Better railway engineering through statistical model checking

Enno Ruijters and Mariëlle Stoelinga
University of Twente, EWI-FMT, P.O. Box 217, 7500 AE Enschede, The Netherlands
e-mail: {e.j.j.ruijters, m.i.a.stoelinga}@utwente.nl

Abstract. Maintenance is essential to ensuring the dependability of a technical system. Periodic inspections, repairs, and renewals can prevent failures and extend a system's lifespan. At the same time, maintenance incurs cost and planned downtime. It is therefore important to find a maintenance policy that balances cost and dependability.
This paper presents a framework, fault maintenance trees (FMTs), integrating maintenance into the industry-standard formalism of fault trees. By translating FMTs to priced timed automata and applying statistical model checking, we can obtain system dependability metrics such as system reliability and mean time to failure, as well as costs of maintenance and failures over time, for different maintenance policies.
Our framework is flexible and can be extended to include effects specific to the system being analysed. We demonstrate that our framework can be used in practice using two case studies from the railway industry: electrically insulated joints, and pneumatic compressors.

1 Introduction

In today's world, safety-critical systems are all around us. Complex systems like nuclear power plants, pacemakers, and trains have become essential to the operation of society, and failure of these systems can have disastrous consequentes. It is therefore important to analyse such systems to ensure that they meet dependability requirements.

In addition to safe design, proper maintenance is essential to keeping technological systems functioning. Few systems can remain operational for decades without any maintenance or repairs, and so this must be included in the safety analysis. Traditionally, this has been examined separately from the system design: First components manufacturers specify what maintenance is required, and what the reliability properties are if this maintenance is performed. Then, these properties are used to analyze whole-system dependability, thus assuming this maintenance is performed as specified.

A recent trend in asset management is towards reliability-centerd, a.k.a. risk based, maintenance [10]. This involves focussing maintenance efforts on the more critical components, while performing less maintenance on less important parts. Thus, better dependability can be achieved at lower cost. Planning such maintenance, however, requires knowledge of how maintenance at the component level affects whole-system dependability. We have developed a framework called *fault*

maintenance trees [11] combining maintenance and system design, which can be analysed using statistical model checking to obtain quantitative information about system dependability under different maintenance policies. This can then be used to find cost-optimal maintenance plans without compromising on safety.

Concretely, we combine the industry-standard for dependability analysis, fault trees, with maintenance models. The combined models are translated into timed automata, which can be analysed using the UPPAAL-SMC [6] model checking tool. We can obtain key performance indicators such as system reliability, expected number of failures over time, and expected costs. Our case studies for the railway industry show that this framework is suitable to maintenance policies found in practise, and yields the information necessary to optimize maintenance policies.

This paper first explains key information about maintenance and maintenance policies in Section 2. Next, Section 3 explains our framework of fault maintenance trees. Section 4 describes two case studies, and we conclude in Section 5.

1.1 Related work

The automata used in this work are based on the Input/Output Interactive Markov Chains used by the DFTCalc tool [2], which uses stochastic model checking to analyse dynamic fault trees without maintenance.

For an overview of a large number of analysis techniques and extensions for fault trees, we refer the reader to [12]. We mention some of the most closely related works here.

Bucci et al. [4] extend tradition fault trees with non-Markovian failure distributions and present a tool to analyse such FTs. This tool can be used to analyze components that wear out over time, but does not consider maintenance to undo this wear.

Buchacker et al. [5] present an alternative extension called Extended fault trees, to model systems where the failure rates of some components depend on the status of other components. This formalism does not include repairs or non-exponentially-distributed failure times, nor maintenance decisions based on the state of an entire subtree.

Aside from fault trees, numerous methods have been developed for the analysis and optimization of maintenance strategies. As this field is very broad, we refer the reader to surveys such as [1] for techniques based on simulation or [15] on o.a. analytic approximations and Bayesian reasoning.

One technique that is particularly close to our work is by Carnevali et al. [7] and examines the effect of maintenance in phased systems. Here resources are used by several tasks in a sequence, and in-between these tasks faults can be detected and repaired.

2 Maintenance

Most long-lived systems require some form of maintenance to avoid premature failures. From simple procedures like inflating your tires, to large overhauls of entire power plants, certain operations must be performed to keep a system in working order. This section explains how maintenance is typically performed in the railway industry.

2.1 Types of maintenance

Maintenance actions can be broadly divided into two categories: *preventive* and *corrective* maintenance. Preventive maintenance is performed before a system or component experiences a failure, where failure means that the system or component is no longer able to performs one of its intended functions. Corrective maintenance is performed after a failure has occurred, and is intended to restore the system or component to a functioning state.

Note that we consider component failures separate from system failures, as not all components are always necessary for the entire system to operate. For example, a datacenter with a redundant power supply can experience a failure of one power source while the datacenter as a whole remains fully functional. Section 3 explains fault trees, which are a formalism to describe how component failure combine to cause system failures.

An important aspect to preventive maintenance is the notion of *degradation* of components. Due to time and use, components typically degrade over time until they reach a point where a failure occurs. For example, the treads on a car's tires gradually decreases with use, until the tread is too worn to perform a necessary function of the tires, namely retain grip on the road when wet.

The choice of which type of maintenance depends on several factors, including the different costs of maintenance and failures, and the practical applicability of preventive maintenance. In systems where failures are much more expensive than maintenance, such as a nuclear power plant, preventive maintenance is almost always cost-beneficial. Conversely, when failures not more expensive than the planned downtime for maintenance, such as for your home lightbulbs, corrective maintenance is the better options. Some types of failures, such as lightning strikes, cannot be prevented by periodic maintenance.

2.2 Planning of maintenance

Besides what maintenance needs to be performed, it is important to decide when to do this maintenance. In general, we can distinguish three types of planning: *use-based*, *condition-based*, and *failure-based* maintenance [9].

Use-based maintenance is the simplest type of maintenance plan for preventive maintenance: It performs certain activities after some specified amount of use of the system, in whatever units of use are relevant. For example, changing the oil in a car can be performed after a given number of miles have been driven, or after a given length of time has elapsed.

Condition-based maintenance is more elaborate, as it specified the future maintenance plan given the current condition of the system. A simple example of such a plan is replacing the battery in your smoke-detector when it starts emitting low-battery beeps. Most condition-based maintenance plans also involve some use-based component, such as inspections at fixed times, since most systems do not measure their own condition well enough to completely on for maintenance.

Finally, failure-based maintenance is mostly used for corrective maintenance, where action is only taken when a failure occurs. This type of plan is not necessarily the same as only performing corrective maintenance, as it may involve repairing or replacing still-functional parts of a system as preventive maintenance against future failures.

3 Fault maintenance trees

One of the industry-standard methods for reliability analysis is fault tree analysis. A fault tree describes how failures at a component level interact to cause system failures. Our framework of *fault maintenance trees* extends fault trees by including maintenance. This section gives a brief overview of fault trees and fault maintenance trees.

3.1 Fault trees

Fault trees are directed acyclic graphs describing the combinations of component failures that lead to a system failures. The leaves of a fault tree, called *basic events* (BEs), denote the component failures. The internal nodes of the graph, called *gates* or *intermediate events*, describe the different ways that failures can interact to cause (sub)system failures. The root node of the graph is called the *top level event* (TE), and denotes the failure of the entire system.

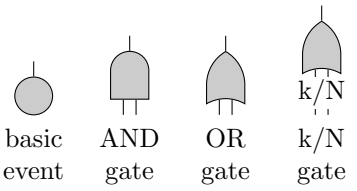


Fig. 1. Images of the elements in a standard fault tree

The gates in a fault tree can be of several types, shown in Figure 1, describing the various forms of interactions between failures. The AND-gate fails when all of its children fail, the OR-gate when any of its children fail, and the k/N-gate when at least k out of its N children fail.

These fault trees, also called *standard* or *static* fault trees, do not capture all possible interactions that can occur in practical systems, and many extensions have been developed to model more complex behaviours. An overview of such extensions can be found in [12].

To analyse the dependability of a system modeled using a fault tree and obtain quantitative values, the basic events must be decorated with numeric attributes describing their failure behaviour. The most common approach is to a

attach probabilities or *failure rate* to each basic event. Fault trees decorated with probabilities abstract away the evolution over time, while failure rates provides the parameters of exponential distributions governing the times when the events fail.

3.2 Metrics

Given an FT with failure information, several metrics of the system can be computed:

Reliability denotes the probability of the system failing within a given time window. Formally, if we describe the behaviour of the system described by a fault tree F using $X_F(t) = 1$ when this system has failed at time t, and 0 if it has not, the reliability is defined as $Re_F(t) = \mathbb{P}(X_F(t) = 0)$. Conversely, we use the term *unreliability* for the probability that the system has failed. For fault trees with only probabilities, the reliability is constant over time.

Availability denotes the expected fraction of time in a given time window that the system is functioning. Formally, we say $A_F(t) = \mathbb{E}(\frac{1}{t}\int_0^t X_F(x)\mathrm{d}x)$.

Mean time to failure denotes the expected time before a failure occurs. Formally, $MTTF(F) = \mathbb{E}(\mathrm{argmin}_t\, X_F(t) = 1)$.

Expected cost denotes the expected cost incurred within a given time frame. Although not very useful for FTs without maintenance, costs are very useful when comparing different maintenance strategies. Typically costs are incurred either on a per-event basis, e.g. a fixed cost to replace a broken component, or per unit time, e.g. lost productivity while a system is down. Formally, we write $C(t)$ for the cumulative cost incurred up to time t, hence the expected cost is either $\mathbb{E}(C(t))$ for a fixed time window, or $\frac{1}{t}\mathbb{E}(C(t))$ for the average cost per unit time.

3.3 Fault maintenance trees

The industry-standard approach to including repairs in a fault tree is to equip basic events with a *repair rate* as well as a failure rate [16]. This repair rate gives the parameter of an exponential distribution governing the time taken to repair the component after it has failed.

More complicated repair policies can be modeled using *repair boxes* described in [3] and [8]. While this approach supports complex policies for repairs after component failures, is does not allow for preventive maintenance, nor does it support the modeling of components with non-exponentially-distributed failure times.

We propose the formalism of *fault maintenance trees*, which supports complex preventive and corrective repair policies as well as components with arbitrary distributions of failure times. This formalism extends basic events by introducing *degraded states* (similar to those in extended fault trees [5]) in which the component is still functional but has worn to some extent. The tree structure is also augmented with *repair modules* and *inspection modules*, which act on the

extended basic events by returning them to a less degraded state, or initiating a repair depending on the current state.

We further introduce a new gate type, the *rate dependency* or RDEP, depicted in Figure 2. This gate describes a situation where the failure of one component (called the *trigger* causes the accelerated degradation one or more other components (called the *dependent children*. For example, if one pump in a redundant setup fails, the other pump is sufficient to keep the system functioning but the increased load results in faster wear of the functional pump. When the trigger is subsequently repaired, the dependent children return to the normal wear rate, but do not return to their original state of degradation.

3.4 Analysis through priced timed automata

To compute quantitative metrics of FMTs, such as reliability and availability, we translate the FMT into a network of priced timed automata (PTA), which we then analyze using the UPPAAL-SMC [6] model-checking tool.

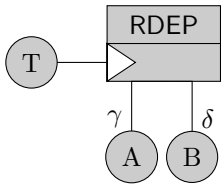


Fig. 2. RDEP gate where the failure of BE T leads to accelerated wear of BE A with a factor γ, and of BE B with factor δ.

A priced time automaton is a model consisting of locations and transitions between these locations. The locations represent states of the system, and transitions describe situations when the system may move from one state to another. Constraints on the edges and invariants on locations may be used to block or force certain transitions at certain times. These constraints and invariants are specified in terms of clocks, which increase linearly over time but may be reset when a transition is taken. Multiple PTAs can be combined using synchronisation on transitions, where some edges waiting for a signal *sig?* can only be taken simultaneous with a transition in another PTA emitting the corresponding signal *sig!*.

An example of a PTA can be seen in Figure 4, describing an inspection module (IM). The initial location is the one on the left. Here, the clock x denotes the time since the previous inspection, and increases until it is reset when an inspection is performed. The invariant on the initial location prevents the PTA from remaining in this state when the time to perform an inspection has been reached. Before this time, the guard on the self-looping transition prevents a premature inspection. When the clock x is equal to the time *interval*, the self-loop is taken and the clock is reset. The edge to the location on the right is a synchronization transition on the channel *thres*, and is taken when a component has degraded enough to take the corresponding transition its PTA. After this, the IM still waits for the inspection time, but the transition back to the initial location now also synchronizes with the repair module to begin a repair. Finally, both transitions corresponding to performing an inspection add a fixed amount *cost* to a global counter.

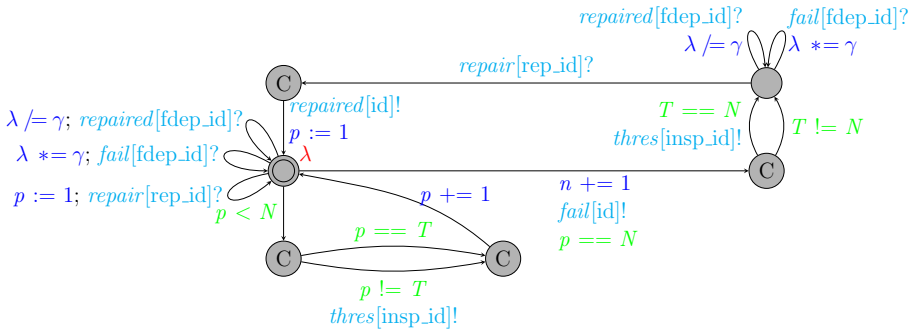


Fig. 3. PTA of a basic event with a failure time governed by a $(N,\ \lambda)$-Erlang distribution, with a threshold for the inspection at phase T. The counter p denotes the current phase, and is incremented according to exit rate of the initial state. If the current phase is equal to the threshold phase, a signal *thres*[insp_id] is send to the listening IM. When the current phase equals the number of phases N in the distribution, the PTA emits a signal *fail*[id] to all listening gates, possibly emits the threshold signal, and waits for a signal *repair*[red_id] from the RM. When this repair signal is received, the PTA emits a signal *repaired*[id] to any listening gates, reset the current phase to 1, and returns to the initial state. The signal *fail*[fdep_id] triggers an acceleration of the degradation due the the failure of an FDEP trigger, and *repaired*[fdep_id] return the rate to normal.

An addition to PTA for statistical model checking is the option to attach an *exit rate* to a location, which specifies an exponential distribution for the time that a transition is taken, unless an invariant forces a transition before this time.

To analyze an entire FMT, we convert each element (i.e. basic event, gate, IM, and RM) into a PTA, with appropriate synchronization depending on the structure of the FMT. Each element is assigned a unique ID to coordinate the signals for synchronization. The PTA for the basic event, IM, RM, and AND-gate are shown in Figures 3, 4, 5, and 6 respectively. The other gates are constructed analogously to the AND gate.

After converting an FMT into a network of PTA, the model-checking tool UPPAAL-SMC is used to compute quantitative metrics of the model. The different metrics described in Section 3.2 can be expressed in the tCTL-like logic of UPPAAL as follows, where x denotes a clock counting global time:

Reliability For convenience we describe only the unreliability, which is the probability of experiencing a failure within time t. If we denote the failed state of the top event as $T.Failed$, the unreliability corresponds to the formula $\mathbb{P}[x \leq t]\{\diamond T.Failed\}$.

Availability To compute the expected fraction of time the system is up, we introduce an auxiliary clock a that is stopped, but not reset, while the top event is in the failed state. The availability within time t can then be expressed as $\mathbb{E}[x \leq t]\{\max : a/t\}$.

Expected number of failures : To count the expected number of failures within a time bound, we introduce a variable that is incremented every time the top event enters its failed state, and use the formula $\mathbb{E}[x \leq t]\{\max : n\}$.

Expected cost : Our model tracks several variables corresponding to different costs (e.g. C_{total} for total costs, C_{insp} for the cost of inspections, etc.). To find the expected total cost of the system, we use the formula $\mathbb{E}[x \leq t]\{\max : C_{total}\}$. The other costs can be expressed by changing the counter.

4 Case studies

To demonstrate the practical applicability of fault maintenance trees and SMC in practice, we have applied this method to two cases from the railway industry: The electrically insulated railway joint (EI-Joint) [14], and a trainbound pneumatic compressor [13].

4.1 EI-joint

The electrically insulated joint is a component used to physically connect two railroad tracks while maintaining electrical separation between them. This is necessary since many train detection systems use electrical signals to determine whether a train is present, and such systems can only detect which isolated section of track the train is occupying. Failures of these joints are a major contributor to disruptions of train service.

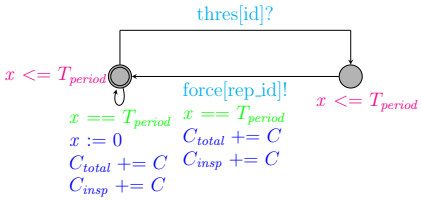


Fig. 4. PTA for an inspection module. The PTA begins in the leftmost state, and waits until either the time until the inspection interval (T_{period}) elapses, or until a threshold signal (thres[id]) is received from a BE. If the time elapses before a signal is received, then the inspection cost is incurred and the timer resets. If a threshold signal is received, the module waits for the scheduled inspection time, then signals its associated repair module to begin a repair (force[rep_id]), and then resets the timer.

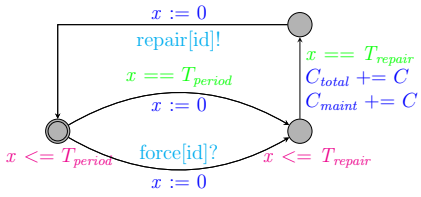


Fig. 5. PTA for a repair module. The PTA begins in the leftmost state with clock x initially zero. It waits until either the waiting time for a periodic repair (T_{period}) elapses, or a repair request signal (*force[id]*) is received. In either case, the module waits some time T_{repair}, incurs the C for a repair, sends a signal (*repair[id]*) to any BEs repaired by this module, and resets the timer.

A fault tree of the EI-joint can be seen in Figure 7, the exact failure modes are listed in Table 1. In broad terms, failures of the joints can be divided into mechanical failures where the physical connection between the rails is broken, and electrical failures where an electrical connection is made between the rails.

Aside from the distribution of the failure time, all failure modes have an associated 'condition' which is required for the failure to occur. For example, a joint can be short-circuited by metal shavings when the wheels of a train scrape against the track, which only occurs in joints installed where the track curves. We model this by a probability in each basic event, corresponding to the fraction of all joints that are susceptible to the particular failure mode.

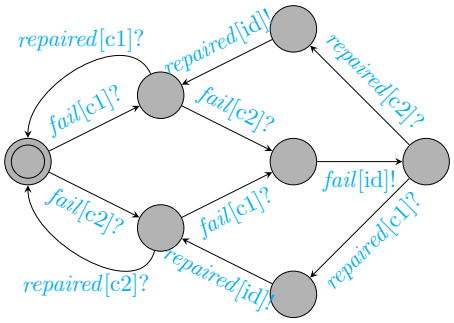


Fig. 6. PTA of an AND gate with two children with IDs c1 and c2. The gate listens for the failure signals of its children, and emits its own failure signal when both children have failed. Likewise, it listens for the signals that its children have been repaired, and emits its own repaired signal if either child is repaired after the gate emits its failure signal.

The maintenance policy for the joints is fairly straightforward: Periodic visual inspections are performed, and any problems found are corrected shortly after. Some failures, such as when a conductive path is formed by iron shavings, can be easily corrected, e.g. by sweeping away the shavings. Mechanical defects and faults internal to the joint can only be repaired by replacing the entire joint. We leave out exact details of the current policy for reasons of confidentiality.

The goal of this case study was to improve the current maintenance policy with respect to cost. We consider three categories of costs: (1) costs of inspections, (2) costs of maintenance (preventive and corrective), and (3) costs of failures. These costs were provided by ProRail, as an average over all the joints they maintain (as the actual costs vary, e.g. a failure in a high-traffic rail is more expensive than in some rarely-used sidetrack). The costs of failure include both monetary cost, and a model of the social costs of unavailability based on passenger delays.

Results. We have analyzed the model of the EI-joint, both under the current maintenance policy and under several possible improvements. In general, we find that our results for the current policy closely match historical records of failures, indicating that the model is a good representation of the actual system. we find that the current policy is close to optimal, and that this optimum is fairly insensitive to small variations in inspection frequency.

The results shown in this section are computed using 40,000 simulation runs, resulting in a 95% confidence interval with a width less than 1% of the indicated

BE nr.	Failure mode	Parameters			Failure rates		
		ETTF (yrs)	Phases	Prob. cnd.	Predicted	Actual	Difference
1	Poor geometry	5	4	10%	110	48	62
2	Broken fishplate	8	4	33%	129	83	46
3	Broken bolts	15	4	33%	2.3	2.1	0.2
4	Rail head broken out	10	4	33%	68	30	38
5	Glue connection broken	10	4	33%	70	37	33
5a	Manufacturing defect	-	-	0.25%			
5b	Installation error	-	-	0.25%			
6	Battered head	20	4	5%	3.4	5.5	2.1
7	Arc damage	5	3	0.2%	7	3.4	3.6
8	End post broken out	7	3	33%	12	9.4	2.6
9	Joint bypassed: overhang	5	4	100%	212	200	12
10a	Joint shorted: shavings (normal)	1	4	12%			
10b	Joint shorted: shavings (coated)	10	4	3%			
10	Joint shorted: sharings (total)				156	150	6
11	Joint shorted: splinters	200	1	100%	254	261	7
12	Joint shorted: foreign object	250	1	100%	199	200	1
13	Joint shorted: shavings (grinding)	5000	1	100%	10	10	0
14	Sleeper shifted	5000	1	100%	19	18	1
15	Internal insulation failure	5000	1	100%			

Table 1. Parameters and results of the basic events of the FMT for the EI-joint. The column 'ETTF' lists the expected time to failure, assuming no maintenance is performed. The column 'prob. cnd.' gives the probability that a given joint is subject to the condition that allows this failure mode to occur. The last three columns give the number of failures per year in a population of 50,000 joints as predicted by the model and as observed in practice. Modes 5a and 5b have a fixed probability of occurring every time a joint is installed. Failure data for mode 15 was not available, and therefore not included in the analysis.

values. When comparing to historical records, we consider the entire population of 50,000 EI-joints in the Netherlands.

In detail, Table 1 also shows the predicted and actual number of occurrences of the various failure modes. We observe that most failure modes occur about as frequently as predicted. Furthermore, we consider the total number of failures each year, which the model predicts at 3680 replacements per year, while historical records indicate approx. 3000 joints are purchased. We expect that this difference is due to some failure modes being modeled as needing a complete joint replacement, but which can be repaired by minor maintenance if the degradation has not progressed very far.

Figure 8 shows a breakdown of the costs of a joint over a 50-year timespan. We note that the costs increase almost linearly, and thus we do not need to consider a specific time bound when evaluating the annual cost of a maintenance policy.

We now consider alternative maintenance policies where we vary the inspection frequency. The results of this analysis are shown in Figure 9. As expected, the cost of inspections increases linearly with frequency and the cost of failures decreases but with diminishing returns. The cost for maintenance is nearly constant, as the inspection will only determine whether a repair action is performed before or after failure, but does not change the number of needed repairs.

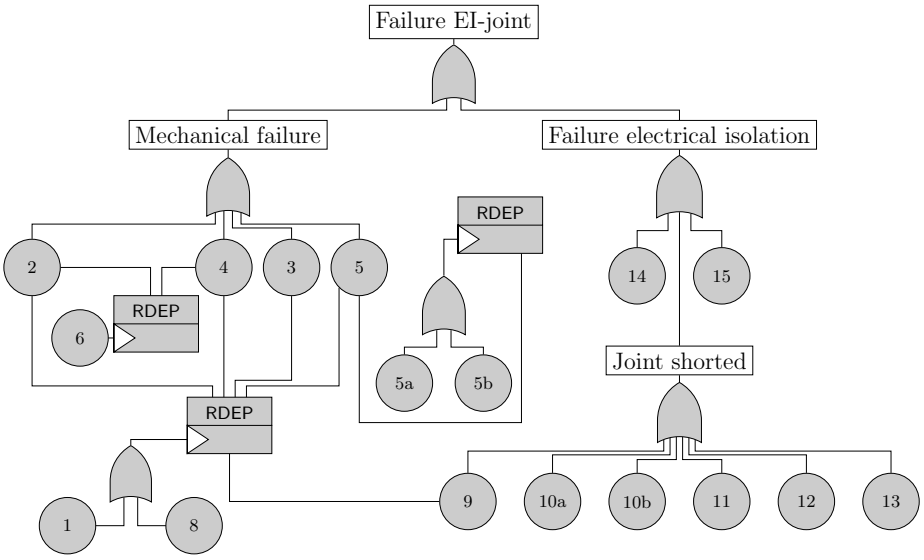


Fig. 7. Fault tree of the electrically insulated railway joint. The numbers in the basic events correspond to those in Table 1.

We find that the optimal inspection frequency is approx. four inspections per year, but the total cost is nearly constant between two and six inspections per year. The current policy lies within this range, so it is as optimal as our model can predict.

Next, we examine several qualitative changes to the maintenance policy. The three changes are: (1) always replacing the entire joint rather than correcting single defects, (2) reducing the threshold for when corrective action is taken after an inspection, and (3) periodically replace the entire joint after a given time rather than waiting for its condition to deteriorate. The results of these policies are shown in Table 2.

We observe that periodic replacements have only a small impact on the failure frequency but incur significant costs, and are thus not useful. Replacing whenever a defect is found is more productive, but still prohibitively expensive. Finally, the reduced threshold cuts the number of failures nearly in half for only a small increase in total cost. Nonetheless, since the total cost includes the social costs of the failures, we do not expect this policy to be an improvement overall.

4.2 Pneumatic compressor

Our second case study concerns the pneumatic compressor used in a Dutch trains of the VIRM type. Each train has one such compressor, which provides compressed air for the operation of the brakes, automatic doors, etc. The systems that operate on this compressed air are designed to be fail-safe (e.g. the

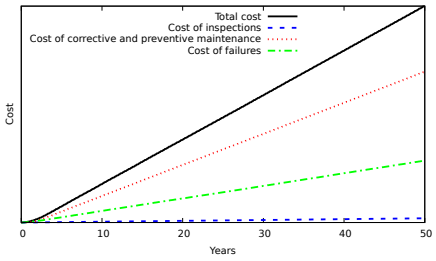


Fig. 8. Cumulative costs of one EI-joint over time, split up by type of cost.

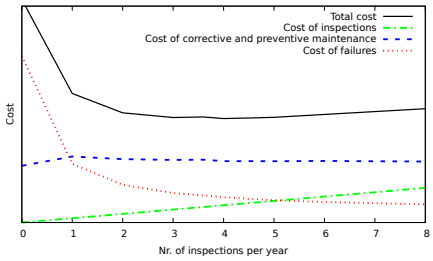


Fig. 9. Different types of total costs for one joint, depending on the inspection frequency.

Policy	Maint. cost	Total cost	Failure frequency
Current	1	1	1
Replace instead of repair	2.20	1.65	0.76
Reduce threshold by $\frac{1}{3}$	1.49	1.16	0.48
Replace every 5 yrs.	2.49	1.85	0.88
Replace every 10 yrs.	1.59	1.34	0.96
Replace every 20 yrs.	1.30	1.17	0.97

Table 2. Comparison of the effects of different maintenance policies, relative to the current policy.

brakes are automatically applied when air pressure drops), but a failure of the compressor leaves the train stranded resulting in delays for the passengers.

The model of this compressor was developed in cooperation with NedTrain, the company responsible for maintenance of Dutch trains, among others. For reasons of confidentiality, all times in this section (e.g. failure rates, inspection intervals, etc.) are scaled by a constant factor.

Figure 10 shows the fault tree for the compressor, and the exact failure modes are listed in Table 3.

The compressor has a more complex maintenance policy than the EI-joint, with different kinds of inspections and repairs. The policy consists of (1) inspections and minor corrective repairs every two days, (2) more involved check-ups and preventine maintenance every three months, (3) a minor overhaul every three years, and (4) a major overhaul every six years after which the compressor is considered as good as new.

Results We again consider various alternative maintenance policies, the results of which are shown in Figure 11. We notice that removing the overhauls has very little effect on the failure rate, which leads us to question their cost-effectiveness (with the caveat that degradation behaviour past the 6-year overhaul time is not known, so nonlinear effects such as metal fatigue may cause a large increase in

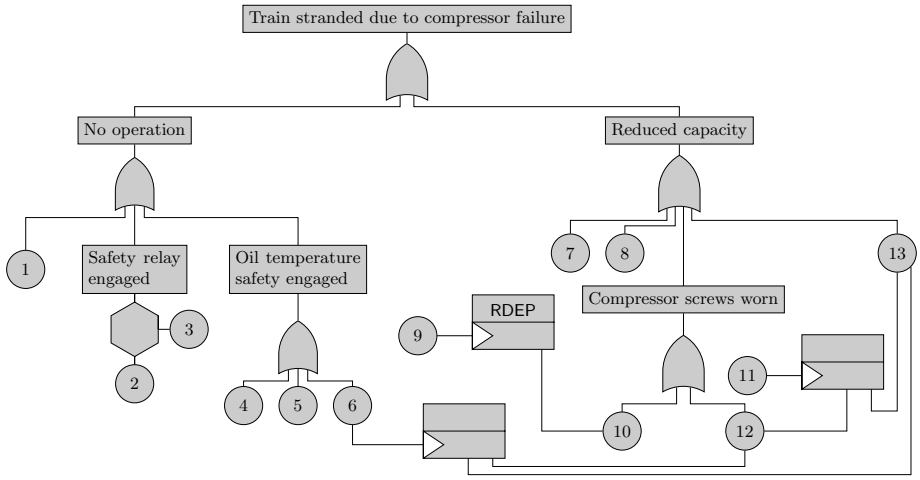


Fig. 10. Fault tree of the pneumatic compressor. The numbers in the basic events correspond to the failure modes in Table 3.

failure rate). Changing the service interval does have a substantial effect, indicating that this is an important parameter when deciding the policy. Unfortunately, since we do not have information on costs, we cannot show what frequency would be optimal.

5 Conclusion

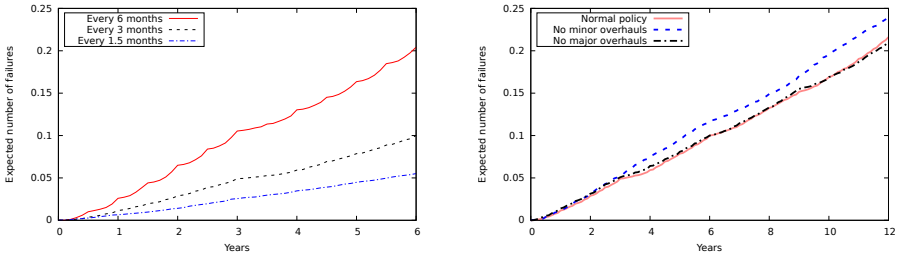
This paper presents the framework of fault maintenance trees, integrating maintenance into fault trees to analyze the dependability of systems under different maintenance regimes.

We have shown how these fault maintenance trees can be analyzed by converting them into priced timed automata and applying statistical model checking. This analysis yields quantitative metrics such as system reliability and expected cost, which can be used to find optimal maintenance strategies.

Two case studies from the railway industry demonstrate that this framework is applicable in practice, and

Nr.	Failure mode	Nr. of phases	ETTF
1	Motor does not start when asked	3	16.6
2	De-aeration valve defective	3	200
3	Two starts in short time	2	0.001
4	Radiator obstructed	4	5.5
5	Oil thermostat defective	3	16.6
6	Low oil level	4	5.5
7	Pressure valve leakage	3	3.3
8	Air filter obstructed	2	500
9	Degraded air filter	4	5
10	Particle-induced rupture	4	120
11	Oil pollution	4	5.5
12	Lubrication-induced wear	4	120
13	Motor/bearings degraded	4	120

Table 3. Parameters of the failure modes of the compressor. The values have been scaled for anonymity.



(a) Effect of different frequencies of the small service.

(b) Effect of the minor and major overhauls.

Fig. 11. Expected number of failures for variations on the maintenance policy of the pneumatic compressor.

yields results that can be used in decision-making to reduce expenses and improve system dependability.

Acknowledgements

This work has been supported by the STW-ProRail partnership program ExploRail under the project ArRangeer (122238) with participation by Movares.

References

1. A. Alrabghi and A. Tiwari. State of the art in simulation-based optimisation for maintenance systems. *Computers & Industrial Engineering*, 82:167 – 182, 2015.
2. F. Arnold, A. Belinfante, F. van der Berg, D. Guck, and M. Stoelinga. DFTCalc: A tool for efficient fault tree analysis. In *Proc. 32nd Int. Conf. on Computer Safety, Reliability and Security (SAFECOMP)*, volume 8153 of *LNCS*, pages 293–301, Toulouse, Fance, 2013.
3. A. Bobbio and D. Codetta-Raiteri. Parametric fault trees with dynamic gates and repair boxes. In *Proc. Reliability and Maintainability Symp.*, pages 459–465, 2004.
4. G. Bucci, L. Carnevali, and E. Vicario. A tool supporting evaluation of non-markovian fault trees. In *Proc. 5th Int. Conf. on Quantitative Evaluation of Systems (QEST)*, pages 115–116, September 2008.
5. K. Buchacker. Modeling with extended fault trees. In *Proc. 5th IEEE Int. Symp. High Assurance Systems Engineering (HASE)*, pages 238–246, 2000.
6. P. Bulychev, A. David, K. G. Larsen, M. Mikučionis, D. B. Poulsen, A. Legay, and Z. Wang. UPPAAL-SMC: Statistical model checking for priced timed automata. In *Proc. 10th workshop on Quantitative Aspects of Programming Languages (QAPL 2012)*, 2012.
7. L. Carnevali, M. Paolieri, K. Tadano, and E. Vicario. Towards the quantitative evaluation of phased maintenance procedures using non-markovian regenerative analysis. In *Proc. 10th European Performance Engineering Workshop (EPEW)*, volume 8168 of *LNCS*, pages 176–190, September 2013.

8. Daniele Codetta-Raiteri, Giuliana Franceschinis, Mauro Iacono, and Valeria Vittorini. Repairable fault tree for the automatic evaluation of repair policies. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, pages 659–668. IEEE, 2004.
9. C.W. Gits. Design of maintenance conceps. *International Journal of Production Economics*, 24(3):217–226, March 1992.
10. J. Moubray. *Reliability centered maintenance*. Industrial Press, 1997.
11. E. Ruijters, D. Guck, P. Drolenga, and M. Stoelinga. Fault maintenance trees: reliability centered maintenance via statistical model checking. In *Proc. Reliability and Maintainability Symp.*, January 2016.
12. E. Ruijters and M. Stoelinga. Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. *Computer Science Review*, 15–16:29–62, 2015.
13. Enno Ruijters, Dennis Guck, Peter Drolenga, Margot Peters, and Mariëlle Stoelinga. Maintenance analysis and optimization via statistical model checking: Evaluating a train pneumatic compressor. In *Proc. Int. Conf. Quantitative Evaluation of SysTems (QEST)*, 2016. Submitted for publication.
14. Enno Ruijters, Dennis Guck, Martijn van Noort, and Mariëlle Stoelinga. Reliability-centered maintenance of the electrically insulated railway joint via fault tree analysis: A practical experience report. In *Proc. Int. Symp. Dependable Systems and Networks (DSN)*, 2016. Accepted for publication.
15. A. Sharma, G. S. Yadava, and S. G. Deshmukh. A literature review and future perspectives on maintenance optimization. *Journal of Quality in Maintenance Engineering*, 17(1):5–25, 2011.
16. W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl. *Fault Tree Handbook*. Office of Nuclear Regulatory Reasearch, U.S. Nuclear Regulatory Commision, 1981.