Quantifying system safety: A comparison of the SBOAT & Safety Barrier Manager tools

Herbert-Hansen, Zaza Nadja Lee; Duijm, Nijs Jan; Markert, Frank; Herbert, Luke Thomas

Published in:
Proceedings of the European Safety and Reliability Conference (ESREL) 2015

Publication date:
2015

Document Version
Peer reviewed version

Citation (APA):
Hansen, Z. N. L., Duijm, N. J., Markert, F., & Herbert, L. T. (2015). Quantifying system safety: A comparison of the SBOAT & Safety Barrier Manager tools. In Proceedings of the European Safety and Reliability Conference (ESREL) 2015

Quantifying system safety: A comparison of the SBOAT & Safety Barrier Manager tools

Z.N.L. Hansen
N.J. Duijm
F. Markert
Department of Management Engineering
The Technical University of Denmark, Lyngby, Denmark
L. Herbert
DTU Compute
The Technical University of Denmark, Lyngby, Denmark

ABSTRACT: This paper presents two software tools for analyzing safety risks, SBOAT (Stochastic BPMN Optimisation and Analysis Tool) and SBM (SafetyBarrierManager®). SBOAT employs principles from stochastic model checking to allow for the quantitative verification of workflows. SBM supports the creation of valid safety-barrier diagrams and allows the quantitative analysis of the probability of all possible end states of the barrier diagram, i.e. the outcomes if one or several of the barriers fail to perform their barrier function. We compare the foundations of these tools and describe how they can be used and how they complement each other by means of the analysis of a production workflow inspired by a real-world industry case.

1 INTRODUCTION

Ensuring the safe execution of workflows is essential in production operations. Even simple workflows may be embedded in larger networks of processes leading to complex aggregate systems. A range of tools have been developed that allow for automated quantitative analysis of workflows which seek to assist in building safer and more cost-effective systems.

Ensuring safety in such systems frequently involves the addition of control points, where safety properties are verified at key points in the process. Determining the ideal location of such control points at design time leads to safer and cost-effective systems. It may not always be possible to determine the points in a process where failure may occur, so to archive an acceptable level of risk controls are commonly implemented at key stages to identify faults in the preceding subprocess. Determining the optimal placement of control points (safety barriers) must balance a number of factors; including the mean time after a failure has occurred until it is detected, the resources that have been consumed/wasted when failure is likely to be detected and the placement of control points to ensure that a fault does not pass undetected. In this context an appropriate and not too resource demanding risk analysis is essential in building large scale systems.

This paper presents a production workflow inspired by an industry case and compares and explores the combination of the application of two software tools for analysing risks; SBOAT (Stochastic BPMN Optimisation and Analysis Tool) and SBM (SafetyBarrierManager®).

SBOAT allows a user to model processes using an extended form of the Business Process Model and Notation (BPMN) (Object Management Group 2011) language which allows for the incorporation of resource modelling and stochastic behaviour. This tool is based on the mathematical foundation of stochastic process calculi and appropriate model checking techniques and is designed to be easy to use by business practitioners. In practical terms, using SBOAT a business practitioner can model production workflows using familiar notation and include quantitative data such as time or cost and then automatically determine the probabilities of various behaviours, freely defined using formal temporal logic and incorporating specific states of the qualitative properties.

SBM is likewise a tool developed for risk analysis of processes, SBM supports the creation of valid safety-barrier diagrams (Duijm 2009) and allows for the quantitative analysis of the probability of all possible end states of the barrier diagram, i.e. the outcomes if one or several of the barriers fail to perform their barrier function. As input the tool requires system models with specified barrier locations and outputs risk matrices which assess the tolerability of possible outcomes.

1.1 *Safety analysis tools*

The key benefits of safety assessment tools are that they create awareness about risk and safety concerns and can help identify potential hazards. Without this it would not be possible to determine where precautionary measures should be incorporated in a process.

Many risk assessment tools and methodologies have been developed in recent years in order to help and assist enterprises and organisations assess their health and safety risks. What method or tool is best suited for a given organisation depends on several factors, including workplace conditions, for example the number of employees, the complexity of tasks, the work activities and workplace equipment, the physical set-up/location for the work activity and several other factors.

The most common risk assessment tools are checklists, which are a useful tool to help identify potential risks (The European Agency for Safety and Health at Work 2015). Other type of risk assessment tools include: guides, guidance documents, handbooks, brochures, questionnaires, and software tools. Some tools are generic while others have been developed with a specific industry or even a specific branch/risk in mind.

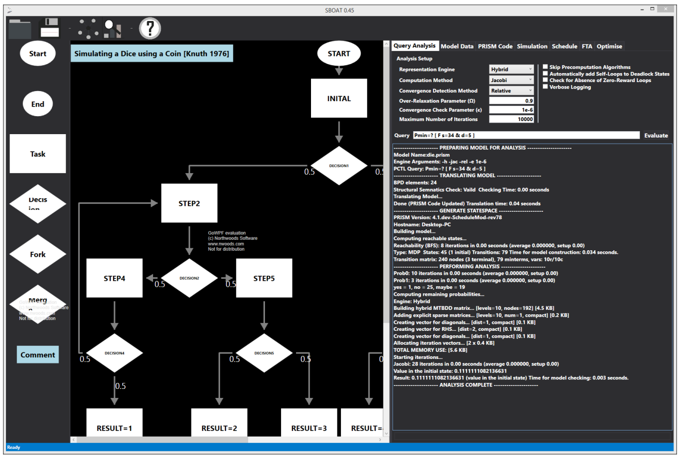
The two tools this paper focus on, SBOAT and SafetyBarrierManager®, are both generic tools intended to help enterprises identify risks and take precautionary measures to the extent deemed necessary and viable for the given enterprise.

2 STOCHASTIC BPMN OPTIMISATION AND ANALYSIS TOOL (SBOAT)

2.1 *Tool Description*

Workflow management as a discipline has traditionally suffered from a proliferation of process definition languages based on similar, but subtly different, concepts and constructs. After numerous attempts, standardization efforts have converged towards the *Business Process Model and Notation* (BPMN) language (Object Management Group 2011), which is intended for modelling business processes primarily during the analysis and design phases. BPMN has emerged as a standard notation for capturing business processes, especially at the level of domain analysis and high-level systems design.

SBOAT is the Stochastic BPMN Optimisation and Analysis Tool, which allows a user to model, and annotate with rewards and stochastic branching, a business processes as a BPMN Business Process Diagram (BPD). The tool is based on solid mathematical principles and is designed to be easy to use through a graphical interface, see fig. 1, which allows for editing or importing BPMN models with minimal additional training compared to a standard BPMN modelling tool.



Analysis Setup

Representation Engine: Hybrid
Computation Method: Jacobi
Convergence Detection Method: Relative
Over-Relaxation Parameter (Ω): 0.9
Convergence Check Parameter (ϵ): 1e-6
Maximum Number of Iterations: 10000

□ Skip Precomputation Algorithms
□ Automatically add Self-Loops to Deadlock States
□ Check for Absence of Zero-Reward Loops
□ Verbose Logging

Query: Pmin=? [F s=34 & d=5]

```
-------------------- PREPARING MODEL FOR ANALYSIS --------------------
Model Name:die.prism
Engine Arguments: -h -jac -rel -e 1e-6
PCTL Query: Pmin=? [ F s=34 & d=5 ]
-------------------- TRANSLATING MODEL --------------------
BPD elements: 24
Structural Semnatics Check: Vaild  Checking Time: 0.00 seconds
Translating Model...
Done (PRISM Code Updated) Translation time: 0.04 seconds
-------------------- GENERATE STATESPACE --------------------
PRISM Version: 4.1.dev-ScheduleMod-rev78
Hostname: Desktop-PC
Building model...
Computing reachable states...
Reachability (BFS): 8 iterations in 0.00 seconds (average 0.000000, setup 0.00)
Type: MDP  States: 45 (1 initial) Transitions: 79 Time for model construction: 0.034 seconds.
Transition matrix: 240 nodes (3 terminal), 79 minterms, vars: 10r/10c
-------------------- PERFORMING ANALYSIS --------------------
Prob0: 10 iterations in 0.00 seconds (average 0.000000, setup 0.00)
Prob1: 3 iterations in 0.00 seconds (average 0.000000, setup 0.00)
yes = 1, no = 25, maybe = 19
Computing remaining probabilities...
Engine: Hybrid
Building hybrid MTBDD matrix... [levels=10, nodes=192] [4.5 KB]
Adding explicit sparse matrices... [levels=10, num=1, compact] [0.2 KB]
Creating vector for diagonals... [dist=1, compact] [0.1 KB]
Creating vector for RHS... [dist=2, compact] [0.1 KB]
Creating vector for diagonals... [dist=1, compact] [0.1 KB]
Allocating iteration vectors... [2 x 0.4 KB]
TOTAL MEMORY USE: [5.6 KB]
Starting iterations...
Jacobi: 28 iterations in 0.00 seconds (average 0.000000, setup 0.00)
Value in the initial state: 0.111111082136631
Result: 0.111111082136631 (value in the initial state) Time for model checking: 0.003 seconds.
-------------------- ANALYSIS COMPLETE --------------------
```

Figure 1: SBOAT version 0.45 Overall User Interface (enlarged)

Using SBOAT a business practitioner can model production workflows with details like time cost and other constraints such as limits of available resources. Verification of properties of interest of the system can be performed and are calculated for all possible execution paths of the system. In this manner it is possible for a company to experiment with different types of workflows to find the one which maximises throughput while maximising safety parameters (Herbert & Sharp 2014).

The capabilities of the SBOAT tool also allow for the generation of fault trees which produce include failure probabilities which allow for the underlying stochastic elements of a workflow and determine resource consumption at points of failure (Herbert & Sharp 2013b). In addition SBOAT allows for scheduling analysis and resource analysis (Herbert & Sharp 2012). Optimisation can also be achieved with regard to properties described using a rich specification language. In this case SBOAT searches for possible improved restructurings of a workflow, by means of an evolutionary algorithm (Herbert & Sharp 2013a).

2.2 *Theoretical Underpinnings*

SBOAT employs principles from stochastic model checking (Baier & Katoen 2008) to allow for the automated analysis of workflows. The workflow is converted to an algebraic description for which a statespace, describing all possible system the workflow may assume, is generated. This approach differs from simulation in that all possible states are examined which may not be the case after any finite amount of simulation.

Analysis in SBOAT is based on efficiently checking each state in the statespace against a temporal logic (Baier & Katoen 2008) formulae expressing properties of interest. These are encoded in *Probabilistic*

Computation Tree logic (PCTL) allowing determination of probabilities of specific execution paths. For example, given a model M the expression:

$$M : Pr_{=?}[\mathsf{F}(\text{temp} > 10^\circ \wedge \text{t} < 100s)\mathsf{U}(\text{valve}_A = \text{true})]$$

would identify the probability (Pr), during execution, of arriving in a state where a temperature quantity exceeds 10° in less than the first 100 seconds of execution followed by a specific valve A being opened.

In general the following operators, further described in (Herbert & Sharp 2014), may be used to build analysis queries:

- $\mathsf{X}a$ The *next* operator is a unary operator that specifies for a path that a given property a holds in the path's next state.
- $a\mathsf{U}b$ The binary *until* operator specifies that, for a given path, in some state of the path, the property b is true and in all preceding states the property a is true.
- $\mathsf{F}a$ The unary *eventually* operator specifies that, for a given path, a eventually becomes true at some point along the path.
- $\mathsf{G}a$ The unary *always* operator specifies that, for a given path, a is true in all states along the path.

The overall software structure of SBOAT shown in fig. 2.

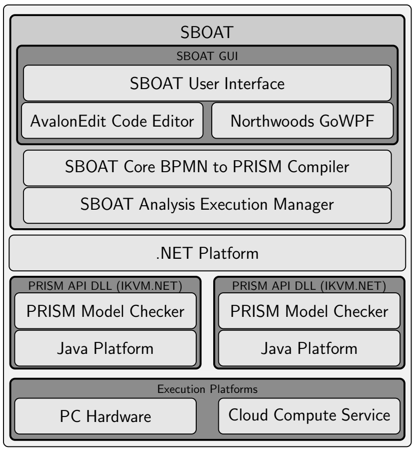


Figure 2: Architecture of the SBOAT tool and environment.

Using SBOAT involves constructing an annotated Core BPMN model, defining the analysis one wishes to perform and then running the analysis. It may be desirable to set a number of analysis optimisation settings first if one expects the analysis to be particularly complex and it is also possible to distribute checking across multiple compute resources.

The framework's modelling language includes the tracking of freely defined real-valued quantities associated with the process (such as time, cost, and temperature). In addition, this formalism also allows for an intention preserving stochastic semantics able to model both probabilistic- and non-deterministic branching behaviour. These may be combined such that a systems make make a non-deterministic choice where each decision then has different probabilistic outcomes. In SBOAT we further extend this formalism to allow for the introduction of error states which allow for both fail-stop behaviour and continued system execution (Herbert & Sharp 2014).

Stochastic model checking allows one to efficiently explore the entire state space of a workflow. The checking algorithm allows for the weighted generation of PCTL queries that can be used to express a desired balance between the occurrence of errors and data quantities associated with the workflow. It should be noted here that the algorithm performs exhaustive generation of all possible states, including error states, that could arise during execution and therefore our method can determine the probability of complex events, such as combined faults, while accounting for the basic probabilistic structure of the system being modelled. Combined this allows for the expression of queries which identify the factors in operations which have the largest impact on the state being reach; for example which step in the production workflow contributes the largest safety risk and may be the point at which an additional control should be implemented.

In SBOAT these queries along with a model of an existing workflow are used as inputs to an evolutionary algorithm which iteratively, through a process of mutation and cross-over, generates candidate improved workflows (Herbert & Sharp 2013a). The stochastic model checking of a weighted set of queries is used as a fitness function for determining the degree of improvement of candidate workflows. Being an evolutionary algorithm, when a candidate workflow shows improvement it is used as the basis for the next round of mutation and cross-over. Further, we allow this evolutionary procedure to be constrained such that candidates can also be eliminated based on constraint requirements also expressed in PTCL (these would typically encode fundamental properties of the workflow such as actions that must be performed prior to other actions or safety constraints that no acceptable workflow will be allowed to violate). Further, the constraints allow the removal of evolutionary dead-ends at an early stage, a necessary step to achieving acceptable performance in a practical implementation of this algorithm.

3 SAFETY BARRIER MANAGER (SBM)

3.1 *Tool Description*

SBM supports the creation of valid safety-barrier diagrams and it allows the quantitative analysis of the probability of all possible end states of the barrier diagram, i.e. the outcomes if one or several of the barriers fail to perform their barrier function. The tool provides a graphical interface for drawing the elements of the diagram (safety barriers, events or conditions and other logical operators, such as logical gates and event tree branches). Safety barriers may consist of sub-elements that can be shared between several barriers, and that thus represent dependency between barriers.

Quantification of the safety-barrier diagram is performed by transferring the logical structure of the diagram into a fault tree notation, which is then solved using a Binary Decision Diagram (BDD) algorithm. The BDD algorithm allows the use of inverse logic (necessary to both describe events on failure and on success of a barrier).

Performance specifications and management requirements can be added to the description of the safety barriers to support inspection, maintenance and life-cycle management. The tool has the option to include the quality of management, as assessed by means of audit and inspection, into the quantitative probability assessment using the techniques developed during the ARAMIS project (Duijm & Goossens 2006) (Guldenmund, Hale, Goossens, Betten, & Duijm 2006).

Results can be presented by means of risk matrices in order to assess the tolerability of certain out-comes in terms of risk.

3.2 *Theoretical Underpinnings*

SafetyBarrierManager® (SBM) is a tool to describe and analyze safety barrier diagrams (Duijm 2008), (Duijm 2009), (Duijm 2007). A safety barrier diagram is a variation of a cause consequence diagram. These diagrams de-scribe the temporal evolution of an accident sequence (an accident scenario) with a focus on the actions and measures that are foreseen to: a) prevent the deviations that cause the accident, and b) mitigate the consequences of the accident. Such actions and measures are named “safety barriers” (see fig. 3).

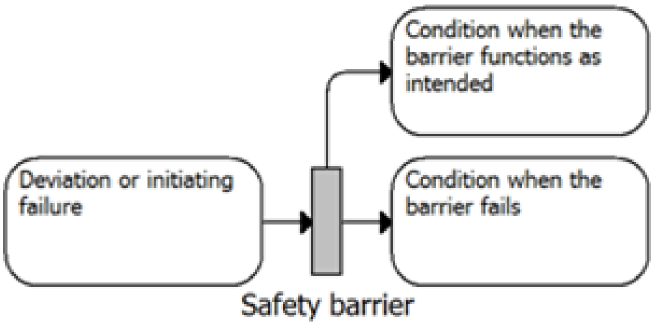

Condition when the barrier functions as intended
Deviation or initiating failure
Condition when the barrier fails
Safety barrier

Figure 3: Safety Barrier

Safety barriers implement safety functions, where a safety function is a function designed to abort the development of an adverse situation.

Safety barriers are graphically presented as in Figure 1. To the left of the barrier is the condition that triggers the barrier function (often an initiating failure condition); to the upper right the condition when the barrier correctly deals with the situation and to the right (i.e. “through” the barrier) the condition when the barrier fails to perform its barrier function. From a point of view of risk analysis, the main important property of a safety barrier is its probability of failure on demand (PFD). The safety strategy is to implement several safety barriers in order to create redundancy and minimize the probability of an accident. Complicating issues that should be considered are: a) common cause failures or dependencies between barriers that reduce the redundancy; and b) the necessity of maintaining the PFD of all the safety barriers during the life time of the system. This requires management of safety barriers. Note that safety functions can be performed by hardware, software and human action. Safety-barrier management thus covers management of design and maintenance of hard and software as well as human re-source management. Safety-barrier diagrams are developed by further analysis of causes and consequences of hazards identified using hazard identification techniques such as FMEA, HazOp or HACCP. Such analysis is similar to fault tree analysis (finding sufficient and necessary conditions for adverse events) and event tree analysis (analyzing the possible outcomes of an adverse event). Such analyses are mainly manual desk-top exercises using detailed knowledge of the system in question.

4 COMPARISON OF THE TWO TOOLS

SBM focusses on accident scenarios and does not address the analysis of normal (successful) workflows. SBM does not automatically create failure cases based on a system description, the analyst has to identify those cases himself, and also describe the events following the failure or success of a safety barrier. However the tool is highly efficient and check even large models quickly. The output can take the form of industry standard risk matrices.

In contrast SBOAT can model workflows using a standard notation which are both successful and those which contain errors or are otherwise not optimal. The user has to define analysis queries for the workflow, e.g. what is a failure state, using a somewhat complex syntax. The output takes the form of probabilities of specific events or an optimised system which seeks to minimise properties of interest.

The following table compares key parameters of the two tools:

Feature	SBM	SBOAT
Easy of use	Average	Average
Precision	Exact (based on scenarios)	Exact (based on computations)
Performance	Fast	Slow
Feature sets	Narrow	Wide

Table 1: Comparison between the two tools

As can be seen from table 1 each tool is best suited to a particular context. In general, SBOAT is more precise and detailed and makes it easy to directly convert a workflow written in commonly used business process languages like BPMN at the expense of considerable computational requirements. However, SBM is a faster tool which quickly gives the user an overview of hazard workflows, without the intermediate step of detailing a normal workflow.

5 INDUSTRY APPLICABILITY

5.1 *Case Description*

We will use a workflow description from the food industry to illustrate the two tools. The food industry has several safety demands; for example safe working conditions for employees, traceability and hygiene requirements to ensure the food is safe for consumption. The example used here is from a slaughter and processing plant for Danish chickens. Only a selection of the process is shown here.

Figure 4 shows the processing workflow as described the case company, meaning that in the figure it is assumed that the safety barriers hold. When the chickens have been slaughtered and the feathers removed a veterinarian checks them. If they pass this health check the chickens continue to a scanning which scans for "hard breast syndrome" which means the breast meat cannot be eaten by humans as it is too hard and thus needs to be removed and used for pet food processing. If the scanning shows the breast meat is ok the chicken is weighted. If it weighs over a given limit an amount of chickens is selected to be sold as whole chickens. The rest are processed in selected packages with bigger portions. If this limit is not met the chicken is processed into the selected pieces the plant needs to fulfil its orders (for example breast meat, drumsticks, hearts, liver, wings, minced meat).

The entire process is automated; chickens hang on a line like clothes hanging out to dry and get carried around the plant as they pass through each step. The chicken starts out hanging on the line as one whole chicken and if the chicken is not sold as a whole chicken it is cut into selected pieces over a number of steps (for example one step cuts off the feet, one the wings, one the drumsticks etc.) until the whole chicken has been cut into smaller pieces.

Figure 5 shows an example of the many barriers which are built into the process to try and prevent a failure state, meaning what happens if a failure happens and is not detected. In this example it is the checks and consequences if a chicken with illness falsely get processed as healthy (note, this example show only a small selected of the process and has been simplified for illustration purposes). Here it should be kept in mind that if one chicken is ill, the whole herd is likely ill too and in any case the whole herd needs to be destroyed for safety reasons. Furthermore, illness can be tied to a certain chicken farmer which means all chickens from that farmer could be infected. This means that meat from that farmer, which can arrive with days or weeks in between, would need to be investigated. At the slightest doubt all meat from that farmer within a certain timeframe determined by veterinarians (depending on the illness in question) will need to be destroyed. It is therefore vital that chickens which contain illness are discovered and destroyed as quickly and early in the process as possible to prevent huge loss (e.g. loss in terms of production time, transport and potential loss of customers or decreased brand value should illness be connected to the chicken producer).

As can be seen then the later the failure is detected the higher the risk and cost of the failure. If the meat from an ill chicken makes it to a client and the client gets ill the consequences are not only financial but can also result in legal action. For this to be possible to chicken needs to have either passed or not been selected for random checks

5.2 *Using Safety Barrier Manager on the case*

To use SBM on the case it is necessary to re-draw the workflow in the style of SBM, illustrating the safety checks. The tool does not concern itself with normal flows; only with what happens in hazard situations.

The case includes three checks for different adverse situations: firstly a vet checks randomly selected chickens from each batch (meaning from each farm and flock that enters the plant) for infectious disease and bacteria which, if gone unnoticed can result in illness in humans of a more or less severe degree, chickens with "hard breasts", and chickens too small to be sold as whole chickens. This can be represented by 3 barriers. The success is the proper action to re-move the chicken for further processing, while the "failure" of the barrier is the processing of unsuitable chickens, see fig. 6.

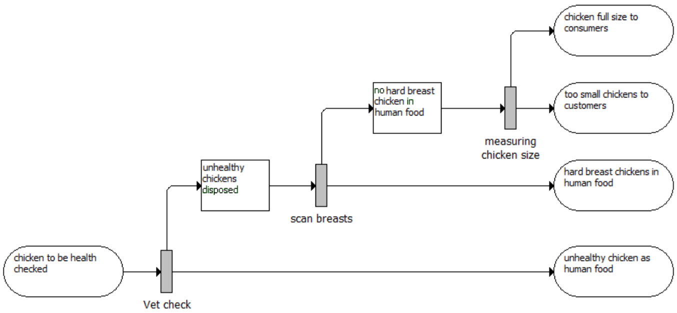

chicken full size to consumers
no hard breast chicken in human food
too small chickens to customers
measuring chicken size
unhealthy chickens disposed
hard breast chickens in human food
scan breasts
chicken to be health checked
unhealthy chicken as human food
Vet check

Figure 6: SBM diagram of the chicken process safety functions.

Using SBM on the case gives the user a quick overview of the safety precautions they currently have in case. The output can be used as input for debate regarding the adequacy of the current process and whether any of these safety functions can be improved. However, there is some work required to in-

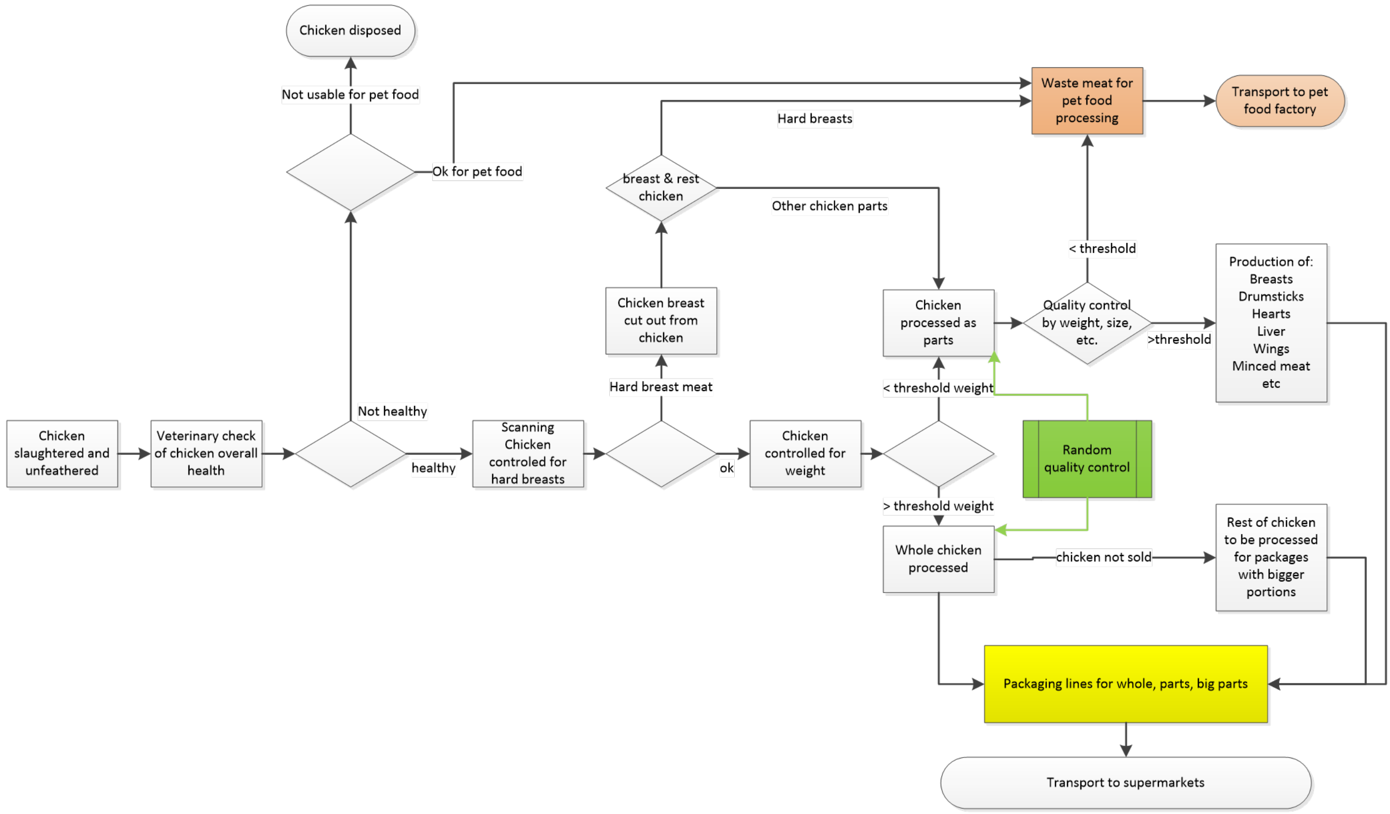

Chicken disposed
Not usable for pet food
Ok for pet food
Waste meat for pet food processing
Transport to pet food factory
Hard breasts
breast & rest chicken
Other chicken parts
< threshold
Production of: Breasts Drumsticks Hearts Liver Wings Minced meat etc
Chicken breast cut out from chicken
Chicken processed as parts
Quality control by weight, size, etc.
>threshold
Hard breast meat
< threshold weight
Not healthy
Chicken slaughtered and unfeathered
Veterinary check of chicken overall health
healthy
Scanning Chicken controled for hard breasts
ok
Chicken controlled for weight
Random quality control
> threshold weight
Whole chicken processed
chicken not sold
Rest of chicken to be processed for packages with bigger portions
Packaging lines for whole, parts, big parts
Transport to supermarkets

Figure 4: Normal workflow in a chicken processing plant.

put the workflow as SBM needs the user to draw the workflow using safety barriers and not commonly used workflow description tools.

5.3 *Using SBOAT on the case*

To use SBOAT we first model the process in the tool, annotating it with rewards and constraints as shown in fig. 7. These include resource usage, for example the time it takes for each processing step to complete, and decision outcome probabilities.

SBOAT takes as input the normal production model with minimal modifications. The user needs only annotate the process with rewards and restraints as illustrated in fig. 7. The resulting model is useful in several ways. It can be used as a basis for debate about the current workflow as well as for the current hazard steps and current precautions installed in the process to prevent these. Once a point of interest in the process is found queries can be constructed to determine the probabilities of a range of values of the annotated quantities at that point. Further points and execution paths can be “discovered” using queries, such as all paths that lead to a system deadlock.

In this manner the tool is useful in that a debate regarding risk management often needs to take the normal workflow as a basis and weigh efficiency, probability and risk as well as the impact of a possible precaution against a risk into consideration before implementation.

Further SBOAT may be used to suggest improved workflows when a unacceptable risk has been discovered.

6 DISCUSSION

The two tools have specific strengths and weaknesses which makes each of them suited to various context. SBM is a useful tool if only hazard situations are of interest and there is a need to get a quick overview of the current precautions. SBOAT, however, is useful if also the normal workflow is of interest and if a more detailed picture of the workflow quantitative properties is desired.

Ideally these two tools should be used together to analyse the same workflow as they complement each other. They show different aspects of the case. SafetyBarrierManager® shows where safety barriers currently are and can be used to analyse whether these adequately cover all potential risk situations. SBOAT can be used to give a complete and detailed overview of the current workflow, including the normal state. The output from both tools can help organisations improve not only their risk management process but only ensure these steps are optimal in terms of the normal operating state.

We would suggest using the two tools as follows:

1. Draw the current process in SafetyBarrierManager®. Include all safety barriers.

2. Use SafetyBarrierManager® to analyse the current flow and safety barriers. Are these adequate?

3. Formulate desired parameters, for example time or temperature and constraints, for example that step A must take place before step B.

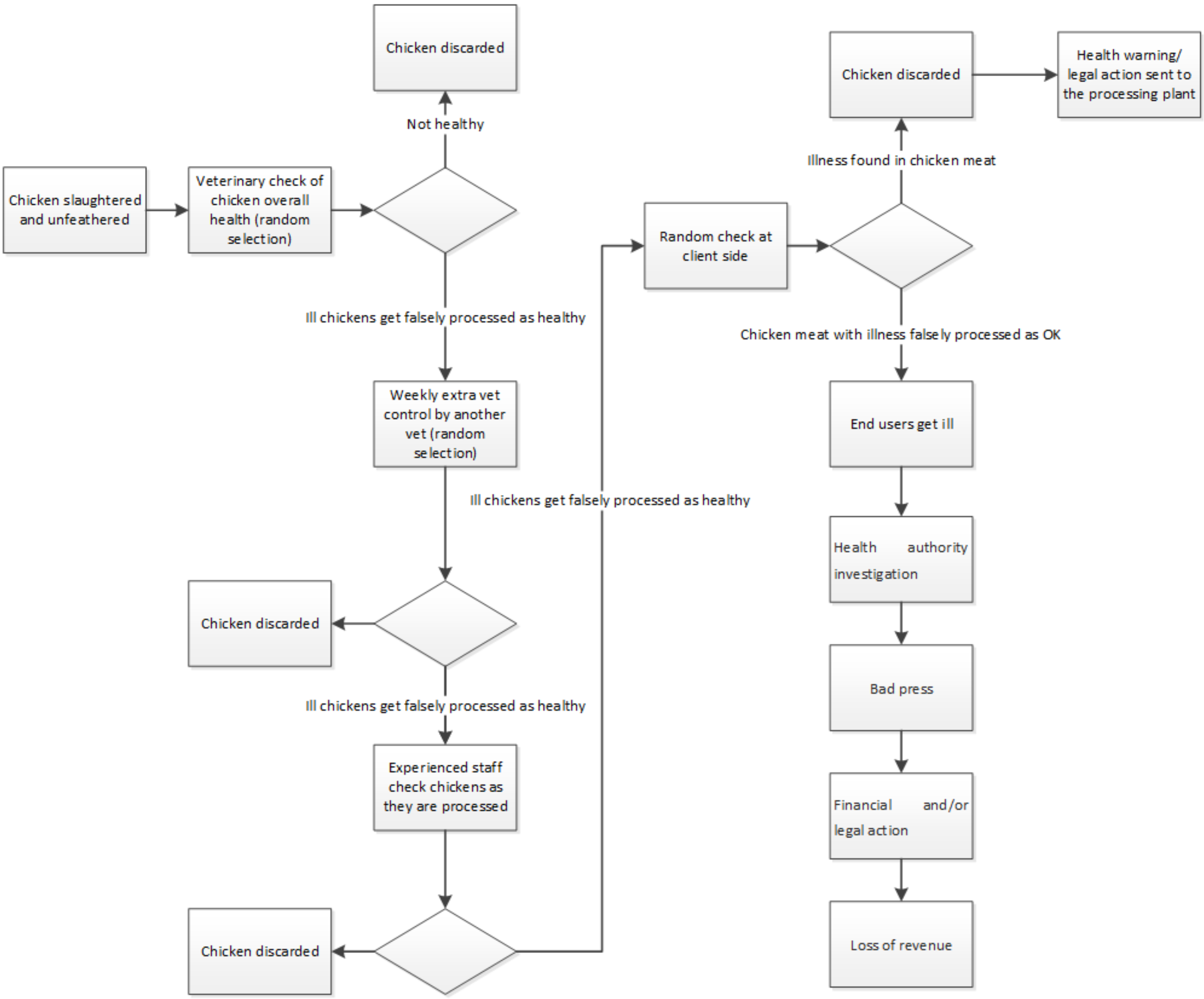

Chicken discarded
Not healthy
Chicken slaughtered and unfeathered
Veterinary check of chicken overall health (random selection)
Ill chickens get falsely processed as healthy
Weekly extra vet control by another vet (random selection)
Ill chickens get falsely processed as healthy
Chicken discarded
Ill chickens get falsely processed as healthy
Experienced staff check chickens as they are processed
Chicken discarded
Random check at client side
Chicken discarded
Health warning/ legal action sent to the processing plant
Illness found in chicken meat
Chicken meat with illness falsely processed as OK
End users get ill
Health authority investigation
Bad press
Financial and/or legal action
Loss of revenue

Figure 5: Barriers and workflow in a chicken processing plant concerning discovering illness in the chickens being processed.

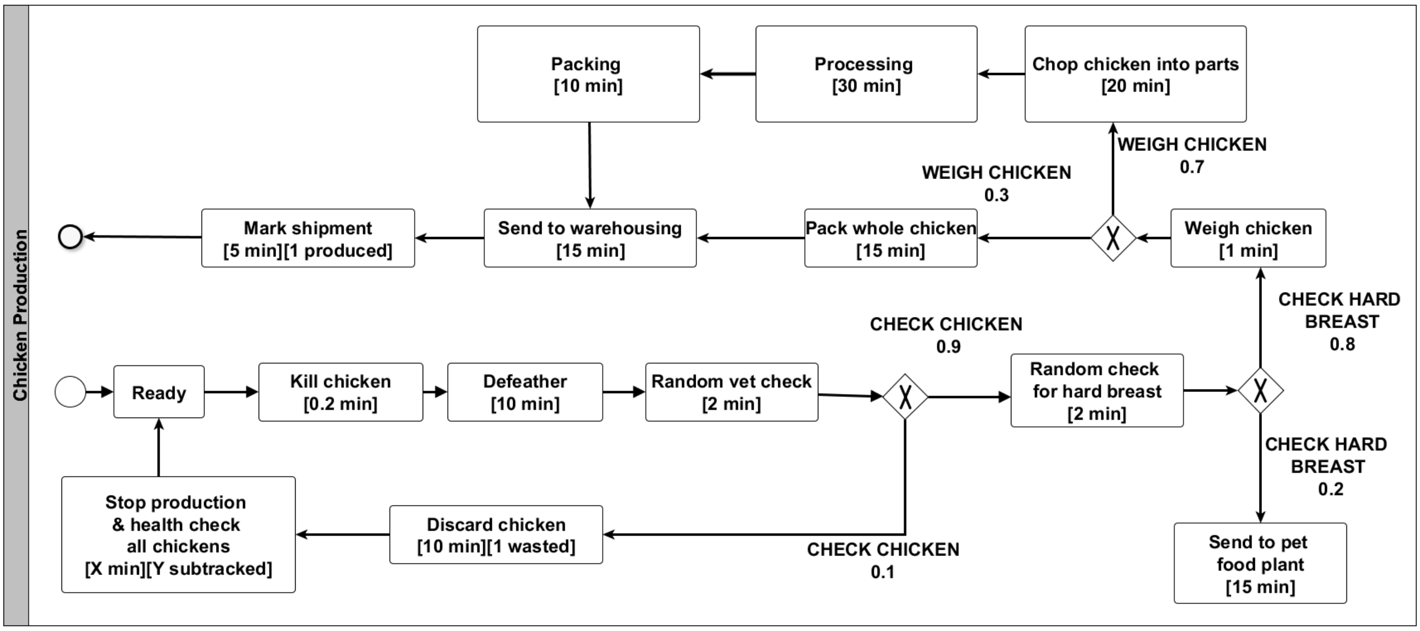

Chicken Production
Packing [10 min]
Processing [30 min]
Chop chicken into parts [20 min]
WEIGH CHICKEN 0.7
WEIGH CHICKEN 0.3
Mark shipment [5 min][1 produced]
Send to warehousing [15 min]
Pack whole chicken [15 min]
Weigh chicken [1 min]
CHECK HARD BREAST 0.8
CHECK CHICKEN 0.9
Ready
Kill chicken [0.2 min]
Defeather [10 min]
Random vet check [2 min]
Random check for hard breast [2 min]
CHECK HARD BREAST 0.2
Stop production & health check all chickens [X min][Y subtracked]
Discard chicken [10 min][1 wasted]
CHECK CHICKEN 0.1
Send to pet food plant [15 min]

Figure 7: Quantitative safety analysis of the chicken process.

4. Use SBOAT to draw the current process, using the input parameters found in step 3. Debate the output. Is the current normal state as desired? Could the risk management steps be improved?

5. Discuss and suggest potential desired improvements to the current workflow, within the given restraints. Use SBOAT to draw up these different scenarios.

6. Evaluate the potential improved workflows outputted from SBOAT. Draw these in SafetyBarrierManager® to see whether safety barriers have remained the same, improved or worsened the risk implications.

7. Decide on a new workflow to use which fulfils the restraints defined in step 3 and has at least the same, ideally more or improved, safety barriers as shown in the SafetyBarrierManager® tool.

By using SafetyBarrierManager® and SBOAT together it is possible to create a workflow which is both improved according to safety parameters as well as other desired parameters. The main drawback at the moment by using both tools in the same workflow is that the tools use different inputs so that the user need to input parameters in both tools.

7 CONCLUSIONS AND NOTES FOR FURTHER RESEARCH

This paper investigated two graphical tools for workflow improvement which can help reduce errors; SBOAT (Stochastic BPMN Optimisation and Analysis Tool) and SBM (Safety Barrier Manager).

While these tools can yield significant safety improvements when used on their own it is possible, when using them on the same case, to produce an improved workflow both in terms of safety barriers and in terms of other desired parameters for the given workflow.

Future research will focus on further testing the applicability of these two tools on the described case in further detail. The use of both uses on cases from other industries is also of interest in order to compare results across industries.

Furthermore, we would like to investigate the possibility to combine the tools in one user interface, so that the user can use each tool separately or together but only need to use one program. A part of this would be that the user only need to input all choices and descriptions, including safety barriers, rewards and constraints, once.

REFERENCES

Baier, C. & J.-P. Katoen (2008). *Principles of Model Checking*. Cambridge MA, USA: The MIT Press.

Duijm, N. J. (2007, September). Safety-barrier diagrams. In *European Safety and Reliability Conference*, pp. 9–16.

Duijm, N. J. (2008). Safety- barrier diagrams. *Journal of Risk and Reliability 222*, 439–448.

Duijm, N. J. (2009). Safety-barrier diagrams as a safety management tool. *Reliability Engineering & System Safety 94*, 332–341.

Duijm, N. J. & L. Goossens (2006). Quantifying the influence of safety management on the reliability of safety barriers. *Journal of Hazardous Materials 130*, 284–292.

Guldenmund, F., A. Hale, L. Goossens, J. Betten, & N. J. Duijm (2006). The development of an audit technique to assess the quality of safety barrier management. *Journal of Hazardous Materials 130*, 234 – 241.

Herbert, L. & R. Sharp (2012, October). Using stochastic model checking to provision complex business services. In *High-Assurance Systems Engineering (HASE), 2012 IEEE 14th International Symposium on*, pp. 98–105.

Herbert, L. & R. Sharp (2013a, August). Optimisation of BPMN business models via model checking. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2013)*, ASME Conference Proceedings.

Herbert, L. & R. Sharp (2013b). Workflow fault tree generation through model checking. In *Proceedings of the 2013 European Safety and Reliability Association ESREL conference*.

Herbert, L. & R. Sharp (2014). Model-checking business processes. In *Advances in Computational Sciences and Information in Engineering*, ACIER book series. ASME Press. (Accepted for publication).

Object Management Group (2011, January). Business process model and notation (BPMN) 2.0. Standards Document formal/2011-01-03, Object Management Group, Needham MA, USA.

The European Agency for Safety and Health at Work (2015). Risk assessment tools database. `https://osha.europa.eu/en/practical-solutions/risk-assessment-tools/index_html`. Accessed: 2015-04-02.