteN/Route pointers are updated to track the next waypoint and to denote this train is working on the given route.

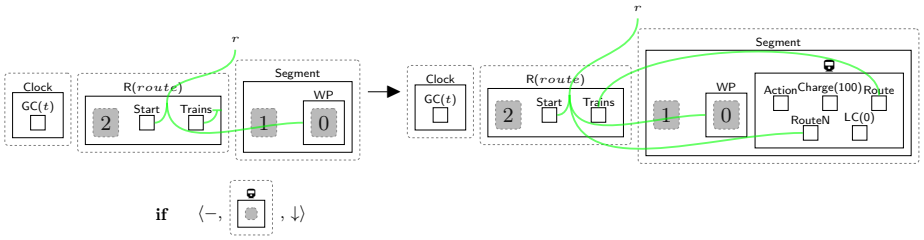


Fig. 3: The `train_starts`$(t, route)$ rewrite rule.

2. **Battery Update.** While detailed battery modelling is an active research field in its own right [18,17], we intentionally abstract away from such complexity here. Our aim is to capture only the high-level charging and discharging behaviour through a simplified rule-based model. Modelled by three rules `gain_charge`(t, c) (Fig. 4), `lose_charge`(t, c), and `keep_charge`(t, c), each train gains 10% charge when on an electrified segment (e.g. tracks 9-11), loses 5% *moving* on a non-electrified segment, or maintains current charge if stopped. The rules all have the same form (and so we only show `gain_charge` here): match on the existing charge levels (Charge(c)) and segment setup, e.g. is it electrified, and perform the required update. Battery-update actions tag the Train with the token Update to signal readiness for the next step.

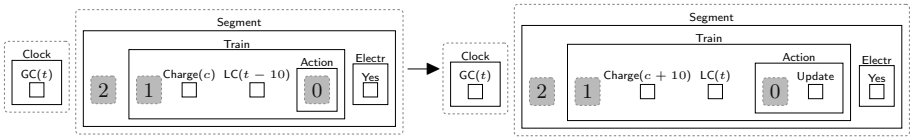


Fig. 4: The `gain_charge`(t, c) rewrite rule.

3. **Station-Wait Handling.** If the train's Action is currently set to Waiting, then we do no movement and instead consume the Update token and set Waiting to Over. If Over is already present, or the train was already moving, Action is updated to Action.Move.
4. **Train Movement.** Trains with a Move action are moved using the rule `move_train` (Fig. 5). This rule looks at the Next waypoint in a route and

moves the train to the segment containing that waypoint. If there is no next waypoint, we are at a terminus and this rule does not apply.

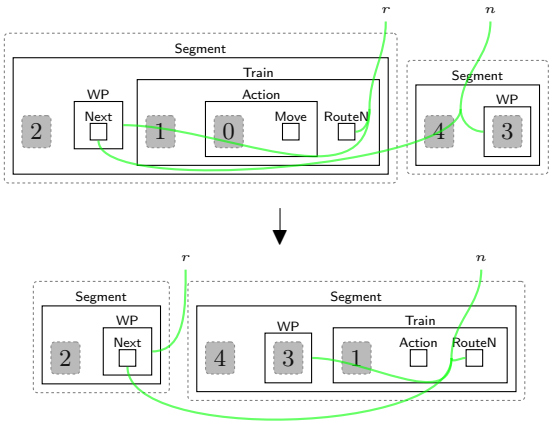


Fig. 5: The `move_train` rewrite rule.

5. **Station-Entry Handling.** The rule `stop_on_track` determines when the next action, e.g. the action in the next round, of a train should be to wait in a segment (to model station stops). The rule checks if a segment is a designated stop on the train's route using the Stops links in the Routes region and, if so, sets the train's Action to Wait.
6. **Train Exit.** Finally, the rule `train_ends` handles trains reaching their terminus. In this case, we find a waypoint with no Next and remove the train.

Once all rules have been applied, the global clock is incremented with the `tick`(t) rewrite rule (not shown) to restart the process.

3.3 Extension: (Probabilistic) Electric Track Failures

The previous sections describe a model of an electrified rail network that is assumed stable: i.e. there is always electricity if we require it. This is not always the case in practice, e.g. fully renewable supply in remote locations causing blackouts. A core benefit of the model is that we can easily extend it, by adding additional entities/rules, to consider failure cases.

Here, we use a probabilistic BRS, to capture (probabilistic) power failure on an electrified track segment. The extension requires only a single rule `electrical_failure` (that moves a segment from electrified to non-electrified). We do not include the inverse rule to re-enable the power, but this could be easily added if required.

The `electrical_failure` becomes the second non-instantaneous rewrite rule in the PBRS (apart from `start_clock`), and we assign weights to both

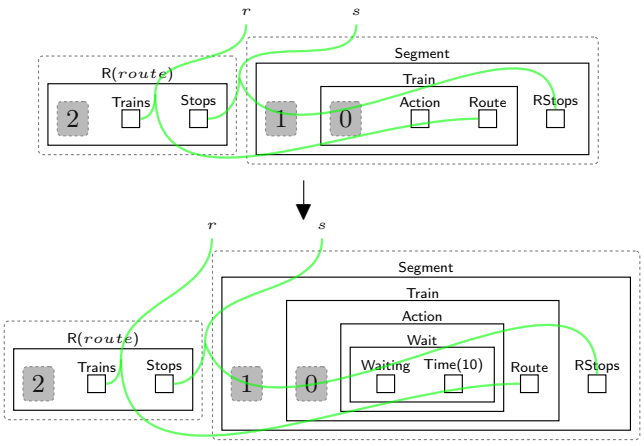


Fig. 6: The `stop_on_track`($route$) rewrite rule.

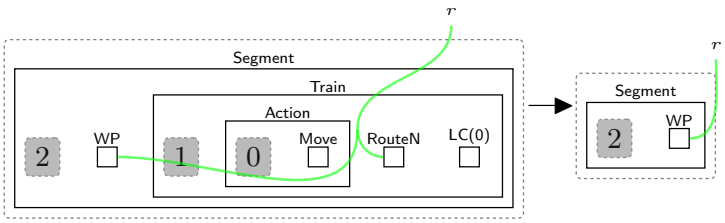


Fig. 7: The `train_ends` rewrite rule.

to tune the probabilities of either being applied. We give `electrical_failure` weight 0.01 and `tick` 1.0, making `tick`(t) 100 times more likely than `electrical_failure` when both can be applied. Note that these are not probabilities but probability weights, which are scaled relative to the number of matches possible. This means that in between train movements (when the only rewrite rules that can be applied are `tick` and `electrical_failure`), if there is only one electrified segment then the probability of it failing is 1%, and if there are four segments still electrified then the probability of any of them failing is 4%. We use these rules for probabilistic analysis in Section 4.1.

4 Model Analysis

We have implemented our model in BigraphER [15]: an open-source framework for constructing, manipulating, and executing bigraph models. For analysis, BigraphER supports model checking by generating a transition system—where

states are bigraphs and transitions are rewrites (up to bigraph isomorphism). The resulting transition system can be analysed using external (probabilistic) model checkers such as PRISM [10]. Our BigraphER model, and analysis scripts, are available online[5].

For analysis it is helpful to identify states that include bigraphs of interest, for example those where a train has run out of charge. In BigraphER, this is achieved using bigraph predicates which are essentially the left-hand sides of rewrite rules. States matching a predicate are labelled in the resulting transition system and these labels can be used in logical formulae when model checking. We use (Probabilistic) Computational Tree Logic ((P)CTL) [6,8] and its reward-based variant [11] to specify the properties of interest.

We can use a PCTL formula such as $\mathbf{P}_{=?}[\,\mathbf{F}\,\texttt{oob}\,]$ to compute the probability of reaching a state where a train is out of battery (true when $\mathsf{Charge}(0)$) is satisfied ($\mathbf{F}$ corresponds to the *eventually* operator). The result for the above query on our model was 0, confirming that there are no possible executions in which a train ends up in an unsafe state stranded with no charge. Other predicates include checking if two trains are on the same segment which could have catastrophic consequences. The same set of predicates could also be used in this regard to validate/debug the model. We used predicates such as battery levels below (above) 0 (100), inspecting wrongful links, checking if we update the same information twice, etc.

4.1 Impact of Electricity Failures on Final Battery Charge

The additional `electrical_failure` rule of Section 3.3 models failure of an electrified segment, e.g. a blackout. As the probability of failure is configurable (via setting different relative weights) we can use this to ask how robust the battery rail system is. In this case, we use final battery charge as a proxy for robustness, i.e. we want to avoid trains running out of charge and blocking routes.

BigraphER can generate a discrete time Markov chain corresponding to the extended model. It has $2,140$ states, $5,776$ transitions and can be generated in a few minutes on a commodity laptop. We then ask "What is the probability of reaching scenarios when k segments have failed and a train has a particular battery level?", expressed as $\mathbf{P}_{=?}[\,\mathbf{F}\,\texttt{charge_fail}(n,c)\,]$, where n is the number of failures and c the charge level. Results of probabilistic analysis are in Table 1.

Note that although four tracks may have failed, there can still be trains that do not use the failed tracks; hence, we can observe states where the battery levels of some trains remain at 100%. The same state may also reflect a lower battery level for the trains affected by the failures. The lowest battery level observed in our model is 80%. Additionally, since track segment failure is an extremely rare event (we have varied the weights assigned to the rule that disables electricity on tracks), we analyse the resulting proportions. For calculating these proportions, the probability of trains with a 100% battery level is used as the reference.

[5] `https://zenodo.org/records/16895542`

Table 1: Impact of Track Failures on Battery Levels. Top values indicate the weights of the electricity failure rule; table entries show the probability of a train reaching each battery level given the number of track failures.

	Failure Rate														
	0.01					0.05					0.1				
	Number of Failures					Number of Failures					Number of Failures				
Battery	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
80%	0.79	0.87	0.67	0.33	0.07	0.33	0.64	0.79	0.89	0.89	0.13	0.39	0.62	0.80	0.99
85%	0.85	0.87	0.67	0.32	0.07	0.48	0.74	0.86	0.94	0.89	0.26	0.57	0.76	0.90	0.99
90%	0.92	0.91	0.68	0.32	0.07	0.69	0.88	0.92	0.95	0.88	0.51	0.78	0.88	0.93	0.99
95%	0.96	0.91	0.69	0.33	0.07	0.83	0.91	0.93	0.95	0.88	0.71	0.86	0.89	0.93	0.99
100%	1.00	0.94	0.72	0.34	0.07	1.00	0.97	0.97	0.97	0.87	1.00	0.95	0.95	0.96	0.99

For example, with one failure (under a failure weight of 0.01), we observe a $(0.87/0.94) \times 100 = 92.5\%$ probability of trains reaching battery levels of 80%.

Let us consider the results when the failure weight is 0.01. With no segment failures, the proportion of trains reaching the lowest battery level is significantly lower. As the number of failures increases, the likelihood of trains reaching the minimum threshold also rises. For instance, with four track failures, all trains that began with a 100% charge ended with 80%, compared to 97%, 93%, and 92.5% for three, two, and one failures, respectively. For zero failures, this value drops to 79%. As the failure weight increases, the probability of trains reaching 80% grows sharply; e.g. at a failure weight of 0.1, this probability approaches one (0.99).

4.2 Optimal Electrification Strategy

The model introduced in Section 3.2 considers a network configuration in which some segments are electrified while others are not. In real-world settings, the decision to electrify parts of the railway network is a complex policy matter, influenced by a range of economic, environmental, and operational considerations. In this section, we explore a variant of the model aimed at identifying the optimal electrification strategy that ensures all scheduled trains reach their destinations with sufficient remaining battery charge. To support this analysis, we extend the initial bigraph with an energy-log region. For each route, this mechanism records a safe initial charge level that will ensure a train completes its journey while maintaining battery levels above a predefined threshold ($threshold_c$) at all times, represented by the $\mathsf{MinBatt}(safe_c)$ entity. We augment the Routes region with a Consumption entity, linked to the corresponding Log within the EnergyLog region. Additionally, each train starts its journey with the $\mathsf{Lowest}(100)$ entity, which will track the lowest battery level observed during the route.

To accommodate this functionality, we adapt the behaviour defined in Section 3.2 by introducing non-deterministic actions, transforming the BRS into an ABRS. This augmented system introduces an electrification phase prior to the

original movement phase. In the electrification phase, we assume a predefined set of segment sequences, each of which may independently be electrified or not. The ABRS begins by branching over all possible combinations of electrification configurations (e.g. for three segment sequences, there are $2^3 = 8$ branches in total). Each branch proceeds to the movement phase, which need only consider one train journey per route. It advances as previously defined but with the following modifications:

- **Energy Log Creation** (after **Train Departures** and before **Battery Update**): When a train for a given route departs, a new Log is created under the corresponding EnergyLog region, initialized with MinBatt(100). This log remains open until this train completes its journey.
- **Battery Update**: This step is modified to account for battery depletion. If a train runs out of charge, its Charge(c) and Lowest(l) entities are replaced with OutOfBattery, preventing it from continuing.
- **Energy Log Completion** (between **Station Wait Handling** and **Train Movement**): If a train successfully completes its route (i.e. it did not deplete its battery), the log is closed with MinBatt($safe_c$), where $safe_c = initial_c - lowest_c + threshold_c$. Thus, $safe_c$ is a starting level of charge that accounts for the route discharges and guarantees the battery will remain within safe levels at all times.

The ABRS model described above has 623 states and 622 transitions and takes about a minute to generate. Note that, since instantaneous rules and failures not being modelled, many states are omitted, resulting in a smaller model compared to the PBRS. This reduction in the state space makes model checking more efficient.

4.3 Analysis of Optimal Electrification Strategy

In this section we consider how we can utilise the bigraph model to perform optimisation. The idea here is to show how the approach can be applied, and the results are likely not directly transferable to a real scenario given we only show a small rail topology.

For the next set of analysis, we assign reward structures to our models. A reward (dually cost) structure can be used to capture additional aspects of the system modelled by the MDP, such as track usage, battery levels, etc. We identify states to which we assign a reward using predicates. This would imply that leaving a state (which satisfies the predicate along) a path in the model will accumulate the associated reward. In the context of reward based model checking, *total reward* properties capture the accumulation of state and transition rewards over an entire (potentially infinite) path, similarly to *reachability* and *cumulative reward* properties. This is denoted in literature using **C** formulae [11]. For total rewards accumulated up until a time instant t, this is denoted as $\mathbf{C} \leq t$. In contrast, *instantaneous reward* properties evaluate the reward at a specific point in time. The reward property $\mathbf{I} = t$ assigns to each path the reward associated

with the state it occupies precisely at time t. Due to non-determinism in MDPs, querying exact reward values is impossible. Instead, we analyse reward bounds by resolving non-determinism to either maximize or minimize accumulated rewards. While train routes and schedules are fixed and unaffected by non-determinism, metrics like battery charge are. These concepts will guide the analyses discussed below.

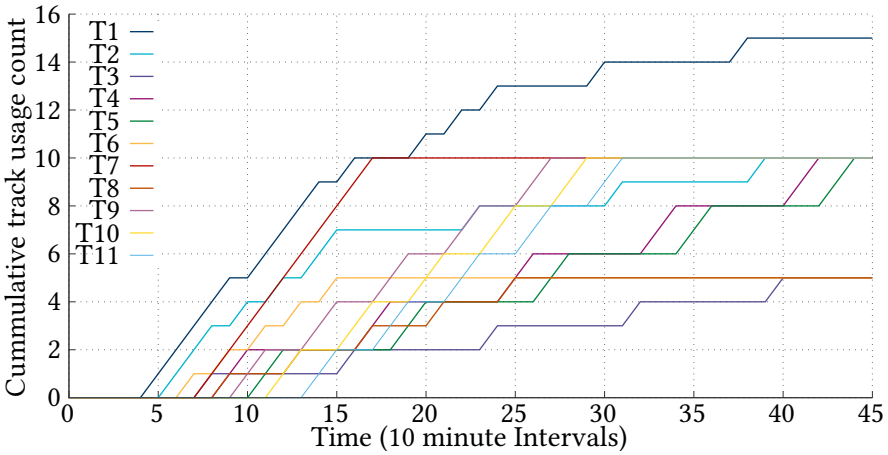


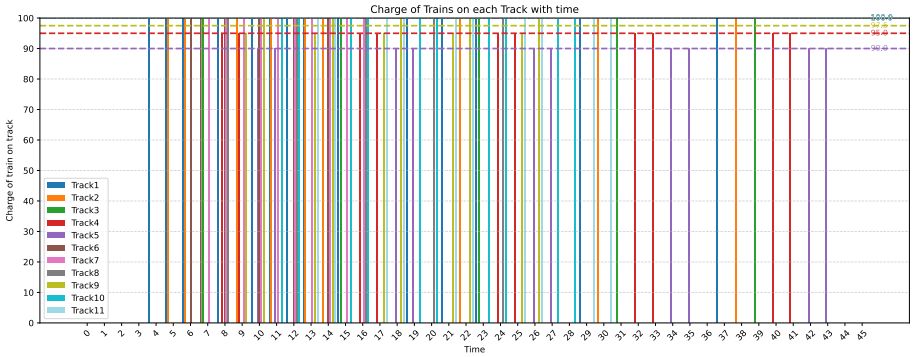
Fig. 8: Total number of times a track has been used as time progresses.

To analyse track usage, we employ total reward properties. A predicate identifies whether a track segment is live, i.e. currently occupied by a train—and a reward of 1 is accumulated each time this condition holds. Plotting the total rewards over time reveals increasing usage as more trains traverse the network. Segments with the highest accumulated rewards correspond to the most frequently used tracks. The results are shown in Fig. 8. Note, all trains considered in the time-table of our model reach the destination by $t = 45$.

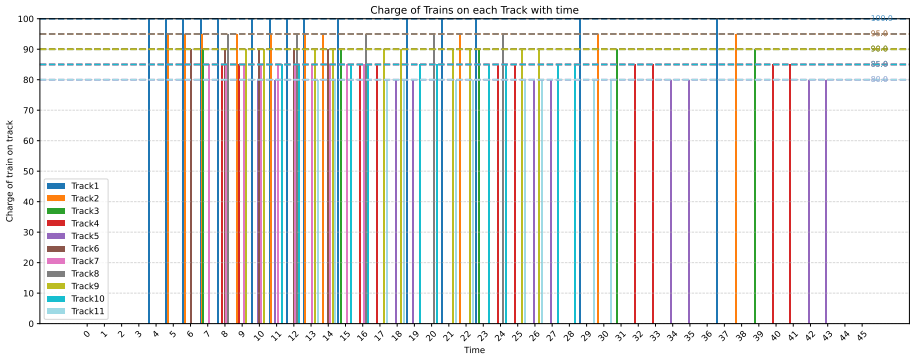
Track 1 exhibits the highest cumulative reward, which also serves as a validation point, as all trains begin their journey from this segment. Tracks 7, 9, 10, and 11 follow as the next most frequently used, with Tracks 2, 4, and 5 showing similar reward values but at later timestamps. These delays indicate that trains using these tracks have already passed through earlier segments—potentially electrified—where they could recharge. This highlights the utility of the graph in guiding infrastructure planning: early-use, high-frequency segments should be prioritised for electrification, as they contribute more directly to sustained battery levels.

In contrast, tracks 3, 6, and 8 are used significantly less. Thus, electrifying tracks 7, 9, 10, and 11 emerges as a key insight. However, practical constraints such as installation costs must be considered (e.g. electrifying continuous seg-

ments is generally more efficient than isolated ones), which could make tracks 9-11 good candidates for electrification.



(a)



(b)

Fig. 9: Tower graph showing battery levels on trains when they are on a track at the t-th time. The dotted lines show the mean charge of trains on each track. (a) shows maximum rewards (electrified tracks), while (b) shows minimum rewards (non-electrified tracks).

An additional insight concerns the safety of leaving a track segment non-electrified. To evaluate this, we examine the average battery level of trains upon entering each segment. A predicate accumulates a reward of x when a train with charge x occupies a segment, and this instantaneous reward is calculated over time. A value of zero at any time indicates no usage, not the presence of a train with no charge, which is impossible as previously established.

In Figure 9, bars represent the battery levels of trains occupying each segment at time t, while the dotted line shows the average battery level, computed from non-zero values. Based on prior analysis, Tracks 2–8 show lower usage and

are candidates for non-electrification. Specifically, Track 6 was modelled with optional electrification; this analysis supports the decision to leave it unelectrified, as trains traversing it maintain an average charge of 95% (Fig. 9b). This analysis reinforces confidence in the system's reliability, offering valuable guidance for future policy decisions. In the scenario where Tracks 9, 10, and 11 are electrified and Track 6 is not, the results demonstrate a resilience analysis. Specifically, we simulate a worst-case scenario in which no tracks are electrified—representing disruptions such as power outages or natural disasters. Even in such conditions, trains successfully complete their routes with an average battery level of 80%, assuming an initial charge of 100%.

The maximum reward values (Fig. 9a) anticipate future extensions beyond Track 11, informing infrastructure planning if those segments are electrified. This is further supported by a filtering mechanism that identifies underutilised tracks and evaluates the average battery levels on the remaining ones. Collectively, these findings provide a robust framework to guide policymakers in identifying which track segments can be safely excluded from electrification.

5 Conclusion and Future Work

This work explores the use of Bigraphical Reactive Systems (BRS) for modelling and analysing railway electrification strategies. We have demonstrated that bigraphs provide an intuitive and extensible formalism for representing both the structural and dynamic aspects of railway systems. Our model captures partially electrified infrastructure, introduces an energy-tracking mechanism to simulate battery usage, and is extended to handle probabilistic events such as power failures.

Using model checking, we validate key safety predicates, including that no train runs out of battery and no two trains occupy the same segment concurrently. We further introduced track failures as a modelling variable, enabling analysis of battery levels under adverse conditions and supporting resilience assessment. The modified ABRS allows us to model system dynamics and extract track usage patterns using reward structures, identifying segments with high cumulative usage and earlier utilisation—crucial indicators for prioritising electrification.

The ABRS is an extension in which electrification is not predefined but synthesised: the model systematically explores all possible configurations and identifies those that ensure all scheduled trains reach their destinations with sufficient remaining charge. This forms a basis for future design automation and optimisation.

Additionally, track-wise battery level analysis was conducted by monitoring each train's charge as it traverses the network under two extremes: when all tracks are electrified and when none are.

- In the non-electrified case, this serves as a resilience baseline, confirming that trains retain adequate charge (e.g. 80%) even under worst-case scenarios.

– In the fully electrified case, the results support future planning, indicating how battery levels evolve across the network and where additional electrification would have the most impact.

These insights guide selective electrification, helping balance safety, operational efficiency, and infrastructure cost. Several directions remain for further development:

– Improved Temporal aspects and Movement Modelling: Enhancing temporal granularity to represent variable train speeds and segment-specific speed limits would enable modelling of diverse services (e.g. high-speed, intercity, regional trains). This would support exploring trade-offs between travel segments offering higher speeds/shorter charges vs. lower speeds/longer charges.
– Electrification Case Study with Real-World Data: We plan to apply our approach to a real-world rail network segment currently under electrification consideration. Future work will focus on automatically generating the initial state of the bigraph (i.e. its topology) from network maps as shown in [2]. By assigning realistic costs to each segment, we can perform multi-objective strategy synthesis to identify the lowest-cost configuration that guarantees safe operations—validating the model's practical applicability.
– Operational Disruptions and Delay Propagation: Extending the model to handle complex disruptions such as train delays and their propagation across routes would further test its flexibility and support robust, fault-tolerant rail system design.
– Faults and Probabilistic Repairs: Incorporating initial/dynamic faults and probabilistic repair mechanisms would enable analysis of recoverability and survivability, enhancing the model's ability to support realistic and resilient system design.

These extensions progressively align the model with the characteristics of a Digital Twin (DT) by integrating real-world data, operational variability, and fault dynamics—enabling high-fidelity simulation and decision support for real rail systems.

Acknowledgments. This work is supported by the Engineering and Physical Sciences Research Council, under grant EP/Z533221/1 (TransiT: Digital Twinning Research Hub for Decarbonising Transport) and an Amazon Research Award on Automated Reasoning.

References

1. Albalwe, M., Archibald, B., Sevegnani, M.: Modelling real-time systems with bigraphs. Electronic Proceedings in Theoretical Computer Science **417**, 96–116 (Mar 2025). https://doi.org/10.4204/eptcs.417.6
2. Ang, K.R.R.: Building bigraphs of the real world (2025), https://arxiv.org/abs/2508.00003

3. Archibald, B., Calder, M., Sevegnani, M.: Conditional bigraphs. In: Gadducci, F., Kehrer, T. (eds.) Graph Transformation - 13th International Conference, ICGT 2020, Held as Part of STAF 2020, Bergen, Norway, June 25-26, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12150, pp. 3–19. Springer (2020). https://doi.org/10.1007/978-3-030-51372-6_1
4. Archibald, B., Calder, M., Sevegnani, M.: Probabilistic bigraphs. Formal Aspects Comput. **34**(2), 1–27 (2022). https://doi.org/10.1145/3545180
5. Archibald, B., Calder, M., Sevegnani, M.: Practical modelling with bigraphs. Form. Asp. Comput. (Feb 2025). https://doi.org/10.1145/3721142
6. Clarke, E.M., Emerson, E.A.: Design and synthesis of synchronization skeletons using branching-time temporal logic. In: Kozen, D. (ed.) Logics of Programs, Workshop, Yorktown Heights, New York, USA, May 1981. Lecture Notes in Computer Science, vol. 131, pp. 52–71. Springer (1981). https://doi.org/10.1007/BFB0025774
7. Das, S., Almeida, R., Archibald, B., Sevegnani, M.: Formal analysis of resilience in transport systems with bigraphs. In: Safety/Reliability/Trustworthiness of Intelligent Transportation Systems. SAFECOMP 2025 Workshops - CoC3CPS, DECSoS, SASSUR, SENSEI, SafetyNXT, SCSSS, SRToITS and WAISE, Stockholm, Sweden, September 9, 2025, Proceedings. Lecture Notes in Computer Science, Springer (2025 - To Appear)
8. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Asp. Comput. **6**(5), 512–535 (1994)
9. Ingleby, M., Mitchell, I.: Proving safety of a railway signalling system incorporating geographic data. IFAC Proceedings Volumes **25**(30), 129–134 (1992). https://doi.org/https://doi.org/10.1016/S1474-6670(17)49419-5, iFAC Symposium on Safety of Computer Control Systems (SAFECOMP'92), Zürich, Switzerland, 28-30 October 1992
10. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) Proc. 23rd International Conference on Computer Aided Verification (CAV'11). Lecture Notes in Computer Science, vol. 6806, pp. 585–591. Springer (2011)
11. Kwiatkowska, M., Norman, G., Parker, D.: Stochastic Model Checking, pp. 220–270. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72522-0_6
12. Lu, C., Zou, Q., Zhou, J.: Toward a modeling and analysis method of cyber-physical systems architecture evolution based on bigraph. Sci Rep **15**(8766) (2025). https://doi.org/10.1017/S089006041900012X
13. Milner, R.: The Space and Motion of Communicating Agents. Cambridge University Press (2009)
14. Morley, M.J.: Safety-level communication in railway interlockings. Science of Computer Programming **29**(1), 147–170 (1997). https://doi.org/https://doi.org/10.1016/S0167-6423(96)00033-0, https://www.sciencedirect.com/science/article/pii/S0167642396000330, cOST 247, Verification and validation methods for formal descriptions
15. Sevegnani, M., Calder, M.: BigraphER: Rewriting and analysis engine for bigraphs. In: Chaudhuri, S., Farzan, A. (eds.) Computer Aided Verification - 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9780, pp. 494–501. Springer (2016). https://doi.org/10.1007/978-3-319-41540-6_27

16. Sevegnani, M., Kabac, M., Calder, M., McCann, J.: Modelling and verification of large-scale sensor network infrastructures. In: 2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS). pp. 71–81 (2018). https://doi.org/10.1109/ICECCS2018.2018.00016
17. Vykhodtsev, A.V., Jang, D., Wang, Q., Rosehart, W., Zareipour, H.: A review of modelling approaches to characterize lithium-ion battery energy storage systems in techno-economic analyses of power systems. Renewable and Sustainable Energy Reviews **166**, 112584 (2022). https://doi.org/https://doi.org/10.1016/j.rser.2022.112584
18. Wang, Y., Tian, J., Sun, Z., Wang, L., Xu, R., Li, M., Chen, Z.: A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems. Renewable and Sustainable Energy Reviews **131**, 110015 (2020). https://doi.org/https://doi.org/10.1016/j.rser.2020.110015