Modelling Smart Buildings Using Fault Maintenance Trees

Alessandro Abate[2], Carlos E. Budde[1(✉)], Nathalie Cauchi[2], Arnaud van Harmelen[1], Khaza Anuarul Hoque[4], and Mariëlle Stoelinga[1,3]

[1] Formal Methods and Tools Group, University of Twente, Enschede, The Netherlands
{c.e.budde,a.vanharmelen,m.i.a.stoelinga}@utwente.nl
[2] Department of Computer Science, University of Oxford, Oxford, UK
{alessandro.abate,nathalie.cauchi}@cs.ox.ac.uk
[3] Department of Software Science, Radboud University, Nijmegen, The Netherlands
[4] Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, USA
hoquek@missouri.edu

Abstract. Increasingly many industrial spheres are enforced by law to satisfy strict RAMS requirements—reliability, availability, maintainability, and safety. Applied to Fault Maintenance Trees (FMTs), formal methods offer flexible and trustworthy techniques to quantify the resilience of (abstract models of) systems. However, the estimated metrics are relevant only as far as the model reflects the actual system: Refining an abstract model to reduce the gap with reality is crucial for the usefulness of the results. In this work, we take a practical approach at the challenge by studying a Heating, Ventilation and Air-Conditioning unit (HVAC), ubiquitous in smart buildings. Using probabilistic and statistical model checking, we assess RAMS metrics of a basic fault maintenance tree HVAC model. We then implement four modifications augmenting the expressivity of the FMT model, and show that reliability, availability, expected number of failures, and costs, can vary by orders of magnitude depending on involved modelling details.

Keywords: Fault maintenance trees · Reliability · Availability
Maintenance · Model checking · PMC · SMC · Smart buildings · HVAC

1 Introduction

The current rapid momentum in the number of available sensing devices and the advances in communication technologies has resulted in a growing interest towards making things "smart." This shift has not escaped the building sector, where engineers and researchers are working towards a new type of building termed *smart buildings*. These are equipped with sensors to deliver services that are cost effective, compliant with RAMS—reliability, availability, maintainability, and safety—requirements, ubiquitous, and ensuring occupant comfort and

R. Bakhshi et al. (Eds.): EPEW 2018, LNCS 11178, pp. 110–125, 2018.
https://doi.org/10.1007/978-3-030-02227-3_8

productivity, e.g. proper temperature and high air quality. A key element is the correct application of timely and cost-effective maintenance: Comfort and correct building operation, i.e. reliable and dependable, can be maintained only as long as the components are available and operating with sufficient performance.

In this work we focus on the Heating, Ventilation and Air-Conditioning unit (HVAC) of a smart building, whose optimised operation is essential for the correct running of the premises. Early fault detection and maintenance can improve the lifespan and reliability of an HVAC. In the literature, maintenance can be optimised following different methods—see e.g. [17]. *Fault maintenance trees* (FMT, [19]) are a novel technique to model and analyse systems, which allow planning maintenance strategies to balance costs and system (failure) resilience. FMTs extend dynamic fault trees (DFT, [11]) encompassing degradation and maintenance concepts. Degradation modelling represents component health decay via elemental modules known as Extended Basic Events. Maintenance modelling incorporates different maintenance concepts like inspections, repairs, and replacements. Typically, FMT analysis is performed via statistical model checking (SMC, [19]). Analysing (smart buildings using) FMTs via probabilistic model checking (PMC) was introduced in [5]. In that work, component degradation of an HVAC is discretised in time using phases, with a stepwise degradation behaviour approximated via Erlang distributions, and using inspection and repair modules to regulate maintenance actions.

Standing on the FMT model framework introduced in [5], in the following sections we present a sequence of modelling setups which extend the central case study of that work. We enhance the modelling and analysis of the HVAC FMT by adding realistic flavours, to attune the maintenance policies towards their application in the real world. To that aim, we first perform an encoding of the FMT in terms of continuous-time Markov chains and priced time automata, which we then respectively analyse using PMC and SMC. For each technique we highlight the trade-offs and limitations encountered. From that basis, we extend the FMT model in four stages: First, we individualise maintenance actions and make a clear distinction between *cleans* and *repairs*. Second, we drop the Erlang approximation of time periods in lieu of *truly deterministic intervals*. Third, we model component redundancy by introducing *spare gates* for some elements of the HVAC. Fourth and last, we refine the degradation of some extended basic events to follow a *continuous stochastic (generalised) behaviour*.

We use both PMC and SMC to analyse the first two modelling setups, i.e. the basic setting and the first extension; for all other extensions we use only SMC. PMC explores all states of the model (relevant for the current property query) and does not need statistical bounds to decide convergence. In contrast, SMC uses statistical theory to infer conclusions with arbitrary levels of confidence and precision. On each stage we demonstrate the implications and the resulting modifications needed for analysing system reliability, availability, expected number of failures, and the total costs. We also delineate the impact on these key performance indicators w.r.t. the previous models.

This article has the following structure: Sect. 2 presents the fundamental theoretical basis; Sect. 3 introduces the central case study, an HVAC unit, where the root HVAC model inspired in [5] is presented in Sect. 3.1; The four modelling extensions are introduced and analysed in Sects. 3.2, 3.3, 3.4 and 3.5; Sect. 4 concludes this work and outlines possible tracks of future research.

2 Preliminaries

Fault Maintenance Trees. *Fault trees* are directed acyclic graphs describing combinations of failures in system components, that can lead to a system failure or *Top Level Event* (TLE) at the root of the tree [21]. The leaves in fault trees are *basic events* which denote atomic component failures, typically following the exponential distribution. The internal nodes or *gates* describe how failures in basic events and lower level gates interact, as they propagate towards the TLE. The internal events are labelled as *intermediate events* (IE). Each gate models a different interaction: to propagate a failure, AND gates require all children to fail, OR gates require any child to fail, etc. [20]. In standard fault trees a closed-form solution exists for many RAMS metrics, provided the exact distributions of the basic events are known. *Dynamic fault trees* introduce gates with time- or order-dependent behaviour for which this is no longer true [22]. For instance, the children in priority-AND gates are ordered, and the gate fails if all children fail from left to right. Other than FMTs, there seems to be relatively scarce literature on DFTs that support component health decay combined with preventive maintenance, e.g. acting *before component failure* [20]. In [13] a tool is presented to compute RAMS metrics from DFTs in the presence of a maintenance policy. FMTs offer a formalism for this: They are a superset of DFTs which can model and assess various maintenance concepts [19]. Extensions over DFTs include:

- *Extended Basic Events* (EBEs): basic events whose failures follow an Erlang distribution. Its stepwise degradation allows identifying light decay, allowing restorations before an actual failure (that may trigger a TLE) occurs;
- *Repair Modules* (RM): perform periodic checks that can trigger maintenance actions. This encompasses with the phased degradation of EBEs, allowing early detection of degradation and potentially cheaper maintenance, as opposed to repair boxes that can repair a component only after it has failed.

Metrics. To measure compliance with RAMS requirements, it is common to set a time horizon $T > 0$ and quantify failures in the time window $[0, T]$. Maintenance is also a cost-driven concept, hence the operational and maintenance costs incurred within the time window provide further insight on how well the system is performing. The following *Key Performance Indicators* (KPIs) are commonly used to assess system resilience in the presence of maintenance actions:

- *Reliability* at time T is the probability of not observing a system failure, i.e. a TLE, in the time window $[0, T]$;
- *Availability* at time T is the proportion of time that the system is not failed in the time window $[0, T]$;
- *Expected number of failures* (ENF) at time T is the expected number of times a TLE is observed in the time window $[0, T]$;
- *Expected cost* at time T is the total expected cost incurred in the time window, including operational and maintenance costs (such as costs associated with system inspections and repair of components).

Modelling and Analysis of FMTs. FMTs can be given semantics via Bayesian networks, generalised stochastic Petri nets, etc. [14]. We use continuous-time Markov chains (CTMC) and priced time automata (PTA): two widely extended modelling formalisms with rich tool support, whose expressiveness meets our modelling requirements. For these semantics, the KPIs of interest can be quantified via *quantitative model checking* [7], a well-established formal verification technique used to verify the correctness of finite-state automata. Model checking algorithms take as input (i) a formal model of the system, usually some type of labelled automaton, and (ii) the property queries to verify, usually expressed in terms of a temporal logic. To check whether the model satisfies a property, the algorithms explore exhaustively and automatically all (reachable) states. Quantitative model checking is a broad field that comes in different flavours [8]. In particular we use two (complementary) techniques to analyse our models:

- *Probabilistic model checking* (PMC, [15]) performs a state space analysis in probabilistic (finite) state automata. These are usually state transition models like CTMC, with probability as transitions rates and labels on the transitions and the states. A *probabilistic model checker* computes the probabilities of reaching certain states, or the expected reward over a time horizon.
- *Statistical model checking* (SMC, [24]) samples finitely many runs of "model behaviour," typically execution traces, and uses statistical analysis to estimate an answer to the query from such (random) sample, where the probability of converging to an incorrect answer can be arbitrarily bounded.

Thus, using PMC/SMC one can analyse e.g. CTMC/PTA models of an FMT, computing (approximate) values for the relevant KPIs, which serve to assess the resilience and RAMS compliance of the modelled system.

3 Fault Maintenance Tree Model of an HVAC System

This work is centred around a Heating, Ventilation and Air-Conditioning unit (HVAC) that regulates the internal environment in smart buildings. HVACs offer a decomposition in subsystems fitting nicely the FMT approach. The concrete model studied is taken from [5]: Fig. 1 shows a visual description of the setup.

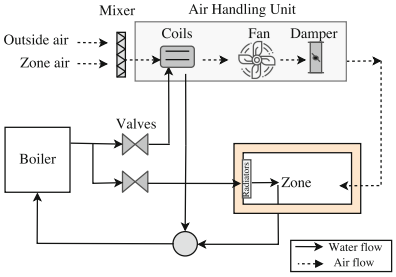


Fig. 1. HVAC system schematic [5].

The HVAC is divided in two circuits, one for air and the other for water flow. Two valves in the water circuit, one for the supply air heating coil and one for the radiators, control the water flow rate. A boiler heats up the supply water, which is then transferred into the heating coil and the radiators. The radiators transfer the water heat directly into the room (or *zone*). The return water goes through the collector back towards the boiler. In the air circuit, the mixer blends outside air with zone air. This goes to the heating coil to warm it up to the desired temperature. The air is then sent back into the zone via the supply fan, at a rate controlled by the Air Handling Unit dampers (AHU).

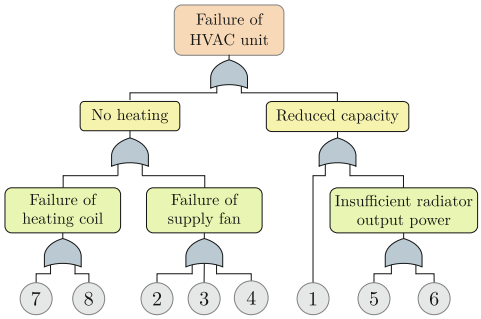
In smart buildings, comfort and running costs depend heavily on the proper functioning of the HVAC unit. Moreover, these are complex machines that can fail in various ways, and repairs can be quite costly. The trade-off between system performance and maintenance costs offers a rich scenario to model with FMTs, and to analyse with model checking in order to estimate relevant KPI metrics.

The degree of confidence in, and utility of, the computed metrics is a function of the realism of the underlying model. With that motivation, the next sections present five (incremental) versions of the HVAC, using FMT models to measure KPIs. We start from the basic case in Sect. 3.1, which mainly corresponds to the model presented in [5]. In Sect. 3.2 we enhance the model by refining the maintenance actions. In those two setups deterministic time delays are emulated via Erlang distributions: In Sect. 3.3 we use true deterministic delays instead. Finally, in Sect. 3.4 we introduce component redundancy by means of spare gates, and in Sect. 3.5 we model EBE degradation using continuous stochastic functions.

3.1 HVAC-0: The Basic Setting

In [5], HVAC failures can derive from malfunctions in the heating coil, the supply fan, or the radiators. Similarly, here we decompose the HVAC FMT into the failures affecting its subsystems; see Fig. 2a for a graphical description. The leaves of the tree are EBEs whose degradation behaviour is detailed in Fig. 2b. Values for N and MTTF, which are the number of degradation phases and mean time to failure respectively, are obtained from [2,12] and are based on a real dataset of measurements on an HVAC system. For instance, EBE 1 models the failure of the AHU via a random variable with distribution Erlang$(4, 4/20)$.

We label the degradation phases (states) of EBEs to allow differentiated maintenance actions. With `new` we label the initial phase of an EBE, corresponding to perfect condition. With `failed` we label the last phase, e.g. phase 4 for EBE 1, corresponding to a failure that may propagate in the tree. With `thresh` we label all other phases to indicate a degraded—but still functional—condition.



(a) FMT leaves are EBEs, IEs reflect the subsystem affected by failures in the children.

EBE ID	Failure mode	N	MTTF (years)
1	AHU damper broken	4	20
2	Fan motor failure	3	35
3	Supply fan obstructed	4	31
4	Fan bearing failure	6	17
5	Radiator failure	4	25
6	Radiator stuck valve	2	10
7	Heater stuck valve	2	10
8	Heat pump failure	4	20

(b) Detailed EBEs; degradation values from $[2, 12]$.

Fig. 2. FMT of HVAC-0

The maintenance policies modelled in [5] distinguish between *inspections*, *repair checks*, and *overhauls*, which in our setting take place every half, two, and fifteen years respectively. Deterministic time delays, e.g. for performing these periodic maintenance checks, are emulated via Erlang distributions. Overhauls trigger a *replace* action that renews the whole HVAC, sending all EBEs back to their states representing the `new` phase. Replace actions take one week to complete. Instead, inspections and repair checks can trigger a *clean* action, that reverts one degradation phase in all the EBEs. Clean actions take one day to complete. When an inspection takes place, a clean is triggered if any EBE is in a `thresh` state. Similarly, a clean is triggered if any EBE is in a `thresh` or `failed` state during a repair check. Maintenance actions act on all EBEs: A clean sends all EBEs back one degradation phase—except those in a `new` state. Notice these semantics are a modelling choice and *not* a general characteristic of FMTs.

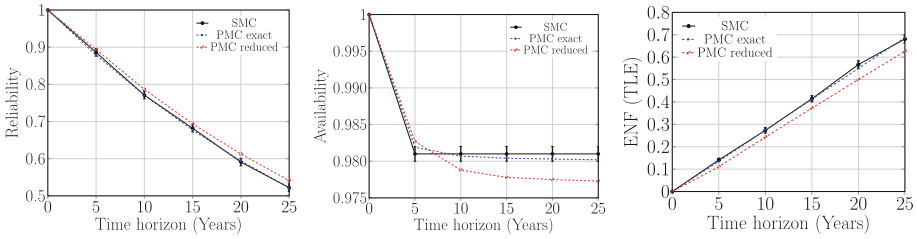
The total costs incurred are divided into operational and maintenance costs. Operational costs accrue €1 per day of system uptime and €4 per day of system downtime. Maintenance costs are €5000 per replace action triggered, €100 per clean action triggered, and €5 per periodic inspection. Repair checks and overhauls incur no additional costs when they take place. These values are based on previous research and expert-knowledge applied to an industrial case study [6].

In [5], the HVAC FMT is modelled using a CTMC with rewards, and the KPIs are computed with PMC via the PRISM model checker [16]. A state-space reduction technique is devised to build "an equivalent abstract CTMC," allowing PRISM to analyse the whole model and estimate (an approximation of) the metrics. We reproduce that approach for HVAC-0, and extend the analysis repertoire with SMC via the UPPAAL tool [9]. SMC estimates confidence intervals rather than point values like PMC does. Once a confidence level and termination epsilon have been set for SMC and PMC respectively, the results yielded by these techniques coincide if the SMC interval contains the PMC value.

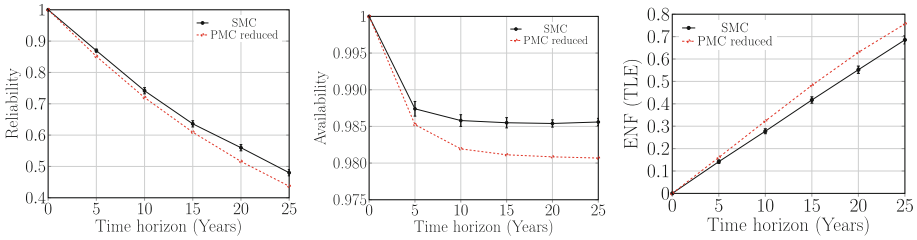
UPPAAL operates with PTAs, a proper superset of CTMC that can encode (general) stochastic and non-linear dynamic behaviour. To substantiate the

semantic coincidence of the models encoded in both tools, we first study subsystems of HVAC-0 for which the non-reduced (*exact*) PRISM model (i.e. without the state-space reduction technique devised in [5]) can be analysed.

In Fig. 3a we show the metrics for five time horizons in the largest of these subsystems, where only EBEs 2, 3, and 4 (i.e. the supply fan subsystem) are missing from the model of Fig. 2a. The metrics coincide between SMC and PMC exact, and differ slightly (as expected) for PMC *reduced*, i.e. using the abstract CTMC. When studying the full system, PMC exact cannot be used due to the state space explosion and the physical memory constraints [5,15]. Thus only SMC and PMC reduced can be compared, for which a difference as that observed in Fig. 3a is expected. This is corroborated in the full system analysis of HVAC-0, as it can be observed in Fig. 3b. The metrics for the total costs are not shown due to space constrains but they exhibit the same trends.



(a) Metrics for HVAC-0 (no supply fan): SMC, PMC exact and PMC reduced



(b) Metrics for HVAC-0 (full system): SMC and PMC reduced

Fig. 3. Comparative model checking for FMT of HVAC-0

In Fig. 3a, reliability and ENF values for SMC are lower than for PMC reduced, whereas in Fig. 3b they are higher. This is due to an interplay between the cleaning actions and the state-space reduction, which (for each time horizon) substitutes OR gates by EBEs with 4 phases and the MTTF of the replaced subsystem. Only EBE 4 has more than 4 phases—see Fig. 2b. Thus in PMC reduced *the number of phases* of all replaced subtrees is greater or equal for the system of Fig. 3a, but *lower* for the full HVAC-0 of Fig. 3b, viz. for the supply fan subtree. Therefore, only for PMC reduced of the full system, cleaning actions have less opportunities to act, which derives in more failures and explains Fig. 3.

3.2 HVAC-1: Refinement of Maintenance Actions

In the basic setting of Sect. 3.1, inspections and repair checks overlap considerably: Both can trigger the same maintenance action, namely a clean, and both will do so in the same system configurations. The only situations when a clean would be triggered by a repair check and not by an inspection, is when there is at least one failed—but no degraded—component. The likelihood of these scenarios decreases with the amount of EBEs and their number of degradation phases.

A more problematic modelling effect is that, when triggered by an inspection, a clean can "repair" a failed EBE and make it operational again. If e.g. EBE 1 is failed and EBE 2 is degraded, an inspection will trigger a clean because EBE 2 is in a `thresh` state; since cleaning actions are system-wide this also affects EBE 1, which then moves from its `failed` to a `thresh` state, becoming operational.

We argue this is not a realistic behaviour: Thus as first improvement over HVAC-0 we propose a more clear distinction between inspections and repairs. The former will remain as is, but repair checks will effectively trigger a *repair* maintenance action iff some EBE is in its `failed` state. As opposed to a clean triggered by an inspection, a repair *will only affect failed components, sending them back* $N - 2$ *degradation phases*. In particular, EBEs in a `thresh` or `new` state are not affected by repair actions. Repairs thus restore the health of failed components significantly albeit not entirely—only replacements triggered by overhauls leave components "as good as new." Repairs are necessarily more complex than cleans: They take longer (2 days) and cost more (€800).

The intuition behind this modification, code name HVAC-1, is that a technician fixes periodically all broken components, which has been named age repair or *age replacement* and is also related to *block replacement* [1,4].

Hence, to become operational again, a failed system must now wait for the next repair check, which takes four times longer than an inspection. This should increase system downtime and reduce availability. It is less clear how system reliability, ENF, and costs would be affected: The degradation mechanisms remain unchanged and the likelihood of failures may not be altered. In turn, total costs might increase due to the higher cost of repairs w.r.t. cleans.

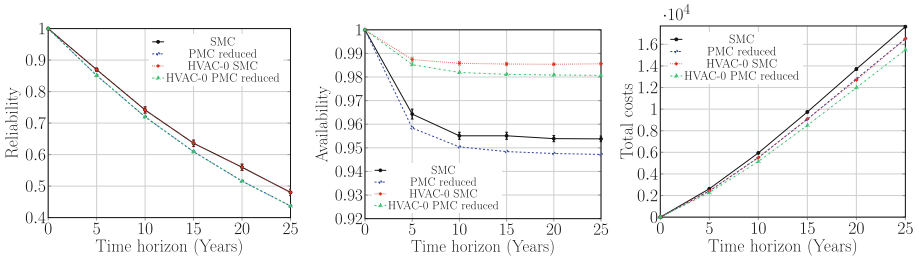


Fig. 4. Metrics for HVAC-1 (full system): SMC and PMC reduced

Figure 4 shows that system un-availability in HVAC-1 is 2 to 3 times higher than in HVAC-0; other metrics are much less affected. Due to space constraints, ENF values are not shown. However, they correlate to those obtained in HVAC-0. Since *PMC exact* creates a state-space explosion, here we opt for PMC reduced; the results obtained validate those achieved via SMC and highlight the trade-off between state-space reduction and deviation from SMC metrics.

3.3 HVAC-2: Deterministic Time Periods

In the previous section we give a first glimpse of how significantly a model refinement can impact a KPI. Here, building on top of HVAC-1, we focus on the modelling of events in time. In the FMTs from Sects. 3.1 and 3.2 and following [5], periodic events like inspections and overhauls are emulated using 3-phase Erlang distributions. Originally, this was needed because the model had semantics exclusively in terms of CTMCs. For the HVAC-2 model presented in this section we employ (more realistic) *deterministic* time periods instead [1,4].

This refinement has a twofold motivation: First, on the modelling side, events occurring at specific time points are a common maintenance policy—e.g. inspections are scheduled *exactly* every six months. Accurately modelling this is relevant for cost analyses, specially for (high) one-time investments like overhauls. Second, on the analysis side, Erlang approximations as phase-type distributions tend to the desired behaviour as the number of phases increases [10]. Thus, to achieve more deterministic-like delays, the state space of PMC models must grow, since e.g. Erlang phases are integral variables in the CTMCs of PRISM. To reach the desired level of realism, all variables encoding periodic time delays (inspections, cleans, etc.) require $\geqslant 10$ values. This would result in $\gg 10^8$ states, meaning PMC via PRISM cannot be performed [15]. Therefore, from this section onwards, we use exclusively PTA models of the HVAC FMT, which can naturally encode (true) deterministic time intervals with no impact on the state space. The KPIs can thus be measured using SMC alone[1].

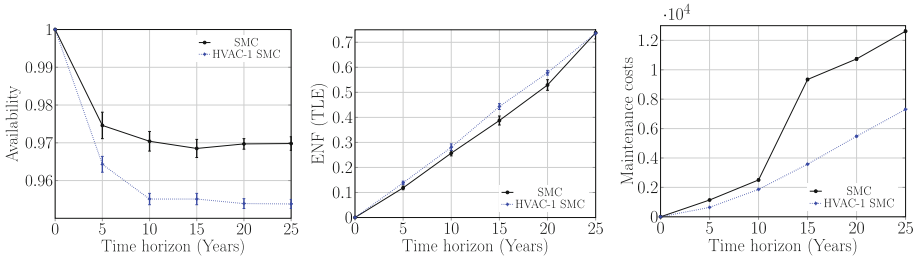


Fig. 5. Metrics for HVAC-2 (full system): SMC

We present the KPIs for HVAC-2 in Fig. 5: The most prominent modification w.r.t. HVAC-1 are in the maintenance costs. The costs incurred by the overhaul

[1] Support for reward analysis on PTA with PRISM is ongoing research, see Sect. 4.

at 15 years can be clearly appreciated for HVAC-2, whereas they are spread-out and less noticeable for HVAC-1. Detailed information like this can be crucial e.g. when assessing investment portfolios. The difference in the reliability and ENF values estimated for HVAC-2 are insignificant when compared to HVAC-1. However, availability is higher for the former. Looking at the individual maintenance actions triggered in the simulations of UPPAAL, after 5 years HVAC-1 performs 7 inspections on average, whereas HVAC-2 performs 10; after 25 years we get 37 vs. 48 respectively, etc. This reveals that, on average, the 3-phase Erlang approximation of the deterministic time delays in HVAC-1 is over-approximating. Repair checks are thus performed more frequently in HVAC-2, where the maintenance protocol is *better emulated* than in previous models. This is corroborated by the number of repairs, which is about ten times higher in HVAC-2 than in HVAC-1, accounting also for the generally higher maintenance costs in HVAC-2. Consequently, since the probability and number of failures are not altered, system downtime is lower in HVAC-2 than in HVAC-1. This explains the availability values from Fig. 5. Computation times for these SMC analyses increases considerably w.r.t. HVAC-0 and HVAC-1: While UPPAAL converged to the desired values in a matter of minutes for those models, it took several hours to compute some of the metrics for HVAC-2. This is discussed in Sect. 4.

3.4 HVAC-3: Spares for Affordable Components

In Sects. 3.1, 3.2 and 3.3 we increasingly refined our model to improve realism. Following the same goal, in this section we extend the HVAC to include spares in some subsystems. Redundancy is a common practice in high-resilience or safety-critical systems: RAID data storage uses extra disks to keep system availability high, cars come with a spare wheel, all modern air-conditioned buildings have spare air filters [3], etc. In HVAC-3 we use spare gates (SPARE) to implement *cold spare components* (whose degradation starts only after a fault occurs, [20]) for the valves in the water circuit, i.e. EBEs 6 and 7.

Two reasons motivated the choice of these components: On the one side, valves are relatively affordable parts (compare them to the boiler or the radiators) for which redundancy should require minor investments. On the other side, in Fig. 2b indicates these EBEs fail the most often. The impact observed in resilience should thus be greatest when providing spares for such components.

We add a SPARE with one spare component for EBE 6, and another (independent) SPARE with two spare components for EBE 7[2]. Spare components are assumed identical to main components, and as soon as the main component fails, the corresponding SPARE will switch to a spare without incurring system downtime[3]. When the main component and all spares have failed, the SPARE fails and propagates a signal to the rest of the FMT. We set at €1000 the cost of using a spare component. This way spares are more expensive than repairing the valve (€800), but cheaper than a full system overhaul (€5000). For all previous cases, the cost of

[2] Higher redundancies lead to *rare failures* that hinder SMC analyses, see Sect. 4.

[3] Notice that a valve can be replaced in hours, whereas all time horizons are in years.

a triggered maintenance action (e.g. a clean) is independent of the number of EBEs affected by it. For HVAC-3, the cost of using n spares during system operation is $n \times €1000$.

Spares are replenished during repair checks. Costs of using a spare are not incurred immediately, but rather in the next repair check occurring after the use of the spare. The intuition behind this is that the technician that periodically visits the company to perform repairs, is also in charge of replenishing the spares. The company pays him then for all pieces that have been used since his last visit.

Unlike in the previous sections, this extension affects only a subsystem of the HVAC, and in particular does not involve the "Supply fan failure" subtree from Fig. 2a. To highlight the effect of these modifications, we measure the KPIs exclusively for the affected part of the model, i.e. an FMT without EBEs 2–4. Accordingly, when referencing models from previous sections, we allude to the KPIs measured for the corresponding FMTs that also disregard fan failures.

The results of our SMC analyses on HVAC-3 are presented in Fig. 6. To exercise the capabilities of spares we also study a scenario where maintenance occurs half as frequently. Thus with "half maintenance," as opposed to "stnd. maintenance," inspections occur every year, repair checks every four years, and overhauls every thirty years. Notice that here we measure the KPIs for seven time horizons, i.e. for $T \in \{5, 10, 15, \ldots, 35\}$ years of system operation.

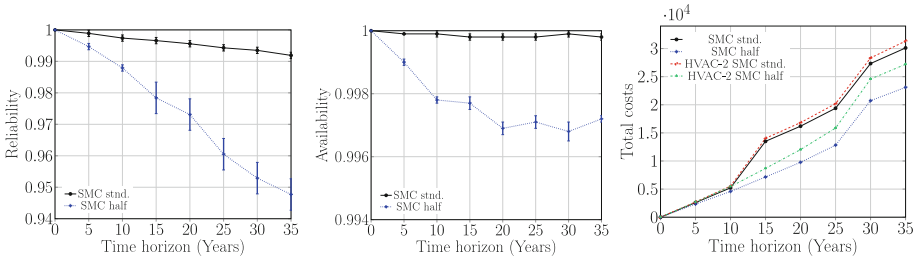


Fig. 6. Metrics for HVAC-3 (no supply fan): SMC

Except for costs, direct comparison of the KPIs from HVAC-3 and its predecessors is omitted, because HVAC-3 is so fault tolerant that its metrics would appear as flat lines on top of the charts. For HVAC-2 with stnd. maintenance, system availability without fan failures converges to 0.97, similarly to the full-system metrics from Fig. 5. For HVAC-3 availability was always above 0.9998, see "SMC stnd." in the availability plot of Fig. 6. With half maintenance, availability for HVAC-2 converged around 0.89 and for HVAC-3 around 0.997. Comparing thus HVAC-3 with half maintenance against HVAC-2 with stnd. maintenance, viz. 0.997 vs. 0.97, we get that using 1 & 2 spares for EBE 6 & 7, *and reducing maintenance checks to a half*, system downtime is reduced by more than 10×.

Reliability exhibits a similar trend: With stnd. maintenance it decreases in a seemingly linear fashion, reaching the value 0.4078 at the 35 years time horizon for HVAC-2, and 0.9919 for HVAC-3. With half maintenance the reliability

values of HVAC-2 and HVAC-3 at 35 years are respectively 0.2053 and 0.9477. The values computed for ENF resemble this ratio: 0.9589 vs. 0.0069 for stnd. maintenance and 1.6407 vs. 0.0565 for half maintenance.

This tremendous increment in system resilience is explained by the Erlang degradation modelling in the EBEs. With one component we get N exponential jumps of rate $\frac{N}{\text{MTTF}}$ each. Adding $m-1$ spares identical to the main component multiplies the number of jumps by $m-1$, yet keeps the rate constant. Therefore, having $m-1$ spares is equivalent to having an Erlang$(mN, {}^{N}/_{\text{MTTF}})$, whose expected value is m MTTF. In the setting of HVAC-3 this means that the MTTF of EBEs 6 and 7 changed to 20 and 30 years respectively, with the corresponding decrease in the probability of failure for a given fixed time window.

Nonetheless, although resilience improved drastically, total costs are actually *lower* for HVAC-3 than for HVAC-2 under both maintenance schemes, see Fig. 6. This is a consequence of the 4-to-1 cost ratio of system downtime/uptime. In spite of the extra maintenance costs incurred to keep stocks of spares, operational costs are much lower due to the very low proportion of system downtime.

It is straightforward to conclude that redundancy (spares) are a must in high-resilience systems. As final comment we mention that SMC analyses took significantly longer than all previous studies. For instance, computing the unreliability at 35 years under stnd. maintenance (0.0081) took 78 h of wall-clock computer time. This issue, discussed in Sect. 4, is due to (i) the addition of two longer time horizons, and (ii) the rarity of observing an event in the time window considered. In comparison, PMC should be less affected by these causes.

3.5 HVAC-4: Randomised Continuous Degradation of EBEs

In all previous sections the EBEs were regarded as atomic elements of the FMT, and modelling did not differ from the basic setting of HVAC-0. Here we refine the degradation semantics of EBEs: Instead of using discrete phases we model degradation continuously. We focus on the "Supply fan failure" subsystem, making the degradation of the fan bearings (EBE 4) resemble a continuous stochastic process known as *Geometric Brownian Motion* (GBM).

We use GBM for two reasons: First, recent studies show GBM can appropriately model bearing degradation [23]. Second, when using N discrete phases of rate λ, time increments between consecutive degradation stages are sampled from N independent and identically distributed (iid) random variables $\sim$ Exponential(λ). Failure times thus follow an N-phase Erlang distribution with expectation MTTF $= N/\lambda$. In contrast, our GBM simulation uses *constant* time increments, and for ever smaller time increments the degradation process is continuous (with probability 1). So degradation is a non-iid non-linear—thus not linearly phased—process, and changes in degradation between consecutive instants are partially stochastic. Consequently, the degradation *speed* of a component is a function of time, and since the expectation of GBM is an exponential function of time, the *failure time follows a log-normal distribution*.

Technically, GBM is the analytical solution to the stochastic differential equation $S(t) = S(0)\exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W_t\right)$. Next we review its main concepts as used in this section, and refer the interested reader to the abundant literature for a deeper insight into GBM. Let $S(t)$ denote the (continuous) degradation of a system component at time t, with $S(0) = 1$ and Δt the time increment. Then the GBM degradation can be expressed and simulated as

$$S(t + \Delta t) = S(t)\exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma W_{\Delta t}\right). \tag{1}$$

In Eq. (1) $W_{\Delta t}$ is a Wiener process or "Brownian motion," meaning $W_{\Delta t}$ is normally distributed with zero mean and variance Δt, and has independent Gaussian increments. Parameters μ and σ in Eq. (1) are respectively the *drift* and *diffusion* coefficients. The expected value and variance of $S(t)$ are given by $\mathbb{E}[S(t)] = S(0)\exp(\mu t)$ and $\mathrm{Var}[S(t)] = S(0)^2\exp(2\mu t)(\exp(\sigma^2 t) - 1)$.

In previous sections, each EBE is characterised by its mean time to failure (MTTF) and number of degradation phases (N). As we now only change the degradation function, we can express the expected degradation value at the MTTF as: $S_{fail} = \mathbb{E}[S(\mathrm{MTTF})] = S(0)\exp(\mu\,\mathrm{MTTF})$. Hence, assuming $\mathrm{Var}[S(\mathrm{MTTF})] = 1$ and setting $S(0) = 1$ and $S_{fail} = N + 1$, this yields

$$\mu = \frac{\ln(S_{fail}) - \ln(S(0))}{\mathrm{MTTF}} = \frac{\ln(N+1)}{\mathrm{MTTF}} \tag{2a}$$

$$\sigma = \sqrt{\frac{\ln\left(1 + \frac{\mathrm{Var}[S(\mathrm{MTTF})]}{S(0)^2\exp(2\mu\mathrm{MTTF})}\right)}{\mathrm{MTTF}}} = \sqrt{\frac{\ln\left(1 + \frac{1}{(N+1)^2}\right)}{\mathrm{MTTF}}}. \tag{2b}$$

Thus, using Eq. (1) with the drift and diffusion from Eq. (2) as degradation function for EBE 4, and keeping EBEs 2 and 3 unmodified w.r.t. HVAC-2, we analyse the FMT for the supply fan subsystem. Since the MTTF values of EBEs 2–4 are among the highest of the model, failures will be rare in the time windows considered. In that sense, this scenario is similar to the one from the previous section, and we thus follow a similar approach: We study a scenario with half maintenance for time horizons $T \in \{5, 10, \ldots, 35\}$.

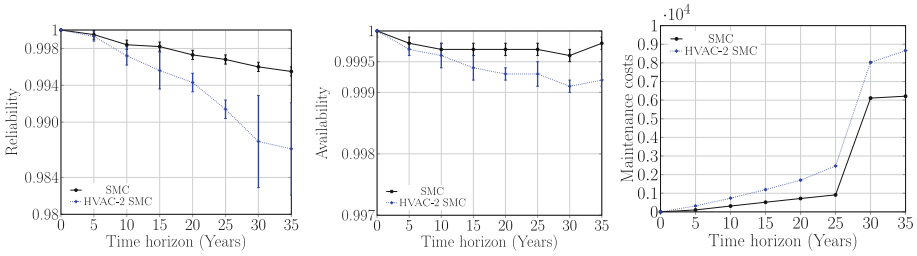


Fig. 7. Metrics for HVAC-4 (supply fan): SMC

The results of experimentation are shown in Fig. 7. Comparing the KPIs against HVAC-2, we observe that HVAC-4 models a more resilient system. This is a direct consequence of using GBM for the degradation of EBE 4, even though it has the lowest MTTF in the subsystem considered. More in detail, the expected degradation of this component in HVAC-4 is continuous and exponentially increasing. The expected time it takes for the GBM to reach a certain degradation value is thus *logarithmic* w.r.t. the degradation value. In contrast, degradation is linearly phased in HVAC-2 for all EBEs, and thus the corresponding expected time is *linear* w.r.t. the degradation state. Because of identical MTTF and failure threshold, the degradation speed in the initial stages of EBE 4 is lower in HVAC-4 than in HVAC-2. Therefore, inspections in HVAC-4 have a higher chance of restoring EBE 4 to its initial value, while in HVAC-2 it is easier to have the component degrade by two phases before the inspection can trigger a clean. As a consequence, given a fixed maintenance scheme and time period, the chances of EBE 4 failing are lower in HVAC-4 than in HVAC-2. This explains the reliability and availability values observed in Fig. 7.

The costs are also an interesting point of comparison. Operational costs are identical for both models, and thus omitted from Fig. 7. Maintenance costs however present a major distinction: HVAC-4 has nearly no costs incurred by repairs, whereas repairs in HVAC-2 increase linearly from €123 after 5 years, to €1984 after 35 years. This is precisely the expected behaviour as per the argument presented above: since most degradation is "early caught" by inspections in HVAC-4, maintenance costs are concentrated in cleans, rather than repairs. This shows, once again, that refining the FMT model can noticeably impact both the resilience KPIs, as well as the costs of system operation and maintenance, in ways and quantities that may concern the interested parties.

4 Concluding Remarks and Future Work

In this work, we demonstrate the importance of the semantic details of an FMT system model, by quantifying the effect of modifications to the model on typical RAMS metrics estimated via PMC and SMC. We propose four (incremental) modelling improvements for a basic HVAC-0 FMT model. We note that (i) the localisation of clean actions to only degraded components increases the unavailability by a factor of $\approx 2.5\times$ (HVAC-1, see Fig. 4), (ii) modelling periodic events with deterministic time delays increases the resilience KPIs and greatly impacts costs (HVAC-2, see Fig. 5), (iii) the use of spares reduces the frequency of maintenance actions while achieving $>100\times$ higher availability and slightly lower costs (HVAC-3, see Fig. 6), and (iv) using GBM to model component degradation increases resilience metrics (in particular reliability) and reduces costs, but makes analyses more involved and, arguably, more realistic (HVAC-4, see Fig. 7). It is thus evident that much can be gained by revisiting an otherwise finished model, and refining any particularly relevant component.

Future Work. There are several areas open for further development. First, from HVAC-2 onwards only SMC could be used because CTMCs cannot emulate deterministic time delays. Current endeavours by the PRISM community to measure reward properties on PTA models [15] are opening the gate to PMC studies of the cases presented in Sects. 3.3, 3.4 and 3.5. Moreover, when the time window is large w.r.t. the event-time-unit simulated (or when the event of interest rarely happens), SMC suffers from longer computation times due to the duration of the (resp. required amount of) simulations, see e.g. Sects. 3.4 and 3.5. If PMC could be used to analyse HVAC-3 and HVAC-4, the time required to converge to an estimate should be faster [15]. Rare event simulation offers an alternative, to apply SMC when the event of interest occurs with very low probability [18]. Parallelly, the next natural step to the EBE refinement from Sect. 3.5 is data validation, i.e. comparing the KPI metrics against measurements from real systems. However, such measurements are scarce due to the long time horizons involved. It would also be interesting to experiment with different degradation functions for distinct EBEs, specialised for the behaviour of each component type concerned. Further relevant extensions include measuring the effect of "on-demand" maintenance in addition to fixed periodic maintenance, and experimenting with different cost schemes to test the robustness of the final conclusions.

Acknowledgements. This work is partially supported by the Alan Turing Institute, UK; Malta's ENDEAVOUR Scholarships Scheme; and the NWO SEQUOIA project.

References

1. Aven, T., Jensen, U. (eds.): Maintenance optimization. Stochastic Models in Reliability, pp. 169–211. Springer, New York (1999). https://doi.org/10.1007/978-0-387-22593-7_5
2. ASHRAE: HVAC systems and equipment. American Society of Heating, Refrigerating, and Air Conditioning Engineers, Atlanta, GA (1996)
3. Au-Yong, C.P., Ali, A.S., Ahmad, F.: Enhancing building maintenance cost performance with proper management of spare parts. JQME **22**(1), 51–61 (2016)
4. Barlow, R.E., Proschan, F.: Mathematical theory of reliability. Science **148**(3674), 1208–1209 (1965)
5. Cauchi, N., Hoque, K.A., Abate, A., Stoelinga, M.: Efficient probabilistic model checking of smart building maintenance using fault maintenance trees. In: BuildSys (2017)
6. Cauchi, N., Macek, K., Abate, A.: Model-based predictive maintenance inbuilding automation systems with user discomfort. Energy **138**(Suppl. C), 306–315 (2017)
7. Clarke, E., Emerson, E., Sistla, A.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Trans. Program. Lang. Syst. **8**, 244–263 (1986)
8. Clarke, E., Grumberg, O., Peled, D.: Model Checking. MIT Press, Cambridge (1999)
9. David, A., Larsen, K., Legay, A., Mikučionis, M., Poulsen, D.: Uppaal SMC tutorial. Intl. J. Softw. Tools Technol. Transf. **17**(4), 397–415 (2015)

10. David, A., Larry, S.: The least variable phase type distribution is Erlang. Communications in statistics. Stoch. Models **3**(3), 467–473 (1987)
11. Dugan, J.B., Bavuso, S.J., Boyd, M.A.: Fault trees and sequence dependencies. In: RAMS, pp. 286–293 (1990)
12. Faisal, I., Mahmoud, M.: Risk-based maintenance (RBM): a quantitative approach for maintenance/inspection scheduling and planning. J. Loss Prev. Process Ind. **16**(6), 561–573 (2003)
13. Guck, D., Spel, J., Stoelinga, M.: DFTCalc: reliability centered maintenance via fault tree analysis (tool paper). In: Butler, M., Conchon, S., Zaïdi, F. (eds.) ICFEM 2015. LNCS, vol. 9407, pp. 304–311. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25423-4_19
14. Junges, S., Guck, D., Katoen, J.P., Stoelinga, M.: Uncovering dynamic fault trees. In: DSN, pp. 299–310. IEEE, June 2016
15. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic model checking: advances and applications. In: Drechsler, R. (ed.) Formal System Verification, pp. 73–121. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-57685-5_3
16. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_47
17. Nicolai, R.P., Dekker, R.: Optimal maintenance of multi-component systems: a review. In: Kobbacy, K.A.H., Murthy, D.N.P. (eds.) Complex System Maintenance Handbook, pp. 263–286. Springer, Heidelberg (2008). https://doi.org/10.1007/978-1-84800-011-7_11
18. Rubino, G., Tuffin, B. (eds.): Rare Event Simulation Using Monte Carlo Methods. Wiley, Hoboken (2009)
19. Ruijters, E., Guck, D., Drolenga, P., Peters, M., Stoelinga, M.: Maintenance analysis and optimization via statistical model checking. In: Agha, G., Van Houdt, B. (eds.) QEST 2016. LNCS, vol. 9826, pp. 331–347. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43425-4_22
20. Ruijters, E., Stoelinga, M.: Fault tree analysis: a survey of the state-of-the-art in modeling, analysis and tools. Comput. Sci. Rev. **15**, 29–62 (2015)
21. Vesely, W.E., Goldberg, F.F., Roberts, N.H., Haasl, D.F.: Fault tree handbook. Technical report, U.S. Nuclear Regulatory Commission, Washington DC (1981)
22. Volk, M., Junges, S., Katoen, J.P.: Fast dynamic fault tree analysis by model checking techniques. IEEE Trans. Ind. Inform. **14**(1), 370–379 (2018)
23. Wang, D., Tsui, K.L.: Statistical modeling of bearing degradation signals. IEEE Trans. Reliab. **66**(4), 1331–1344 (2017)
24. Younes, H.L.S., Simmons, R.G.: Probabilistic verification of discrete event systems using acceptance sampling. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 223–235. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45657-0_17