Building Business Cases for Risk and Reliability Technologies

V. Volovoi
Independent Consultant
Alpharetta, Georgia, USA

ABSTRACT: System reliability tools as a means for supporting business cases are considered. The resulting quantitative insights connect new technology developers with the business and management world. Recent advances in the ability to generate (sense), collect, store, and process large amounts of data have direct implications for the reliability engineering. The long term visions of the “Industrial Internet” and the “Internet of Things” are very attractive indeed. However, in the short term, there is a great need for incremental steps in demonstrating the value of the associated suite of technologies. To this end, improving reliability of complex systems and mitigating the negative impact of their failures are often listed among the short-to-medium term use cases. The paper focuses on the desired properties of system reliability models that can quantify the benefits for these use cases. Specifically, the models must be created relatively fast with the minimal resources and provide a high level of transparency, so that the models can be documented in a way that makes them easily auditable and understandable by the decision makers. To address these needs, the use of a new version of Stochastic Petri Nets (SPNs) called Abridged Petri Nets (APN) as a candidate framework is suggested. Several specific examples are considered. In particular, condition-based maintenance (CBM) concepts with both periodic and continuous monitoring are investigated and compared. The utility of various performance measures for CBM are discussed, including false positives rates and correlation between consecutive inspection success. In addition, a scenario of introducing new technologies (such as advanced fleet-based or Big Data analytics) into the maintenance and spares supply chain is presented.

1 INTRODUCTION

The modern capabilities for dealing with large-scale data collection, storage, and statistical processing have clearly become a focal point of public attention, as testified by the popularity of buzzwords such as “Analytics” and “Big Data” (Siegel 2013). It seems natural to assume that these capabilities are directly applicable for addressing the challenges of predicting failures of complex systems. Furthermore, Amazon, Google, and similar “New Economy” companies at the forefront of the Big Data movement also operate “cloud” infrastructures, and so should be interested in preventing their failures. However, this particular example of complex engineering systems benefits from the feasibility of affordable and highly redundant configurations, with the consequences of individual failures often negligible. This is not always the case with other complex (*e.g.,* safety-critical) systems.

At the same time, at least at the moment, the Big Data embraces correlation at the expense of causation. The vast majority of modern analytics focuses on domain-independent machine-learning methods. They fall under the category of black-box as opposed to white-box models (Blischke & Murthy 2000). In this context, the key question is whether the failure process of an entity is modeled with or without the explicit recognition of individual constituents (components) that comprise the entity. Here “component” refers to an elementary building block of a white-box (system) model, which can correspond to a lower-level entity if models are constructed hierarchically, or to the lowest level of the hierarchy, as determined by practical considerations (*e.g.*, individual modules, such as line-replaceable units or LRU).

White-box models explicitly describe interrelationships among the entities that comprise the system, and so can be considered as a subset of engineering models. In contrast, black-box models do not require explicit modeling of the constituents, as the focus is on the best possible mapping between the inputs and the outputs without any attempt to analyze the causal mechanisms that lead to the observed or predicted mapping. As a result, such methods have broad range of applications, and they can be very useful in predicting regularly occurring events in existing complex systems. However, they are of limited use in predicting the behavior of new systems, the impact of new features introduced to existing systems, or extreme events affecting existing systems for which not enough historical data exists.

The shifting balance between causation and correlation is rather explicitly advocated (Siegel 2013), and can be considered as a natural compensation for over-reliance in the recent past on constructing causal models that were often counter-productive. As eloquently explained by one of the living classics of psychology, humans have clear propensity to construct causal models even when it is not rational (Kahneman 2011), so this current compensation brought in by the Big-Data tide has merit. There is another important distinction, that can also explain why, from this viewpoint, the subject of this paper is distinct from the mainstream Big Data applications. Most of the current applications of Big Data are focused on consumers behavior where causal connections are tenuous at best. In contrast, the application of Big Data to engineering systems (so-called "The Internet of Things"(L. Atzori 2010) as applied to large industrial systems, sometimes referred to as "Industrial Internet") is fundamentally different from this perspective, as the components and systems have been designed (engineered) to follow certain logical relationships, so that known causal relationships are present at least most of the time. It is too early to tell how this increased role of causality will play out in the context of the Internet of Things. However, it is clear that the higher consequences of errors require a more direct reliance on engineering models, and pure domain-independent machine learning is insufficient.

The juxtaposition between causality and correlation is not a zero-sum game, and as happened many times before in the history of science, a tool from one field can successfully migrate into another field, thus enhancing the recipient. The obvious near-term benefits to the system reliability include drastic improvements to and automation of failure data collection, which provide better input for reliability models and allow verification of the quality of reliability predictions. One of the promising directions lies at the intersection of analytics and failure modeling and anomaly detection, where large-scale experiments demonstrated the feasibility and usefulness of automatic processing of operational data (Chu et al. 2011).

SPN is well-recognized as a flexible framework for dynamic system reliability (i.e., "white-box") modeling. However, SPN models are generally perceived as too complicated (and even spaghetti-like) to be effective in facilitating the direct communication with the decision makers (Bowden 2000). As a result, there is a recent trend for using SPN as an intermediate layer of modeling hidden from the end user (Signoret et al. 2013). However, this approach results in the loss of the visibility of the modeled dynamic interactions. In contrast, a streamlined version of SPN called Abridged Petri Nets (APNs) (Volovoi 2013) retains modeling power of SPNs while enhancing the visual modeling clarity, making the models more suitable for the end user consumption and facilitating auditable tool-independent self-documentation. APN in briefly discussed next along with an application-based (as opposed to a computer science) perspective on SPNs.

2 ABRIDGED PETRI NETS

Both Markov chains and Stochastic Petri nets (SPNs) can be viewed as state-space representations of stochastic processes. While each state of a Markov chain represents a possible state of the entire modeled system, Petri nets model the states of individual components rather than the explicit states of the entire system. Graphically, Petri nets complement Markov chain-state diagrams with two new types of objects: first, small filled circles (called tokens) denoting individual components are introduced, each placed inside of one of the larger hollow circles that denote the potential states of those components (the latter entities are named "places" as opposed to "states" in Markov diagrams). Second, in order to model interactions among components, the tokens are routed among places via intermediate stops or junctures, called transitions, which are denoted with solid rectangles. Any two places cannot be connected by an arc directly; instead they must be connected through a transition. The number of input and output arcs does not need to coincide, enabling the merging and splitting of tokens and their routes.

The timing of state changes can be modeled by specifying time delays for transition "firing": an atomic (*i.e.,* indivisible) action that removes tokens from all input places for the transition and deposits tokens into its output places. Such Petri nets are timed Petri nets, or, more specifically, Stochastic Petri Nets (SPNs) (Balbo 2007, Haas 2002), when delays can be nondeterministic and follow a specified distribution. In simulation no limitations on the associated types of distributions is needed. However, historically the name SPN is often referred to models with exponentially distributed delays only, so that they could be converted to Markov chains and solved using appropriate techniques for the underlying differential equations.

Inhibitors provide a "zero test" capability and are known to increase the modeling power of Petri nets to that of a Touring machine (Balbo 2007, Haas 2002). Both inhibitors and enablers (non-zero test arcs) are usually considered supplementary mechanisms for modeling interactions rather than a replacement for fork-joining capabilities provided by transitions. Historically, inhibitors were viewed with a certain degree of skepticism by the Petri net community, as they make traditional analysis of structural properties (such as reachability analysis) more complex. However, the latest view of this drawback of inhibitors is not as straightforward as the new algorithms can successfully handle inhibitors (Ciardo 2004). In addition, there is a sufficient number of applications (e.g., modeling failure and maintenance processes of complex systems) where the state-space of the problem is relatively well understood and the main utility of the modeling consists of quantitative performance evaluation of the system.

These considerations provide an impetus for relying on triggers as a sole (or at least the main) alternative mechanism for describing components' interactions. This enables direct connectivity of places, thus disposing of the hubs (transitions) altogether. When combined with hierarchical representation, the result is a compact yet powerful modeling framework. It can be shown that the combination of enablers and inhibitors with direct transitions between places allows the modeling of any system that can be modeled using Petri nets with multiple inputs and outputs to a transition (in other words, the modeling power is not reduced). The result is referred to as Abridged Petri Nets (APNs) (Volovoi 2013) and resembles a hybrid between traditional Markov chain-state charts and Petri nets: transitions connect places directly (similar to Markov chains), but tokens are present to represent individual components of the system (similar to Petri nets). Importantly, the tokens have discrete labels (colors) as well as continuous labels (age) (Volovoi 2004).

Transitions as the hubs or junctures of tokens' movements (depicted as rectangles) provide an ingenuous mechanism for modeling components' interactions in classical SPNs. The original idea of Petri nets was conceived in the context of chemical reactions (Petri & Reisig 2008), where the merging (joining) and splitting (forking) of entities is quite common, which might explain the fundamental role of transitions that provide a direct means of modeling these processes. The merging and splitting is also important in the context of informational flows. In other applications such events occur as well, but equally if not more important are the changes that occur to entities individually. In fact, there are clear competitive advantages to modular structuring of complex systems (Simon 2002), which might explain its prominence in both natural and engineering systems leading to so-called near-decomposibility (Courtois 1977). With this consideration in mind, entities that comprise complex systems can be considered as operating mainly independently of each other, with interactions occurring relatively rarely. However, when these interactions do occur, they are critical to the system's behavior.

APN modeling framework reflects this near-decomposability and considers independent (parallel) behavior as the default while explicitly focusing attention on the interactions. The joining (merging) and forking (splitting) of entities represents important mechanisms for modeling a system's interactions, and for those types of interactions the use of the junctures for routing tokens among places is appropriate. However, quite often the actual mechanisms are related to enabling or inhibiting state transitions, while joining and forking provide a means of modeling such mechanisms. The difference is subtle and often not noticeable when tokens are indistinguishable (as in classical SPNs), but the consequences become significant when the entities represented by tokens have distinct identities (labels, such as colors (Jensen 1993)). Next, specific properties of APN are briefly described.

1. An APN is defined as a network of places (denoted as hollow large circles) that are connected by directed arcs (transitions). Changes in the system's state are modeled by transition firing: *i.e.,* moving a token from the transition's input place to its output place. The combined position of APN tokens at any given moment represents the net marking and fully specifies the modeled system.

2. Each transition has no more than a single input and single output place (if an input place is of the source type, it generates a new token every time the transition fires, while the output place can be of the sink type: upon the firing of such a transition, a token is removed from the net).

3. The use of token junctures, that correspond to the transitions used in regular SPNs is optional. While junctures do not increase the modeling power, there are situations where the use of merging and splitting of tokens allow for more compact models. In contrast to SPNs, these entities are relatively rare, and analogous to the batching and separating building blocks in process-interaction frameworks for discrete-event simulations (Law & Kelton 2000). These entities are not used in the current paper.

4. Each token can have a discrete label (color) that can change when token is fired in accordance with the policy specified by the firing transition. In addition, tokens have continuous labels (ages) that can change both when tokens move, and with the progression of time while a token stays in the same place (the latter property is specified by the aging transition for the place, which is not necessarily the same as the firing transition (Volovoi 2004)).

5. A transition is enabled or disabled based on the combined marking of the input places of the associated triggers (inhibitors and enablers). Inhibitors are depicted as arcs originating at a place and terminating at a transition with a hollow circle. An inhibitor of multiplicity k disables a transition that it terminates at if the number of tokens in its input place is at least k. An enabler (depicted as an arc originating at a place and terminating at a transition with a filled circle) is

opposite to an inhibitor: a transition is disabled unless an enabler of multiplicity k has at least k tokens in its input place (Volovoi 2006). Enablers are effectively test arcs (Christensen and Hansen 1993)), which are used in system biology modeling (Matsuno et al. 2003), where they are denoted with directed dashed arcs; notations used in this paper are chosen to emphasize the fact that enablers are the opposite of inhibitors. It can be shown that the use of triggers allow to fully compensate for the single-input single-output limitation on the transition (no modeling power is lost as a result).

6. Transitions have color- and age-dependent policies that specify the delay between the moment when the token is enabled and when it is fired (for example one can specify separate distributions for distinct colors, while the age can accumulate as the cumulative distribution function of the aging transition). If a token-transition pair is enabled, a firing delay is specified based on the combination of token and transition properties. If the token stays enabled throughout the delay, after this delay expires the token is fired. If there are multiple enabled tokens in the same place, they all can participate in the firing "race" in parallel. Similarly, the same token can be involved in a race with several transitions. If a token-transition pair is disabled, the firing is preempted and the aging label of the token can change as a result of being enabled for a finite amount of time.

7. The delays can be deterministic (including zero delay) or follow any specified random distributions. The firing after a specified delay is "atomic": it is a single action of moving a token from an input place to an output place (tokens don't dwell between places as they potentially do in some versions of SPNs (Bowden 2000)).

8. The performance of the system is based on the statistical properties of marking of the system. "Sensors" or "listeners" at each place (denoted with small filled square at the right bottom of the place) can evaluate the chances and the number of times a given threshold of the number of tokens is crossed, or evaluate the relevant statistics about the number of tokens at a given place. In the latter case the correlation matrix for all results can be evaluated as well, providing the mechanism for calculating the variances (in addition to the mean values) of global metrics that aggregate the readings of individual sensors.

9. Fusing places, commonly used in hierarchical Petri nets (see for example, (Jensen 1993)) is employed to connect different parts of the model. Fused places appear as distinct graphical entities during the model construction, but represent the same entity in simulation. This feature is not directly employed in the current paper, but it critical for creating system-level models with multiple components.

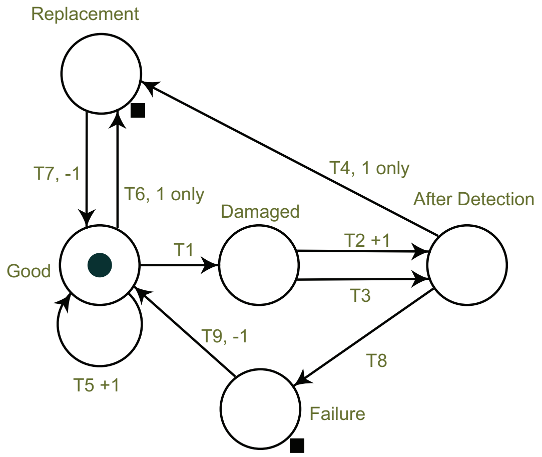

Figure 1: APN model for CBM with continuous monitoring

3 CBM WITH CONTINUOUS MONITORING

There is a significant body of literature devoting to the evaluating CBM performance, including the description of the best practices, see for example (ADS-79D-HDBK 2013). Two types of performance measures can be identified:

1. Diagnostic, or classification performance measures that quantify the ability to classify the entities into the groups directly related to specific maintenance guidance. In the simplest scenario, a two-way classification ("good," no maintenance action vs. "bad," triggering a maintenance action) can be considered. Multi-way classifications are also possible, such as score cards described in (ADS-79D-HDBK 2013) that color-code the entities in accordance with the urgency of the maintenance actions. In either case, the classification correspond to some physical difference (fault) that can be independently verified (e.g. presence of cracks of a certain size, presence of debris, etc.). The corresponding performance measures (such as false positive and false negative rates) can be evaluated based on success/failure criteria using standard statistical procedures.

2. Prognostic or reliability performance that is directly related to the timing of a failure (or a degradation that has direct negative economic consequences). To this end, remaining useful life (RUL) or P-F interval, which is characterized by a distribution is usually discussed. The utilization of this type of measures is significantly more vague. While the measures for accuracy of prediction of RUL have been proposed based on the

> comparison between the predicted and the actual RUL, neither specific performance requirements for this measure, nor statistical procedures for evaluating this measure are provided (ADS-79D-HDBK 2013). The main difficulty of using accuracy-based performance measures for RUL prediction is that RUL is an intermediate indicator that is further transformed into a maintenance decision, and so the consequences of errors in predicting RUL must be assessed in the context of the quality of that decision (*i.e.,* how those errors affected the decision). If a RUL prediction error followed a well-behaved distribution, such as a normal distribution centered around zero (so that the error shrinks as the prognostic distance decreases), averaging the errors would provide a consistent performance measure. However, this is not always the case, as described next, and making this assumption can lead to incorrect conclusions.

There is no consensus about the relationship between the two types of metrics. For example, it is not even clear whether RUL is sufficient on its own, or the classification measures (such as false positive and negative rates) are also needed. Theoretically, if the actual P-F interval distribution is fully known for each entity, one can recover both the missed defects and the false positive rates. Indeed, let us consider an entity that is flagged to be in need of a maintenance action. The actual P-F distribution can be visualized as having a bi-modal shape: the first (larger) peak corresponds to the correct identification of the fault, with the second (smaller) peak located at larger time values corresponds to a mistaken identification of the fault, allowing for the possibility that RUL is quite large. A similar bi-modal structure of P-F distribution for a part that is deemed as normal by the CBM algorithm (with the reversed order of the peaks by size: the first smaller peak corresponds to the missed defects). Depending on the nature of the features used for fault identification, the distribution of RUL can be more complicated. However, from the practical perspective, either some assumptions are made regarding the shape of the underlying distribution, or first two moments (mean and variance) of this distribution are conveyed. As a result, classification performance measures are often lost in translation (see for example (Feldman et al. 2009)) and accuracy-based performance measures are of limited use.

A pragmatic approach consists of combining both types of performance measures, making them complementary to each other. Specifically, this implies that the resolution of RUL prediction should match the classification groups that correspond to the specific maintenance actions. For example if there are four groups used in a score card (ADS-79D-HDBK 2013) going from green to red, a distribution of RUL would be estimated for the entire sup-population of each group (without attempting to make a finer distinction of RUL within each group other than perhaps accounting for the age (usage) of an entity). This subpopulation RUL can be then directly related to the maintenance urgency based on the risk acceptance in a rigorous quantitative manner. Let us next demonstrate this approach for a two-way classification.

A model of a single component CBM with continuous monitoring (Volovoi 2012) is considered and implemented in APN. The resulting model is shown in figure 1, with transition properties are provided in table 1. A two-phase degradation is considered with the parameters of the transitions from the “Good” state (or place as it is referred to in Petri nets) to the “Damaged” place (transition $T1$ in figure 1) and the transition from the “Damaged” place to the “Failure” place ((transition $T8$ in figure 1) provided in (Goode et al. 2000). The token color in the model represents the outcome of the CBM technology - color 0 indicates that the component is deemed to be normal and should remain in operation, while color 1 indicates the damaged diagnosis by CBM. Correspondingly, fast transitions $T4$ and $T6$ are only enabled for color 1. In general, the duration of these two transitions corresponds to the maintenance delay (how long does it for the component deemed to be damaged to be replaced) (Volovoi 2012). In the considered example an immediate maintenance action is modeled to facilitate comparison with periodic inspection scenarios.

When the transition $T1$ fires the token, the modeled component transitions to the damaged state. The two outgoing transitions from the “Damaged” place correspond to the probabilistic choice: the token firing by the transition $T2$ corresponds to the situation when the damage is detected by the CBM technology, and the token changes its color from 0 to 1; otherwise, the damage is undetected, the token is fired by transition $T3$, and the color of the token remains 0. A simple way to implement this probabilistic choice is to assign fast exponential distributions for the firing the delays for the involved transitions with the relative rates proportional to the desired probabilities. In the example considered, $T2$ and $T3$ are chosen to represent 95% detection rate (see table 1). Similar arrangement is made to model false positives. The scale of $T5$ is chosen so that for a given component that is a 5% chance that it will be erroneously identified as damaged while it was in the normal state (5% false positive rate or 95% specificity). Note, that the $T5$ is chosen to follow Weibull distribution with the same shape parameter as the $T1$ transition. As discussed in (Volovoi & Vega 2012), the shape of the distribution is important for the outcome of the race between two transitions. For example, selecting an exponential distribution for $T5$ with the same $19:1$ ratio of mean delays as compared to $T1$, would provide a significantly different (in this case, larger) false positive rate.

Name	Color Change	Color Policy	Type	Par 1	Value 1000 h	Par 2	Value
T1	0	0	Weibull	Scale	0.526	Shape	2.91
T2	1	0	Exponential	Rate	9.5E+06		
		1	None				
T3	0	0	Exponential	Rate	5.0E+05		
T4	0	0	None				
		1	Fixed	Duration	1E-06		
T5	1	0	Weibull	Scale	9.994	Shape	2.91
		0	None				
T6	0	0	None				
		1	Fixed	Duration	1E-06		
T7	-1	0	Fixed	Duration	1E-06		
T8	0	0	Weibull	Scale	0.222	Shape	1.03
T9	-1	0	Fixed	Duration	1E-06		
T10	0	0	Fixed	Duration	2E-06		
T11	0	0	Fixed	Duration	Varies (see Table 2)		

Table 1: Transition descriptions for APN of CBM model

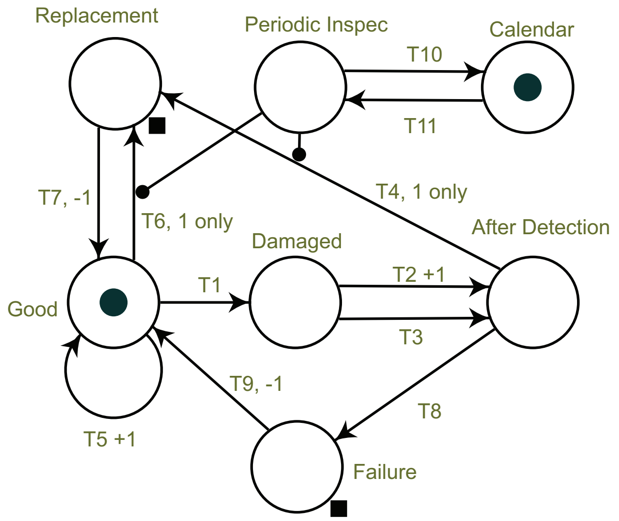

Figure 2: APN model with periodic inspection

4 CBM WITH PERIODIC INSPECTIONS

Next, we develop a model with periodic inspection that is consistent with the model developed in the previous section for continuous monitoring. Specifically, we consider a situation where the observations occur periodically, and those observations effectively identify a “signature” (feature or pattern) of a fault of interest in a repeatable (and usually automated), fashion. Since this “signature” is imperfect, there are cases where both false positive and missing defects error occur. However, in this model we neglect the errors (“noise”) introduced by a single periodic inspection (in other words, if an inspection is repeated immediately after another inspection, no additional benefits are obtained). This model is appropriate for automated repeatable procedures and can be contrasted with a model of periodic inspections where errors are specific to a particular inspection instance (Volovoi 2007). The latter is more appropriate for independent inspections that have individual characteristics, for example, a visual inspection for cracks.

It must be noted that the independence of the inspection results must be treated with caution, especially when it is used to justify a very high overall success rate of a procedure. For example, (ADS-79D-HDBK 2013) considers a scenario where individual probability of detection of a fault is 0.9 with six inspections assumed to achieve six nines probability of detection. The situation is similar to the common cause failures in redundant systems, as it is highly unlikely that the inspections are truly independent. Indeed, the defect location and orientation are common factors affecting the probability of detection (*e.g.*, hard-to-reach location can combine with an orientation that is the least detectable from the viewpoint provided by the likely access to the inspected specimen).

T11 (h)	Failures	Replacements
1000	12.27	2.66
500	9.96	5.99
200	6.53	11.04
100	4.28	14.39
50	2.79	16.65
20	1.75	18.25
10	1.37	18.83
0	1.00	19.40

Table 2: Expected number of failures and replacements for CBM model

The APN model is shown in figure 2. The model is similar to the one for continuous inspection (figure 1), except that two places are added in the top-right portion of the model to represent the periodic inspection schedule. The calendar token needs to be in the “Periodic Inspection” place for the transitions $T6$ and $T4$ to be enabled (see the two enablers to those transitions originating in the “Periodic Inspection” place). Transition $T10$ is chosen to be slow enough to make sure that $T6$ and $T4$ are enabled long enough to fire the component token that has color 1 (i.e., duration of $T10$ is longer than the corresponding policies of those two transitions for color 1, see table 1).

The results of 1 million Monte Carlo simulations for $\tau = 10,000$ hours and various inspection intervals are shown in table 2). The last row corresponds to the continuous monitoring results. The expected number of replacements and failures are shown, so combining this information with the costs of each type of events can provide foundation for the cost benefit analysis. To put these numbers in perspective, we note that the mean time to failure of the modeled component is $MTTF = 688.4$ hours, so the run to failure (RTF) scenario result in $m_\infty = \tau/MTTF = 14.53$ failures for the modeled time duration. Accounting for the finite duration (starting with a new component and simulating τ units of time) provides a very close value $m_{RTF} = 14.11$. Finally, an age-based replacement every 500 hours would result in $m_{a500} = 5.34$ failures with additional 15.46 replacements.

5 UTILIZATION OF ANALYTICS

Let us consider the following scenario that demonstrates interactions of several technologies and potential utilization of the APN capabilities. The new aspect, as compared to the CBM scenarios above, is that here the feasibility of a two-phase process (similar to what is often done in the health-care where the first a screening test that can have potentially high false positive rates, but still serves a useful purpose). The end-item (systems) have repairable line replaceable units (LRU) that have unique serial numbers and upon repair can be installed on an end-item that is different from the original. A portion of the LRU population during the operation transitions to a “bad actor” state (we assume that it happens to 10% of the population every operational cycle). A regular part has a mean time to failure (MTTF) of 250 days, while a repaired “bad actor” part has MTTF of 50 days.

We consider a population of 200 parts, the total duration of the regular repair cycle is eight days (two days retrograde shipment to the depot, five days of repair, and one day shipment back from the depot). A business case is considered for using a new testing equipment that detects the bad actors. The test takes one day, with only one item being tested at a time (so queues are possible). The item determined to be a “bad actor” can undergo an additional repair or replacement that takes two additional days (in addition to the five days required for regular repair). Finally, there is a fleet-level analytics that relies on the Big Data technologies (*e.g.,* as described in (Chu et al. 2011)) that provide (an imperfect) means to predict (based on the usage history and other relevant “markers”) whether the part is a “bad actor”.

The APN model is shown in figure 3 and corresponding parameters are given in table 3. This procedure has 10% false positive and 10% false negative rates (those rates can be changed by adjusting parameters of transitions $T4$ and $T5$). The very definition of a “bad actor” is directly related to the results of the additional test (the units that fail the test are designated as the bad actors, and their performance is shown to be inferior to the units that pass the test). Three scenarios are considered: the baseline when no additional testing, the scenario where all the units are tested, and finally, the “Analytics” scenario that allows selective testing: only items flagged by the fleet-wide analytics are tested. The parameters in table 3 correspond to the third scenario. In order to obtain the scenario with no testing the right branch of the model is turned off by turning off $T5$ transition for both colors. Similarly, turning off $T4$ transition for both colors will route all units through the additional testing.

There are several performance measures that can be considered - including the costs based on the expected number of different maintenance actions (similar to what was considered in the previous example). For the sake of variety we focus instead on the availability measure given by the number of operating units. Figure 4 shows the expected number of operating units as a function of time for the first 300 days of operation. For clarity, the durations of depot actions are chosen to be deterministic, while in reality those durations will vary, allowing for the smoother trends. One can observe a kink when the units start returning from the depot and the three curves diverge. After that, the curve with no testing continues its downward slope, as the portion of “bad actors” in the population will continue to increase. In contrast, both selective and full test scenarios converge to a steady-state solution relatively fast with the selective testing providing superior performance.

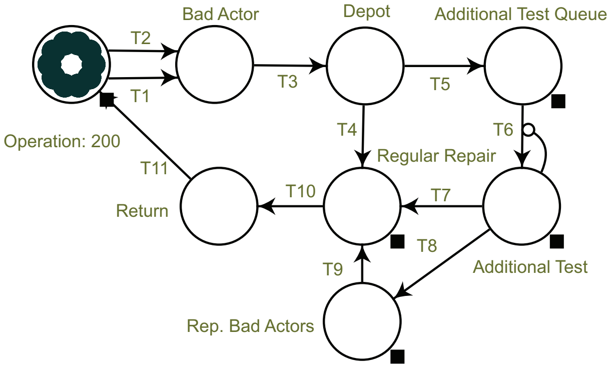

Figure 3: APN model of a simple repair process with a “bad actor” sub-population

Name	Color Change	Color Policy	Type	Par 1	Value Days	Description
T1	0	0	Exponential	Rate	0.0036	Regular failures rate
		1	Exponential	Rate	0.02	Bad actor failures rate
T2	1	0	Exponential	Rate	4E-04	Failure + conversion to bad actor rate
		1	None			
T3	0	0	Fixed	Duration	2.0	Shipment to depot
T4	0	0	Exponential	Rate	9.0E+06	True negative
		1	Exponential	Rate	1.0E+06	False negative
T5	1	0	Exponential	Rate	1.0E+06	False positive
	0	0	Exponential	Rate	9.0E+06	True positive
T6	0	0	Fixed	Duration	1E-06	Immediate
T7	0	0	Fixed	Duration	1.0	Additional test time regular items
		1	None			
T8	0	0	None			Additional test time bad actors
		1	Fixed	Duration	1.0	
T9	-1	0	Fixed	Duration	2.0	Repair/repacement of bad actors
T10	0	0	Fixed	Duration	5.0	Regular repair
T11	0	0	Fixed	Duration	1.0	Shipment from depot

Table 3: Parameter description for the APN model of bad actor problem

For the given set of parameters there is a queue for the additional testing that, on average, has about 1.18 units, while the selective testing has a negligibly small average of units in the queue. If the duration of the additional test is increased to two days, then the system effectively becomes unstable with the queue length growing effectively linear with time until the number of operating units becomes so small that the system reaches an equilibrium. The selective testing is even more beneficial in this case. The example demonstrates how a combination of two technologies (in this case, the introduction of an additional testing and the fleet-wide analytics) provides benefits not achievable by each technology alone. One can envision evaluating the benefits of alternative operational policies, such as testing all units but only once per several cycles of repair, etc.

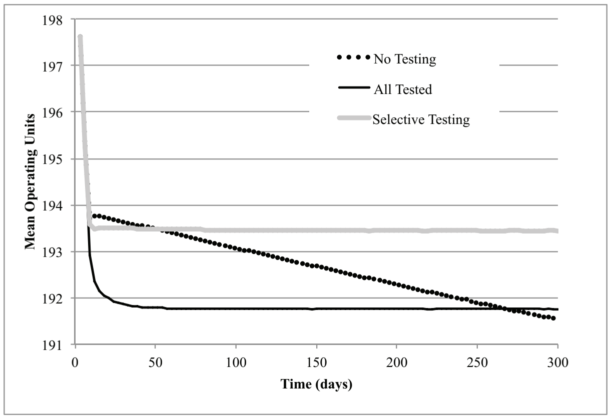

Figure 4: APN results for the expected number of operating units

6 CONCLUSIONS

System reliability tools can rigorously support a business case for a new technology. Quantitative evaluation of value proposition provides a vital link between the technology developers and the business and/or management world. Implementation of a new technology, such as prognostic health management or other changes in maintenance operations carries significant upfront costs. As a result, providing convincing and credible assessment of the value proposition is of great importance. A well-constructed system reliability model created to support a business case clarifies the quantitative requirements for the performance measures of the developed technology. This facilitates not only obtaining the green light for the technology implementation, but provides clear guidelines during the implementation itself. Ideally, the original system-reliability model gets updated, as the technology is developed to reflect a more detailed knowledge about the operation of the new system. To address these needs, the use of a new version of Stochastic Petri Nets (SPNs) called Abridged Petri Nets (APN) is suggested. Several specific examples are considered. In particular, condition-based maintenance (CBM) concepts with both periodic and continuous monitoring are investigated and compared. The utility of various performance measures for CBM are discussed, including false positives rates and correlation between consecutive inspection success. In addition, a scenario of introducing new technologies (such as advanced fleet-based or Big Data analytics) into the maintenance and spare supply chain is presented. In all examples the models are fully described in a tool-independent fashion: the results can be reproduced without the need of a specific software.

REFERENCES

ADS-79D-HDBK (2013). *Condition Based Maintenance System for US Army Aircraft*. Aeronautical Design Standard Handbook. US Army Research, Development, and Engineering Command, Aviation Engineering Directorate.

Balbo, G. (2007). Introduction to generalized stochastic Petri nets. In M. Bernardo and J. Hillston (Eds.), *Formal Methods for Performance Evaluation*, Volume 4486 of *Lecture Notes in Computer Science*, pp. 83–131. Springer-Verlag.

Blischke, W. & D. Murthy (2000). *Reliability. Modeling, Prediction, and Optimization*. New York, NY: Wiley.

Bowden, F. (2000). A brief survey and synthesis of the roles of time in Petri nets. *Mathematical and Computer Modelling 21*, 55–68.

Christensen, S. & N. D. Hansen (1993). Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs. In M. Ajmone Marsan (Ed.), *Application and Theory of Petri Nets*, Lecture Notes in Computer Science, pp. 186–205. Springer Berlin Heidelberg.

Chu, E., D. Gorinevsky, & S. Boyd (2011). Scalable statistical monitoring of fleet data. In *World IFAC Congress*, Milano, Italy.

Ciardo, G. (2004). Reachability set generation for Petri nets: Can brute force be smart? In *Application and Theory of Petri Nets*, Volume 3099 of *Lecture Notes in Computer Science*, pp. 17–34. Springer Berlin Heidelberg.

Courtois, P. J. (1977). *Decomposability: queueing and computer system applications*. New York, NY: Academic Press.

Feldman, K., T. Jazouli, & P. Sandborn (2009). A methodology for determining the return on investment associated with prognostics and health management. *IEEE Trans. on Reliability 58*(2), 305–316.

Goode, K. B., J. Moore, & B. Roylance (2000). Plant machinery working life prediction method utilizing reliability and condition-monitoring data. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering 214*(2), 109–122.

Haas, P. J. (2002). *Stochastic Petri Nets. Modelling, Stability, Simulation*. New York: Springer.

Jensen, K. (1993). *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, Volume 1. Berlin: Springer.

Kahneman, D. (2011). *Thinking, Fast and Slow*. New York, NY: Farrar, Straus and Giroux.

L. Atzori, A. Iera, G. M. (2010). The internet of things: A survey. *Computer Networks 54*, 2787–2805.

Law, A. & W. Kelton (2000). *Simulation Modeling and Analysis* (3rd ed.). New York, NY: McGraw-Hill.

Matsuno, H., Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, & S. Miyano (2003). Biopathways representation and simulation on hybrid functional Petri net. *Silico Biology 3*(3), 389–404.

Petri, C. & W. Reisig (2008). Petri net. *Scholarpedia 3*, 6477.

Siegel, E. (2013). *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die*. New York, NY: Wiley.

Signoret, J.-P., Y. Dutuit, P.-J. Cacheux, C. Folleau, S. Collas, & P. Thomas (2013). Make your Petri nets understandable: Reliability block diagrams driven Petri nets. *Reliability Engineering and System Safety 113*, 61–75.

Simon, H. A. (2002). Near decomposability and the speed of evolution. *Industrial and Corporate Change 11*(3), 587—599.

Volovoi, V. (2006). Stochastic Petri nets modeling using SPN. In *Proceedings of Annual Reliability and Maintainability Symposium*, pp. 75–81. IEEE.

Volovoi, V. (2007). Developing system-level maintenance policies using stochastic Petri nets with aging tokens. In *Proceedings of Annual Reliability and Maintainability Symposium*, pp. 2007RM–170. IEEE.

Volovoi, V. (2012). *IVHM – The Business Case*, Chapter Quantification of System-Level Business Effects of IVHM. Warrendale, PA: Society of Automotive Engineers (SAE) International.

Volovoi, V. (2013). Abridged petri nets. *ArXiv*, arXiv:1312.2865.

Volovoi, V. & R. V. Vega (2012). On compact modeling of coupling effects in maintenance processes of complex systems. *International Journal of Engineering Science 51*, 193–210.

Volovoi, V. V. (2004). Modeling of system reliability using Petri nets with aging tokens. *Reliability Engineering and System Safety 84*(2), 149–161.