A Discrete Event Simulation Approach for Quantifying Risks in Manufacturing Processes

Renaud de Landtsheer[1], Gustavo Ospina[1], Philippe Massonet[1], Christophe Ponsard[1], Stephan Printz[2], Lasse Härtel[3] and Johann Philipp von Cube[3]

[1]*CETIC Research Centre, Charleroi, Belgium*

[2]*Institute for Management Cybernetics (IfU), RWTH Aachen University, Aachen, Germany*

[3]*Fraunhofer Institute for Production Technology (IPT), Aachen, Germany*

Keywords: Discrete Event Simulation, Manufacturing, Supply Chain, Procurement Risks, Risk Management.

Abstract: Nowadays supply chains have to face an increasing number of risks related to the globalisation, especially impacting the procurement processes. Even though tools are available to help companies in addressing those risks, most companies, even larger ones, still have problems to adequately quantify the risks and assess to what extend an alternative could address them. The aim of our work is to provide companies with a software supported methodology to quantify such risks and elaborate adequate risk mitigation strategies at an optimal cost. Based on a survey conducted about the risk management practices and needs within companies, we developed a tool that enables a constant focus on risks by enabling the easy expression of key risks together with the process model and hence help to focus the granularity of the model at the right level. A model-based simulator can then efficiently evaluate these risks thanks to well-known Monte-Carlo simulation techniques. Our main technical contribution lies in the development of an efficient discrete event simulation (DES) engine together with a query language which can be used to measure business risks based on simulation results. We demonstrate the expressiveness and performance of our approach by benchmarking it on a set of cases originating from the industry and covering a large set of risk categories.

1 INTRODUCTION

Companies are faced with increasing procurement risks in the context of globalisation. Those risks can be related to many different factors such as the geographic location, the political and economic situation. Assessing those risks alone is also a difficult task as the risks can reveal themselves at the end of the production chain and it requires also to consider to some extend the impact of internal risks such as the complexity of the manufacturing process (which could decrease the capacity to adapt to a supplier failure) or the level of optimisation in place (which would rise the impact in case of disruption).

Helping company managers to take the right decisions in the presence of such risks is not an easy task. Analytic reasoning is quickly impractical and model-based simulation has proved a very relevant approach (Deleris and Erhun, 2005). Procurement risks put extra challenges as they occur at one end of the process but can sometimes only be measured at the other end, so they require to embrace the whole manufacturing process. Addressing this challenge is precisely the scope of our work, with a focus on small and medium enterprises in the field of mechanical engineering.

Our ultimate goal is to produce a user-friendly tooled methodology that will guide the user through the whole process of risk assessment. In order to reach this goal, our work is structured as follows:

- First, a taxonomy of supplier and internal risks has been identified, starting from the simplest risk of shortage of raw material, which can eventually drain the whole process chain to more elaborated risks related to the kind of order policy used.
- Second, a survey was conducted on the state of practice of risk evaluation in industrial context (Printz et al., 2015b). The results of this survey showed that nearly 66% of the companies perform risk evaluation, although only 10% rely on dedicated software tooling. This means that in practice risks are evaluated by an individual estimation of the cost factor and the probability of occurrence. In general, estimation quality increases by includ-

Landtsheer, R., Ospina, G., Massonet, P., Ponsard, C., Printz, S., Härtel, L. and Cube, J.
A Discrete Event Simulation Approach for Quantifying Risks in Manufacturing Processes.
DOI: 10.5220/0005752403130322
In *Proceedings of 5th the International Conference on Operations Research and Enterprise Systems (ICORES 2016)*, pages 313-322
ISBN: 978-989-758-171-7

ing historical data in the estimation. However, relying on historical data and estimating the impact, like delivery timings, quality of the materials is not possible either in the case of changing suppliers or adding parallel processes in the chain. Based on the requirements identified in the conducted survey a software based risk management framework has been defined.

- Third, we developed a modelling and simulation toolset to identify risks, quantify them and decide on design alternatives that can help to mitigate them. The main technical scope of the present paper is to detail our framework and show how it helps focusing the modelling on the risks to stay efficient in the modelling time, simulation time and result analysis time.
- Fourth, we are also validating our work with a group of companies that are already trying our tool through an easy to use web interface. Although this validation is not yet complete, we could already benchmark our approach on a number of industry cases and assess the expressiveness and performance of our approach.

Our modelling framework includes concepts such as *storages* where items can be stored or retrieved with a maximum capacity, as well as several types of production *processes* with different timing and failure behaviours. In addition, we defined a query framework on models that is fully declarative and includes arithmetic, temporal and logic operators as well as basic probes on the elements of our factory model (contents of a storage, whether a process is running or not, etc). Based on this query language, the toolset is able to calculate the probabilities of different scenarios (e.g. delay in deliveries, defective parts or poor quality) and their impact, based on a timed model of the considered factory processes.

The approach of monetary risk quantification is based on an approach developed in the Q-Risk project (von Cube et al., 2014). The simulation toolkit relies on the discrete event simulation module of the OscaR framework for its base simulation layer, and adds dedicated abstractions, dedicated to the timed modelling of factories, and the modelling of risk-related queries (OscaR, 2012).

Our main contribution lies risk-driven dimension of our framework but also to the attention to usability. Its design is based on a number of trade-offs between expressiveness and simplicity of the modelling language, as well as efficiency of the simulation engine.

The paper is structured as follows: section 2 presents the context of our work; section 3 presents our modelling language for representing factories; section 4 presents our query language that can serve to evaluate risks; section 5 illustrates how complex risks can be included in our query language; section 6 shows the benchmarking of our simulation tool both on the expressiveness and performance dimensions; section 7 discusses some related work; section 8 concludes the paper.

2 BACKGROUND

In order to assess and quantify different kind of risks in manufacturing processes, we model the manufacturing process as a flow graph. This models captures both the key procurement step but also the production process itself. In particular, the resource storage place (like warehouses or stockrooms) and the raw materials flow through basic processes will be explained in section 3.1. The main graphical notations implemented by the graphical part of framework are shown in Figure 1 which is a model used later in our benchmarking. Notations are quite self explanatory: a supplier is a little truck, storage types are represented by different variant of cylinder (the one with vertical bars can overflow) and processes are depicted with the industry icon (also with some variants: multiple horizontal lines means parallel batches, the cross means possible failure, the rounded, the rounded box depict a conveyor belt).

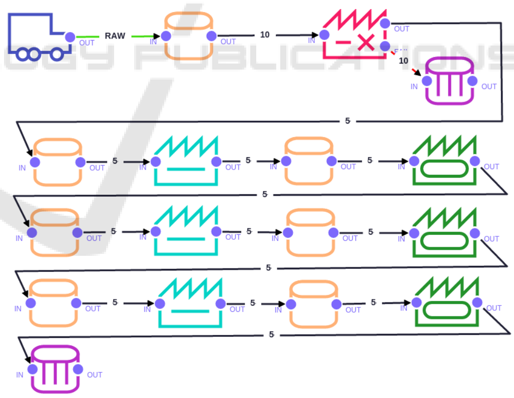

Figure 1: Beer game model.

The operation of the whole manufacturing process can be described as a sequence of timed events. For instance, in a simulation of a single factory, the first events are fetching some materials of a storage that will be worked by a process. This fetch can trigger a new order to a supplier if the storage level reached some threshold, according to a supply chain policy.

In the rest of this section, we will first remind the reader about the nature of risks and the goal of risk management, then we will give some details about Discrete Event Simulation and why it is an adequate

framework to model the operations of manufacturing processes and gather relevant data to quantify risks (Romeike, 2004).

2.1 Risks and Risk Management

In order to develop a Discrete Event Simulation approach to quantify the impact of risks in manufacturing enterprises, the nature of risk and the underlying process of risk management needs to be understood in detail.

2.1.1 Risk

Risks strongly affect enterprises business success directly being related to costs, effort and yield (Zsidisin and Ritchie, 2009). Thereby, risk is understood as an event likely to occur with an undesired consequence. The most common and for the approach most convenient categories of risks are the cause and impact-oriented definition. The root-cause-oriented approach considers uncertainty of information validity as risk (Siepermann, 2008). Considering the chance not meeting a planned target is understood as impact-oriented risk definition. However, only combining both categories of risk lead to the necessary scope of information needed to properly manage risks. Hence, risk needs to be understood as certain likelihood to miss a defined target. Hence, the concept of risk is defined through three components: the *hazards*, or potential dangers, the *consequences* of those hazards, and their predicted frequency, or *likelihood* (Sutton, 2015). A "natural" quantification of the hazard associated to a risk is the product of all the quantified consequences by the likelihood. A cybernetic model of procurement based hazards and their management is presented in (Printz et al., 2015a).

Risk likelihoods can be modeled with probability distributions (Artikis and Artikis, 2015), as the occurrence of a risk hazard in a process or system is naturally *uncertain*. In (Zio, 2013), a theory of probabilistic risk analysis is developed, associated to the concept of system reliability. As risk is defined as the deviation from a planned value, statistical measures can thus be applied to operationalise and compare the possible magnitude of such deviations (Gleißner, 2012). Evaluation of the risk analysis and the reliability of a system can be done with Monte-Carlo methods (Deleris and Erhun, 2005).

2.1.2 Risk Management

The main objective of risk management lies in the assurance of major corporate goals under consideration of risk policy strategies. Hence, risks affecting long lasting business success need to be controlled. However, enterprises will never be able to totally eliminate risks and will always have to consider a certain degree of residual risk (Finke et al., 2010). One key task of risk management is to identify and analyze risks as early as possible to take cost optimal risk treating actions (Zsidisin and Ritchie, 2009).

The basic process of risk management (Figure 2) is described in the standards ISO 31000 and ONR 49000 ff. IEC 31010 provides an overview of corresponding risk management methods and techniques along the process.

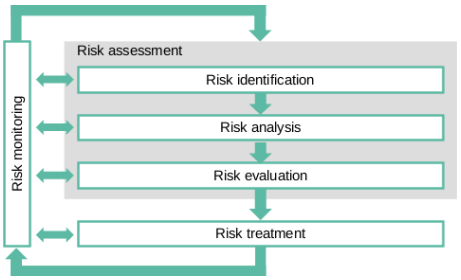

Figure 2: Risk management process.

2.2 Discrete Event Simulation

There are two main approaches for representing the time if we want to simulate the behaviour of a system: the first approach is to use a *continuous time*, in which the events affecting the system occurs in time "ticks" which are proportional to the actual expected time of operation for the system. The other approach is to have a *discrete time*, and concentrate the simulation only in the operational events instead of the time events. This is the basis of Discrete Event Systems (DES).

In the literature (Brailsford et al., 2014; Byong-Kyu and Donghun, 2013), the main components of a DES model are described as: *entities*, which are the items that are flowing and transformed through the simulation, *queues*, representing storage devices or other areas in which entities wait to be used, *activities*, that actually perform some work on the entities, and *resources*, a special kind of entities that are required to operate activities.

DES models define *events* as discrete points of time in which the system state changes. The simulation of the model becomes the checking of a queue of the different events triggered, the "next-in-time" at the first place. Checking an event can trigger other events in the queue. For instance, checking the event of starting an activity will trigger the events of fetching the corresponding entities needed to perform the activity, and ending the activity. The event of activity failure can also be triggered with a given probability.

Several software solutions exist to support DES based modelling for a variety of applications. Among the commercial software, we can cite AnyLogic (AnyLogic, 2015), Arena (Rockwell Automation, 2015) and Plant Simulation (Siemens, 2015).

3 A SIMULATION META-MODEL FOR FACTORIES

All the main elements of manufacturing processes are represented in our simulation meta-model, which allows us to define concrete models that are simulated in a Discrete Event Simulation engine. In addition to this, we designed a query language over concrete simulations in order to collect and analyse data.

3.1 Modeling Factory Processes

This section introduces the basic blocks for representing factories. In our approach, factories are modeled as flows of items through processes and stocks.

Storages represent any kind of stock device or room place for raw materials, like a warehouse, a barrel, a silo or a dumpster. They have a maximum capacity. When this capacity is reached, they either overflow, or block the upfront processes, depending on the setting of the storage. If a full stock can overflow, any unloading material on that stock is lost.

Batch processes are factory processes that work in a batch fashion; supplies are collected from various stocks, then the process runs for some time, and finally the produced outputs are dispatched to their respective stocks before this whole cycle starts again.

Continuous processes are factory processes that typically run on a conveyor belt. Items are continuously picked from input stocks and undergo the process immediately on a physical end of some machine, pass through the machine in a queue, and when they reach the other end of the machine, the resulting items are dispatched to their respective stocks. A simple example is a conveyor belt that passes through a bakery oven; raw pastries are set on one end of the conveyor belt; they go through the oven and are cooked when they reach the other end of conveyor belt where then are dispatched to their output storage.

Splitting processes are similar to batch processes, except that they have several sets of outputs and when it completes, one set of output is selected and the produced items are dispatched to the stocks associated to the selected output. This represents a quality assurance process whose items flow is split into two (or more) separated flows, based on the result of the quality assurance analysis.

Parallel processes are variant of the above processes where several lines of the same process are running in parallel. Basically, all processes introduced here above have a parameter specifying the number of process lines running in parallel.

Items flowing in processes and stocks are indistinguishable at a given point of the factory, since they all share the same part number. Yet, they have some intrinsic features: some items might come from a given process, others might be made out of poor quality supplies, etc. These intrinsic features can influence on the behavior of some processes, such as the splitting process representing a quality assurance process. This notion of intrinsic features lead us to distinguishing between two different types of storage, namely: First In-First Out (FIFO) storage and Last In-First Out (LIFO) storage.

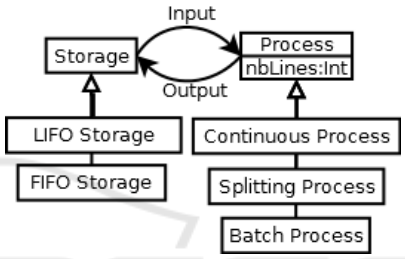

Figure 3: Concepts of our process modelling languages.

3.2 Process Activation and Supply Chain Policies

Supply chain policies are also integrated in our model of the factory, together with activation policies that are able to turn a process on or off, depending on the demand for the output stock. To model these two concepts, we introduce the notion of *activable* and *activation*. An activable is something that can be enabled through an activation. We also associate a magnitude with the activation, that is, an integer. An activable can be a process, or a supply order. In the case of a process, the activation represent the number of batch that the process is allowed to execute. In the case of an order, the magnitude represent the number of ordered items.

In our model, an order is a stationary activable object that represents a class of order that can be passed. The order is passed when the modeled order object is activated.

Activables can be activated based on various rules that are also part of our modelling framework. There are three types of activation rules, namely: regular activations that perform the activation on a regular basis, based on a period of time; order-based activations that perform the activation when an order is received; and

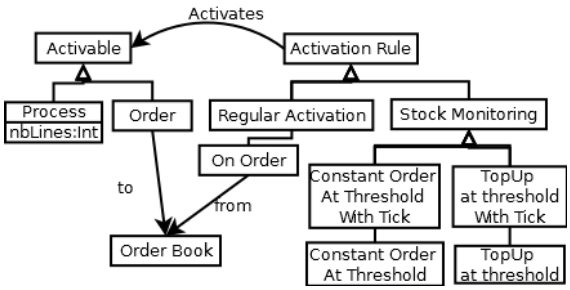

Figure 4: Concept for modelling activation rules.

stock monitoring activation that perform the activation when the stock level gets below some threshold.

3.3 Modelling Intrinsic Item Features

To represent these intrinsic features, we introduce the notion of *item class*, representing the set of intrinsically identical items. Item classes are characterized by a set of boolean *attributes*. A global set of attribute is defined for the whole simulation model, and each piece has its own combination of attribute values, defining the item class to which it belongs.

When an item flows through a process, the process can update the attribute of the item, to reflect that the process was applied to this item. Similarly, when an item flows through a splitting process, the selected output can be specified according to the attributes of the item. At this point, we had to set a trade-off between expressiveness of the modelling language, its simplicity, and the efficiency of the simulation. Our trade-off is to consider that processes can update the class of items through three basic operations: setting an attribute, clearing an attribute, or loading a constant set of attribute. At the level of the simulation engine, any combination of these operations can be aggregated into two efficient bit-wise operations performed using bit masks representing attributes.

Another restriction that we have implemented is to consider how the class of items produced by a process are linked to the classes of potentially several inputs of this process. Our choice is to consider the union of all attributes of all inputs performed for starting a batch of the process, and set this union as the start class of the whole batch. The class transformation function of the process is then applied on this class, and every item output by the process from this batch share the same output class computed by this class transform function.

4 PERFORMING QUERIES OVER SIMULATIONS

The goal of our approach is to perform risk-related queries on factory simulations. These queries are meant to be performed on single runs of simulation occurring inside the Monte-Carlo engine which aggregates the queries results over the runs. It can then be queried afterwards e.g. for mean, median, extremes, variance of these queries over the runs.

Our query language can roughly be split into six sets of operators, namely: probes on processes, probes on storages, logic operators, temporal logic operators, arithmetic operators, and temporal arithmetic operators. Arithmetic and logic operators differ by their their return types; they return numeric and boolean values, respectively.

Since this query language runs over simulated time, we take the convention that the value of the queries are computed at the end of the trace on which they are evaluated. We define the operator of our language together with their semantics by using the $\models$ notation: $t \models P$ is the value of expression P when evaluated at position t of the current trace.

Some fragments of the queries are evaluated throughout the simulation. We take the convention that they are evaluated at the end of each discrete simulation step, after all events happening at this point in time are performed.

Some temporal operators refer to the previous position in time, denoted as $\text{prev}(t)$, notably to compute deltas or assess changes. These should be used with care since we are in an event-based model of time, so adding such operators in the query will add extra time events in the simulation.

4.1 Probes for Processes

The probes on processes are atomic operators that extract basic metrics from processes of the simulation model. Suppose that p is such a process, the following probes are supported:

- $t \models \text{running}(p)$ true if the process is running at time t, false otherwise.
- $t \models \text{completedBatchCount}(p)$ the total number of batches performed by the process between the beginning of the trace, and time t.
- $t \models \text{startedBatchCount}(p)$ the number of batches started by the process between the beginning of the trace, and time t. For a process with multiple lines, it sums up the started batches of each line.
- $t \models \text{totalWaitDuration}(p)$ the total duration where the process was not running between the start of

the trace, and time t. for a process with multiple lines, it sums up the waiting time of each line.

- $t \models \text{anyBatchStarted}(p)$ true if a batch as started by the process at time t

4.2 Probes for Storages

The probes on storages are atomic operators that extract basic metrics from storages of the simulation model. Suppose that s is such a storage:

- $t \models \text{empty}(s)$ true if the storage s is empty at time t, false otherwise.
- $t \models \text{content}(s)$ the number of items in the storage s at time t.
- $t \models \text{capacity}(s)$ the maximal capacity of s. This is invariant in time.
- $t \models \text{relativeCapacity}(s)$ the relative content of storage s at time t, that is: the content of the stock divided by the capacity of the storage.
- $t \models \text{totalPut}(s)$ the number of items that have been put into s between the beginning of the simulation and time t, not counting the initial ones.
- $t \models \text{totalFetch}(s)$ the number of items that have been fetched from s between the beginning of the simulation and time t.
- $t \models \text{totalLostByOverflow}(s)$ the number of items that have been lost by overflow from s between the beginning of the trace, and time t. If s is a blocking storage, this number will always be zero.

4.3 Operators

Logical Operators

- $t \models$ true the constant true.
- $t \models$ false the constant false.
- $t \models !l$ the negation operator.
- $t \models l_1$ op l_2 where *op* is one of $\{\&, \|\}$ represent conjunction, and disjunction operators, respectively, returning their conventional results.
- $t \models a_1$ comp a_2 where *comp* is one of $\{<, >, \leq, \geq, =, \neq\}$ represent comparison operators over numerical values, returning their standard results.

Temporal Logic Operators

- $t \models \text{hasAlwaysBeen } l$ true if for each t' in $[0;t]$, $t' \models l$
- $t \models \text{hasBeen } l$ true if there is a t' in $[0;t]$ such that $t' \models l$
- $t \models l_1$ since l_2 true if there is a position t' in $[0;t]$ such that $t' \models l_2$ and for each position t in $[t,t]$, $t \models l_1$
- $t \models @l$ true if both $t \models l$ and $\text{prev}(t) \models !l$.
- $t \models \text{changed}(e)$ e might be a logic or arithmetic expression; this evaluate to true when $t \models e$ and $\text{prev}(t) \models e$ have different values.

Arithmetic Operators

- $t \models n$ where n is a numerical literal represents a literal constant value
- $t \models a_1$ op a_2 where *op* is one of $\{+, -, *, /\}$ represent the classical arithmetic operators over numerical values, returning their conventional results.
- $t \models -a$ represents the unary negation.

Temporal Arithmetic Operators

- $t \models \text{delta}(a1)$ is a shorthand for $t \models a$ $\text{prev}(t) \models a$
- $t \models \text{cumulatedDuration}(b)$ let be $T = (t_1,t_2) \| t_1 = prev(t_2) \& t_1 \models b \& t_2 \models b$ the accumulated duration of b is the sum over the couples (t_1,t_2) in T of $t_2 t_1$
- $t \models$ time evaluates to t.
- $t \models \min(a)$ the minimum over all the values of $t' \models a$ with t' in in $[0;t]$
- $t \models \max(a)$ the maximum over all the values of $t' \models a$ with t' in in $[0;t]$
- $t \models \text{avg}(a)$ the average of all the values of $t' \models a$ with t' in in $[0;t]$
- $t \models \text{integral}(a)$ the integral of $t' \models a\, dt'$ with t' in $[0;t]$. The integral is computed through the trapezoidal rule taking the events as discretisation base.

5 EXPRESSING RISKS AS QUERIES

Using our query language, we can estimate quantities that can be related to risk on the "normal" operation of a factory. Before, we need to identify the risks we want to quantify. Some of those risks are dependent on one specific factory or stock, whereas some other risks are more general and influences a subset or the whole model.

Risks Specific to Stocks. In overflowing stocks, we are interested in measuring the risk of losing pieces, that are measured by the probe $totalLostByOverflow(stock)$. This allow us to

see if it is needed to adapt the capacity of the stock service. It is also possible that a stock is over-sized, that is, the maximum contents of the stock along the simulation are too low with respect to its capacity. This can be measured with the probe $max(relativeCapacity(stock))$ and verify whether that value is higher than an acceptable percentage of the stock capacity.

Risks Specific to Processes. In complex models, we are interested in processes that do not work enough in the simulation, or even processes that do not work at all. The percentage of idle time for a process p is measured by the probe $mult(div(totalWaitDuration(p), currentTime), 100)$. To detect whether the process p did not operate at all in the simulation, we can use the probe $hasAlwaysBeen(!anyBatchStarted(p))$.

Factory-specific Risks. We can check on failing processes the relative percentages of material that were successfully produced or had to be wasted. For example, if a failing process p with a probability of success of 80% is supposed to produce 5 units of raw material in stock a or waste 10 units to stock b at failure, checking that the materials put on each stock after the simulation, correspond to the expected percentage can be expressed with the complex probes $div(totalPut(a), 5 * completedBatchCount(p))$ and $div(totalPut(b), 10 * completedBatchCount(p))$.

6 BENCHMARKING

In order to assess the approach, we took a benchmarking approach based on a set of representative models. This section first describes our implementation, then the set of models before detailing the benchmarking results on our two main contributions:

1. **Expressiveness:** show that all the risks identified in the cases can easily be captured by the modeling primitives and measurement probe, either based on the set of generic probes identified so far or by writing case specific probes.
2. **Performance:** show that running probes does not degrade significantly the performance of the simulation engine.

6.1 Implementation

Our simulator is implemented using the OscaR DES module (OscaR, 2012) and is written in Scala. A modelling web front-end was developed with JavaScript technology, mainly Bootstrap, JQuery and JointJS. The lightweight Scalatra web framework was used to wrap up the simulator as a set of web services.

All the elements of the factory feature optimal $O(1)$ complexity for their update operations, with an additional cost factor for the attribute manipulations, although these are collapsed into a constant number of bitwise operations. Queries are evaluated incrementally during the simulation, by performing timely inspection of the internal state of the simulation model, so that the trace is actually not generated. Complex queries are split into sub-expressions that must be updated at each step of the simulation, such as integrals, and sub-expressions that only require a single evaluation at the end of the trace, such as a constant weighting factor on the integral. This keeps the overhead of our query language under control.

6.2 Benchmark Models

We selected four representative models out of a set of about 20 examples inspired by classic academic cases (with specific complex aspects) to anonymous cases collected in the industry. The cases also vary in the level of use of random variables. We describe relevant modelling aspects of each supply chain together with specific risk issues associated with each model.

First Case: A Simple Assembler Factory. This case, illustrated in Figure 5, models a simple factory that builds an industrial produce using two kinds of parts, A and B. Each part has its supplier which feeds the stocks when they become lower than a given threshold. For part A, the supplier policy is to refill the stock to its maximum capacity. For part B, the policy is the delivery of a fixed amount of material. Part B must be preprocessed before assembly. The factory combines two units of part A and with one unit of preprocessed part B with 80% of products passing the quality tests. so the assembly process can be represented by a failing single batch process. The goal is to assess if the input stocks are kept within safe limits to cope with production demand.

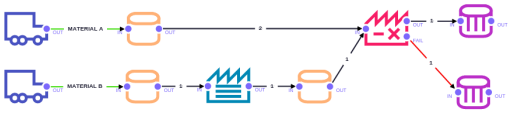

Figure 5: Model 1: a simple assembler factory.

Second Case: A Beer Game Model. Our second case model is a classical problem called ”beer game” (Klimov and Merkuyev, 2006). It is long linear supply chain going from the beer factory to the final retailer, passing by distributors and wholesalers. The

beer factory is considered here as a supplier and single batch processes are used to represent external sources of delay in the transport. Intermediary stocks also add extra delays. The continuity of retailing, distribution and wholesaler processes is modeled by conveyor belts. The resulting model is shown in figure 1. The goal is to assess where potential bottleneck can occur.

Third Case: Multiple Suppliers. This case is inspired by a real industrial case, where the manufacturing involves three different materials having their own supplier and refill policy, with random delays belonging to a gaussian probability distribution. 90% of produces built by a batch in the factory fulfill the quality requirements. We want to evaluate the effects of different supplying policies in order to ensure the supply chains operates at optimal capacity while minimising the frequency of orders.

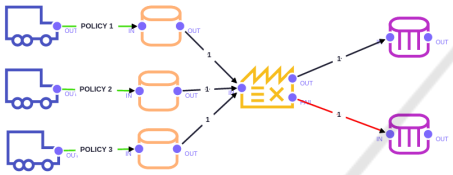

Figure 6: Model 3: multiple suppliers.

Fourth Case: A Complex Assembly Process. Our last case is inspired also in an industrial case of a factory where complex parts are assembled from 3 different materials following a complex process. Two of the parts are preprocessed on factory units that can fail (10% of failures for the first one, 40% for the second one). The process is shown in figure 7.

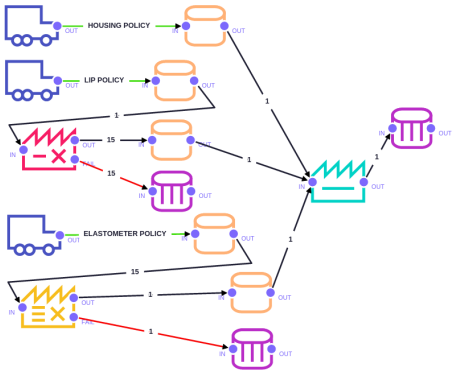

Figure 7: Model 4: complex assembly process.

6.3 Expressiveness Analysis

We identified a number of basic probes relating to risks directly related to model element. Such probes are automatically generated. SO, for each stock, we generate three different probes for measuring the average and maximum contents of the stock and verify whether the stock is full or overflowing. For each process, we generate a probe for measuring the amount of time in which the process was idle or blocked in the simulation.

Table 1: Benchmarking table for expressiveness.

Name	Size	Risk Type	#probes	Comments
M1	9 (2/2/5)	Full stock Process failure	20	Simple manufacturing
M2	17 (1/7/9)	Blocked process Process failure	36	Beer game
M3	8 (3/1/5)	Full stock, Process failure Supplier failure	20	Multiple suppliers
M4	14 (3/3/8)	Stock losses	31	Complex part assembly

In addition, the user can specify extra probes to express business specific risks that are typically more complex queries on the model. Table 1 summarises some model characteristics like size (suppliers/processes/storages), risks and number of probes. To assess expressiveness, we considered a single probe which actually proved enough to cover the targeted risks when used with basic probes. In the two last models, we could also explore risk mitigation strategies.

- In the first case, assessing the stocks of raw materials were full could be achieved the probe $cumulatedDuration(relativeContent(stockA) = 1)$. For most time in the simulation, the stocks were full while the assembler process worked was at full capacity.
- In the second case, we looked at the relative idle times in the process chain. We noticed that the distribution process, just after the fabrication, is the only one that blocks waiting for goods.
- In the third case, we both looked at process idle time (basic probe) and the average contents of stocks using the probe $avg(content(st))$. This helped us discovering the best threshold to trigger order while minimising idle time.
- In the fourth case, a full stock was blocking the production. We mitigate the problem by experimenting with overflowing storage to estimate the right size to avoid the overflowing, using the probe $totalLostByOverflow(lipStorage)$.

6.4 Performance Analysis

We performed a Monte Carlo simulation on each case, using a time limit of 10000 units and 2000 iterations

to have a good precision. The results in table 2 are the computed average. We performed the benchmarks on an Intel Core i7-4600U CPU at 2.10GHz with 8 GB of RAM. Only a single core is currently used. The simulation was triggered from the web interface on the same machine as the server.

Table 2: Benchmarking table for performance.

Name	No probes	Std probes	All probes	Overhead
M1	7,3 ms	11,9 ms	12,5 ms	71,2%
M2	11,3 ms	17,6 ms	17,8 ms	57,5%
M3	25,8 ms	29,2 ms	34,2 ms	32,6%
M4	6,8 ms	13,3 ms	13,4 ms	97,1%

The overhead in the simulation with probes varies from 32,6% in Model 3 to 97,1% in Model 4. Model 3 has the longest run time, because of the randomness in the delays of supplying induced by the probability distributions associated to suppliers. Model 4 has the shortest run time because the simulation stops at an early time, due to a full intermediary stock. In this case the relative overhead is bigger because the load is totally only probes evaluation while the model is stuck given the efficiency of the DES engine.

Globally, overhead are quite acceptable. Some improvements have still possible in the context of integration with a web application, especially to optimise the network requests between the web interface and the simulator and make that interface more responsive. The total simulation time allows to run thousands of simulations in a few minutes and to explore risk mitigation alternatives within an hour.

7 RELATED WORK

A typical risk assessment conducted on a given factory plan is reported in (Schmitt and Singh, 2009) base on the Arena simulation tool, featuring DES and Monte Carlo methods as in our work. It stresses the importance of conducting stress-tests using such simulation platforms. Its focus is mainly on the disruption risk while our work can cope with other class of risks like quality. Our framework provides an added abstraction layer that can cut down the cost of performing these important stress tests, and make them achievable by smaller industries.

A similar analysis has been performed on a beer supply chain in (Klimov and Merkuyev, 2006), whose model was presented in section 6.2. This analysis leads to an evaluation of inventory excessive accumulation, a back ordering. Again, no dedicated tooling was used for representing factories at a higher level, leading to high costs for conducting such evaluation in an industrial setting while our tooling could cope with using the available primitives.

Another simulation-based risk assessment is reported in (Finke et al., 2010). It features an aerospace company with very low production volumes, and leads to the elaboration of a dedicated simulation engine. The engine was first developed with a purely deterministic behavior, and then enriched with failure models and stochastic aspects. It showed of great value to the company even though mainly focusing on disruption risks, it helped the company to develop a risk mitigation procedure. Our framework has a similar purpose and try to propose a good compromise between genericity and efficiency.

(Almeder et al., 2009) presents a general framework that combines optimisation and DES for supporting operational decisions for supply chain networks. Their idea is to iterate between a simulation phase in which some parameters are estimated, and an optimisation phase, that adapts the decision rules for the simulation. Our current work does not cover the minimization of the risk. The tool is rather designed to ease the identification of risk controls by the risk manager. We plan to address optimisation in a later phase, based on the optimisation engines also present in the OscaR framework (OscaR, 2012).

8 CONCLUSIONS

This paper presented a Discrete Event Simulation Approach, supported by a toolset that helps to build a model of a supply chain with the goal to express and assess risks on them with a specific focus on procurement risks. The assessment is conducted using an Monte-Carlo based simulation engine that can also be used to further explore risk mitigation strategies.

Our strength is to support a declarative and easy to use graphical modeling language for representing factory processes and stocks, together with a declarative query language for defining metrics to be measured while simulating the behaviour of the modeled system. We could successfully benchmark our approach both from the expressiveness and performance perspectives on several typical examples of factories, together with their supply chain policies.

Of course, further work is required to fully align it with industrial needs. Our current step is the internal validation with a number of companies by putting the tool in the hands of the risk managers on a pilot case. We already identified a number of requests about:

- extension to the modeling language, e.g. to support the notion of shared resource among processes and have a statistic model of process fail-

ures and breakdowns. More specialised processes allowing controlled fork/join are also required.

- identification of model parameters easing the manual (and later optimised) exploration of risk mitigation strategies.
- availability of a company library of specific risks and related probes
- produce specific reporting (e.g. business continuity plans). We have already explored some work in this direction (Arenas et al., 2015).
- possibly support model refinements and granularity of simulation. However our aim is not to capture the full reality but what will help assessing identified risks.
- parallelisation in case of need of faster simulation times. This is easy to implement.

Our framework combining usability, expressiveness and efficiency is an important milestone in our work to raise the awareness of companies, especially of smaller size, w.r.t. the need to evaluate their procurement risks and elaborate their supplying policies in the most optimal way. We believe it can be used to manage more general risks. Our design ideas can also be used to improve other risk management tools. Our framework is available online (SimQRi, 2015) and is planned for Open Source release.

ACKNOWLEDGEMENTS

This research was conducted under the SimQRi research project (ERA-NET CORNET, Grant No. 1318172). The CORNET promotion plan of the Research Community for Management Cybernetics e.V. (IfU) has been funded by the German Federation of Industrial Research Associations (AiF), based on an enactment of the German Bundestag.

REFERENCES

Almeder, C., Preusser, M., and Hartl, R. F. (2009). Simulation and optimization of supply chains: alternative or complementary approaches? In Günther, H. O. and Meyr, H., editors, *Supply Chain Planning*. Springer-Verlag.

AnyLogic (2015). AnyLogic Multimethod Simulation Software. http://www.anylogic.com.

Arenas, A. E., Massonet, P., and Ponsard, C. (2015). Goal-oriented requirement engineering support for business continuity planning. In *Proceedings of MReBA'15*, Stockholm, Sweden.

Artikis, C. and Artikis, P. (2015). *Probability Distributions in Risk Management Operations*. Springer, London.

Brailsford, S., Churilov, L., and Dangerfield, B. (2014). *Discrete-Event Simulation and Systems Dynamics for Management Decision Making*. Wiley.

Byong-Kyu, C. and Donghun, K. (2013). *Modeling and Simulation of Discrete-Event Systems*. Wiley.

Deleris, L. and Erhun, F. (2005). Risk management in supply networks using Monte-Carlo simulation. In *2005 Winter Simulation Conference*, Orlando, USA.

Finke, G. R., Schmitt, A., and Singh, M. (2010). Modeling and simulating supply chain schedule risk. In *2010 Winter Simulation Conference*, Baltimore, USA.

Gleißner, W. (2012). Quantitative methods for risk management in the real estate development industry. In *Journal of Property Investment & Finance*, volume 30(6), pages 612–630.

Klimov, R. A. and Merkuyev, Y. A. (2006). Simulation-based risk measurement in supply chains. In *20th European Conference on Modelling and Simulation (ECMS 2006)*, Bonn, Germany.

OscaR (2012). OscaR: Scala in OR. https://bitbucket.org/oscarlib/oscar.

Printz, S., von Cube, J. P., Vossen, R., Schmitt, R., and Jeschke, S. (2015a). Ein kybernetisches modell beschaffungsinduzierter störgößen. In *Exploring Cybernetics - Kybernetik im interdisziplinren Diskurs*. Springer Spektrum.

Printz, S., von Cube, P., and Ponsard, C. (2015b). Management of procurement risks on manufacturing processes - survey results. http://simqri.com/uploads/media/Survey_Results.pdf.

Rockwell Automation (2015). Arena Simulation Software. https://www.arenasimulation.com.

Romeike, F. (2004). Der prozess der risikosteuerung und -kontrolle. In Romeike, F., editor, *Erfolgsfaktor Risiko-Management*, pages 236–243. Gabler, Wiesbaden.

Schmitt, A. and Singh, M. (2009). Quantifying supply chain disruption risk using Monte Carlo and discrete-event simulation. In *2009 Winter Simulation Conference*, Austin, USA.

Siemens (2015). Plant Simulator. http://goo.gl/NK7yWg.

Siepermann, M. (2008). *Risikokostenrechnung: Erfolgreiche Informationsversorgung und Risikoprävention*. Erich Schmidt, Berlin.

SimQRi (2015). Online SimQRi tool. https://simqri.cetic.be.

Sutton, I. (2015). *Process Risk and Reliability Management*. Elsevier, second edition.

von Cube, J. P., Abbas, B., Schmitt, R., and Jeschke, S. (2014). A monetary approach of risk management in procurement. In *7th Int. Conf. on Production Research Americas' 2014*, pages 35–40, Lima, Peru.

Zio, E. (2013). *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*. Springer, London.

Zsidisin, G. A. and Ritchie, B. (2009). *Supply Chain Risk: A Handbook of Assessment, Management, and Performance*. Springer.