SimQRi - A Query-oriented Tool for the Efficient Simulation and Analysis of Process Models

(Tool Demonstration)

Christophe Ponsard*, Quentin Boucher *, Gustavo Ospina*
*CETIC Research Centre, Belgium
Email: {christophe.ponsard, quentin.boucher, gustavo.ospina}@cetic.be

***Abstract*—Process models are an abstraction used in several domains such as manufacturing (transformation chains), logistics (procurement and distribution networks), architecture of electronics systems (network of data/computation nodes). Such systems are often subject to requirements related to the processing delay, throughput, overall reliability, or quality attributes of specific outputs. Those characteristics are highly dynamic. Assessing them at design time requires some kind of execution of the model, typically using simulation. As the system is often non-deterministic, several simulations need to be run and combined in order to draw relevant conclusions. In this paper, we describe a tool, called SimQRi, that we developed to efficiently run a large number of simulations over process models, using Discrete Event Simulation combined with Monte-Carlo techniques. Their key point is that the properties to be assessed are formulated as queries over the model with a trace semantics. Queries are evaluated and aggregated through simulations, so there is no need to store traces and perform post-processing on them. Several operators are available on different process-related components (storage content, process activity, number of processes items, etc). In our demo we will demonstrate how the tool can be used 1) to assess several risks on supply chains and 2) to design a green Cloud to cope with response times with optimal energy usage.**

***Index Terms*—Process Models; Risk Assessment; Discrete-Event Simulation; Oscar.cbls;**

I. INTRODUCTION

Process models are very common abstractions in many application domains both in the physical world (logistics, supply chain domains), in computer world (Cloud architecture, signal processing, etc), or even in hybrid domains such as smart manufacturing heavily mixing physical processes with IT data collection and analysis processes based on the Internet of Things and Big Data.

Reasoning on such processes is not always easy because of the dynamic nature of the requirements to enforce. It can require some form of prototyping already beyond the design phase. At design time, an option is to use model-checking, e.g. using the Communicating Sequential Process (CSP) abstraction and tools like FDR [1], [2]. However, such approaches have some limitations in expressiveness and size of manageable models.

We consider a more practical approach based on model simulation. Our primary motivation is to help small and medium enterprises (SMEs) in improving their maturity level to master the processes present in their domain, focusing on supply chains as primary domain, as confirmed by a survey we conducted [3]. Based on this, we developed a framework and its supporting tooling composed of:

- a modelling language to represent process models using quite abstract building bricks (processes, flows, storages) so it can be used in different application domains.
- an expressive query language to capture a large variety of quantifiable properties.
- an editor tool supporting graphical modelling and tabular capture of queries, including the specification of stochastic parameters for most of the parameters and model validation, as well as feedback at the user interface level.
- a simulation engine using Discrete Event Simulation (DES) and Monte Carlo Simulation (MCS) in order to cope with the non-determinism present inside the model.
- a reporting tool to analyse the simulation results.

The key point of such an approach is to reach a high level of efficiency. In order to reach that goal, the tool is designed to compute all the queries during the simulation without the need to store any trace and post-process on them. The ultimate goal of our research project is to help companies to fulfil strategic goals, minimize financial, reputational and productivity losses and improve their overall productivity and reliability.

Our tool is available at https://simqri.cetic.be both as an online web-based application (requiring no installation) and as an Eclipse plugin.

This paper is structured as follows. Section 2 gives a summary of the framework in terms of meta-model, query language and architecture. Section 3 details a use case related to supply chain risk management while Section 4 details another case related to green cloud design. Finally Section 5 draws some conclusions and our development roadmap.

II. FRAMEWORK DESCRIPTION

Our framework is composed of the following elements:

- a Domain Specific Language (DSL) based on a meta-model able to capture all the main elements such as suppliers, storages, processes, and flows.
- an expressive Query Language able the capture a rich set of properties. Those properties can be efficiently measured and statistically processed during the simulation.
- an architecture based on an Open Source simulation engine and both a web-based and desktop-based user interface.

A. Meta-model Overview

Our DSL is industry 4.0 oriented and does not pursue the same goals as business process modelling languages. Its meta-model is described in Figure 1 and is composed of activable components such as storages and processes, through which items can flow.

- *Storages* represent any kind of device or place for storing raw materials (e.g. a warehouse in a supply chain or disk space for an IT infrastructure). They have a maximum capacity. When this capacity is reached, they can either block any process trying to put more items into them or overflow (and lose the overflown items).
- *Batch processes* are processes that work in a batch fashion. The supply items of those processes are collected from several input storages, then they perform their work for some time, and finally the produced outputs are dispatched to their respective output storages before this whole cycle starts again. They can have a single or multiple production lines, which can model, for instance, multi-line physical processes or multi-threaded computer processes.
- *Continuous processes* are processes where items are continuously picked from input storages and outputs are produced like in a conveyor belt. Continuous processes act like pipelines and process several items in a queue.

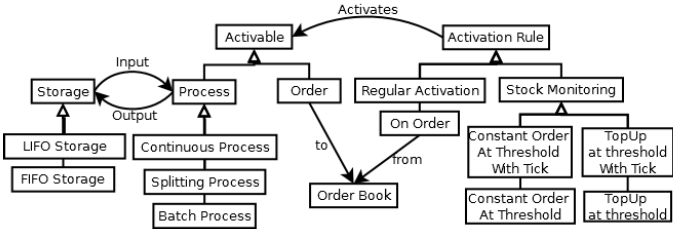


Fig. 1. Meta-model of our tool [4]

The flow of items is controlled through activation rules related to specific policies:

- *On Order Activation* represents on-demand production triggered by a specific order book.
- *Stock Monitoring* are used to implement different procurement policies (or possibly internal stock transfers) based on specific conditions on the demanding storages.

B. Query Language Overview

Here, we give an overview of the main operators that can be used to express queries on the model enabling a large range of model analysis detailed in our usage scenarios. A more detailed description is available from [4].

Queries for Processes: These are atomic operators that extract basic metrics from a process p of the simulation model. Simple operations are available to check for the number of started operations ($\text{startedBatchCount}(p)$) and completed operations ($\text{completedBatchCount}(p)$), given a process can fail with some probability. Other operations can be used to check about the timing and load aspects ($\text{totalWaitDuration}(p)$ and $\text{meanLoad}(p)$). Specific events can also be detected like $\text{anyBatchStarted}(p)$.

Queries for Storages: These are atomic operators which extract basic metrics from a storage of the simulation model. The status of a storage s at evaluation time can be retrieved using classical functions like $\text{capacity}(s)$, $\text{empty}(s)$ or $\text{content}(s)$, respectively returning the (fixed) capacity, the emptiness status and the number of items. Other operations are available for getting usage information, like $\text{totalPut}(s)$, $\text{totalFetch}(s)$ and $\text{totalLostByOverflow}(s)$ respectively returning the number of items put into s, fetched from s or lost by overflow of s.

Operators: Complex queries can be built using following operators over other queries (basic or complex), some of them also referring to one or more states of the considered trace:

- logical: $\text{true}, \text{false}, not(!), and(\&), or(\|), <, >, \ldots$
- temporal logic: $\text{hasAlwaysBeen}, \text{hasBeen}, \text{since}, \ldots$
- arithmetic: $+, -, *, /, sum, \ldots$
- temporal arithmetic: $\text{time}, \text{min}, \text{max}, \text{avg}, \text{integral}\ldots$

C. Tool Architecture

The global architecture of our tool is depicted in Figure 2. The lower layer provides the Discrete Event and Monte-Carlo Simulation infrastructure, based on the OscaR Scala Open Source library [5]. The middle layer is providing a REST API that is used only for the web-based version of the tool. The top layer is the user interface layer provided both as an online web application and an Eclipse interface based on EMF and Sirius technologies [6], [7]. The former requires no installation and support model sharing while the later is more secure and easy to integrate and adapt.

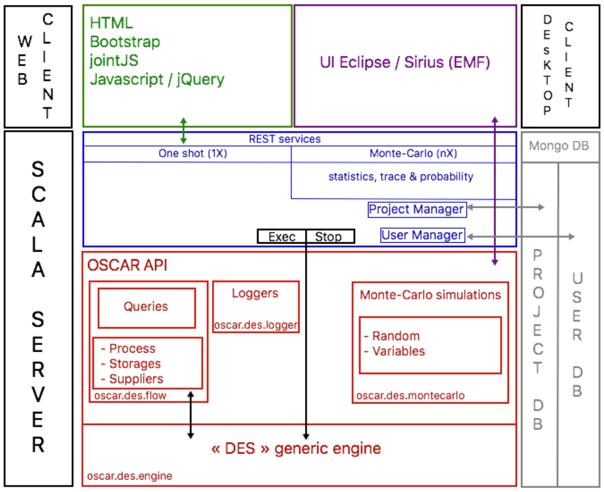


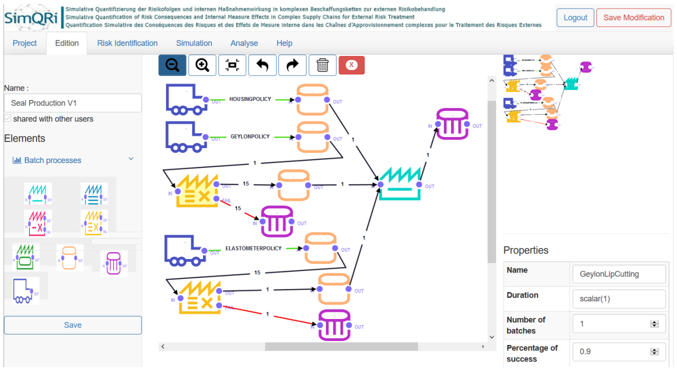
Fig. 2. Tool Architecture

A major point is the simulation efficiency: all elements feature optimal $O(1)$ complexity for their update 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. This ensures the tool is running minimal space and time overhead [4].

III. SCENARIO 1: SUPPLY CHAIN RISK MANAGEMENT

Supply chain risk management is the implementation of strategies to manage both everyday and exceptional risks along the supply chain, based on continuous risk assessment with the objective of reducing vulnerability and ensuring the process continuity [8]. Such risks can occur for several reasons, both external or internal.

In our demo, we investigate a relatively simple case consisting of three components, which are procured externally. The production process can be characterised as an order-driven, small batch, job shop production as depicted in Figure 3.


SimQRi
Simulative Quantifizierung der Risikofolgen und internen Maßnahmenwirkung in komplexen Beschaffungsketten zur externen Risikobehandlung
Simulative Quantification of Risk Consequences and Internal Measure Effects in Complex Supply Chains for External Risk Treatment
Quantification Simulative des Conséquences des Risques et des Effets de Mesure interne dans les Chaînes d'Approvisionnement complexes pour le Traitement des Risques Externes

Name : Seal Production V1
☑ shared with other users

Properties

Name	GeylonLipCutting
Duration	scalar(1)
Number of batches	1
Percentage of success	0.9

Fig. 3. Risk-oriented modelling

For this domain, a specific wizard was developed to capture different kinds of risks (quantity, quality, delay) using a structured risk model that can be automatically translated into specific queries in the query language. E.g. at system level, the quantity risk can be expressed as: $(content(OUTPUT) - content(ORDER)) * ORDER.partCost$. The same model is used to produce the risk analysis dashboard to identify risk causes and then try to address them.

Controlling the risks requires tuning the model in order to minimise the estimated costs induced by risks. However, changing a parameter in the model might introduce multiple (conflicting) effects. The tool supports a simple way to change the values of some parameters to find out their optimal values, given that other parameters remain constants. In our case, this can be used to find optimal ordering frequency.

IV. SCENARIO 2: GREEN CLOUD DESIGN

Cloud application deployment is becoming increasingly popular for the economy of hardware costs, the pay-per-use model and the ability to scale. However, deploying software on the Cloud carries both opportunities and threats regarding energy efficiency. Our tool can help Cloud application developers to learn and reason about the energy consumption of their application on the server-side at design time.

Our case study is a 3-tier web application that is designed to be an online photo manager [9]. It provides social services for uploading, storing and previewing photos, creating and sharing albums. The starting point is a model of the Cloud Application and energy requirements expressed through a specific UML profile [10]. A simplified model is shown in Figure 4. The total energy consumed by the running processes can be expressed as follows: `sum(Process.allInstances() -> select(p:Process|p.energy*p.processedRequests)`

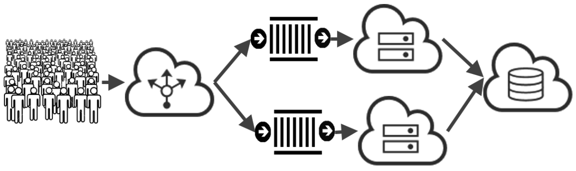


Fig. 4. Photo album model in the web interface

In this case, we can also consider self-adaptation policies of the form *WHEN condition THEN action*. The *condition* part is a query on the model while the *action* part is some model adaptation affecting some parameters like *Process.add/removeInstance*. The following policy can be used to express how to scale down for energy efficiency: `WHEN HasAlwaysBeen(MEDIA.running<0.8*MEDIA.total,10min) THEN MEDIA.removeInstance(1)`

Based on this model, probes and evolution policies, simulations and analysis can be carried out.

V. CONCLUSION AND ROADMAP

The lessons learned so far by applying our tool on different use cases shows it can adapt to different domains related to process modelling and analysis. The query mechanism proved both expressive and efficient on the feedback collected so far.

Our on-going work is mainly to improve the more recent Eclipse-based interface and to extend the reporting capabilities of the tool by adding statistical graphics for the simulation results. Traceability links between simulation results and requirements should also be explored.

ACKNOWLEDGMENT

This work was partly funded by the ASCETiC FP7 project (grant nr. 610874).

REFERENCES

[1] A. W. Roscoe, “Model-checking CSP,” in a Classical Mind, Prentice Hall International (UK) Ltd., 1994.
[2] J. Sun, Y. Liu, and J. S. Dong, “Model checking csp revisited: Introducing a process analysis toolkit,” in *In ISoLA 2008*. Springer, 2008.
[3] S. Printz, P. von Cube, and C. Ponsard, “Management of procurement risks on manufacturing processes - survey results,” http://simqri.com/uploads/media/Survey_Results.pdf, 2015.
[4] R. De Landtsheer, G. Ospina, P. Massonet, C. Ponsard, S. Printz, L. Härtel, and J. P. von Cube, “A Discrete Event Simulation Approach for Quantifying Risks in Manufacturing Processes,” in *Int. Conf. on Op. Research and Enterprise Systems (ICORES)*, February 2016.
[5] OscaR, “OscaR: Scala in OR,” https://bitbucket.org/oscarlib/oscar, 2012.
[6] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks, *EMF: Eclipse Modeling Framework 2.0*, 2nd ed. Addison-Wesley Professional, 2009.
[7] Obeo, “Sirius Designer,” http://www.obeodesigner.com/sirius, 2016.
[8] A. Wieland and C. M. Wallenburg, “Dealing with supply chain risks: Linking risk management practices and strategies to performance,” *Int. Journal of Phys. Distr. & Logistics Mngmt*, vol. 42, no. 10, 2012.
[9] A. Tsebro *et al.*, “Rich faces photo album application,” http://docs.jboss.org/richfaces/latest_3_3_X/en/realworld/html_single, 2009.
[10] C. Ponsard, J. Deprez, and R. Michel, “Guiding cloud developers to build energy aware applications,” in *Software Technologies - 10th International Joint Conference, ICSOFT 2015, Colmar, France, July 20-22, 2015, Revised Selected Papers*, 2015, pp. 397–414.