Implementation of the RAM Analyses into a Discrete Event Simulation of a Process in Early Stages of its Development

Martin Kubic, Maurice Pendola

Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

Implementation of the RAM Analyses into a Discrete Event Simulation of a Process in Early Stages of its Development

KUBIC Martin
Axone
Château de la Saurine, 1985 route de Martina, 13590 Meyreuil, France
m.kubic@axonegroup.com

PENDOLA Maurice
Axone
Château de la Saurine, 1985 route de Martina, 13590 Meyreuil, France
m.pendola@axonegroup.com

***Abstract* —** Modeling complex systems has become a common tool in many fields, especially in engineering, mathematics, military, and transport sciences. It provides a relatively inexpensive way to gather information for decision making. Since the size and complexity of real systems in these areas rarely allow analytical solutions to provide the information, simulation has become the method of choice. For industrial processes, discrete event simulation is a method allowing to model systems decomposed into individual processes advancing in time based on events of other processes. This paper describes an implementation of the aspects of reliability availability and maintainability (RAM) into a discrete event simulation (DES) tool in order to perform architecture choices and allocation based on both physical behavior and RAM considerations in the design phase of an industrial process. The model is implemented in MATLAB, and more precisely within the SimEvents module. A special block element was developed in the SimEvents environment to consider random failures (and associated repair actions) of the components participating in the flow process of a system. The impacts of these random failures are then confronted with the availability and capability requirements of a system to perform at required level while taking into account the management of the various resources (operators, maintenance people, etc.).

Keywords — MBSA, RAM analyses, discrete event simulation, SimEvents

I. INTRODUCTION

Recent studies have shown that engineers continue to spend an enormous amount of time researching information and assembling reports. This trend has only grown with the increase of scale and complexity of systems, resulting in a dramatic increase in system requirements. Thus, managing requirements using simplistic methods is no longer enough. With increasing system complexity, document-centric approaches have become increasingly difficult to manage due to the increased risk of overlooking critical information and key interfaces. This has given rise of the Model Based System Engineering (MBSE) in order to replace the document centered management by a management centered around the models all along the life cycle of a system starting early in the design phase up to the verification and validation phases [1].

Although the modelling of complex systems has recently become a common practice in many industrial projects, the simulation methods are used primarily in the later phases of the projects as a verification tool (Fig. 1) [11][12]. The absence of the application of the simulations methods from the very beginning of a system life leads often to unpredicted additional costs and delays of the projects [13]. Although there exist attempts to implement the simulation in the decision-making stage, several challenges are faced including constant evolution of the system's design or input data uncertainties. Very often, especially in the French nuclear environment, these obstacles overweight the potential outcome the simulation at the large scale of a system can provide.

From another point of view, safety assessment has also to be taken into account in early stages of the design. For that purpose, the model-based approach has been adopted into the reliability and safety analysis, currently known as Model-Based Safety Assessment (MBSA) and it soon started to gain the trust across the industrial and academical spectrum [2]. The classical Boolean analysis like fault trees, event trees or reliability block diagrams [3],[4] are well mastered nowadays but are rather focused on the low-level system modelling which represents limitations especially in the design stages of a system where high-level approach is necessary [5]. In addition, these analyses models have their limitations when the system becomes dynamic, not to say they are hard to maintain as the system design tends to evolve constantly. On the other hand, analyses based on Markov chains [8] or Petri nets [9] allow to model the dependencies between system's elements and to analyze sequential behavior of actions over the time (especially failure propagations) via various events occurring in the system and serve as a base for many MBSA tools.

Nowadays, in industrial projects, these two approaches MBSE and MBSA, when applied, are almost always used as two standalone independent approaches. The idea presented here is to merge some parts of the safety analyses with the MBSE modelling in early stages of the design in order to enriched them progressively till the Verification and Validation stages.

The aim of this paper is to provide an example of the implementation of a simulation method at the very beginning of the design-phase of an industrial process to support the decision-making at the highest architect level. The paper is

structured as follows. Firstly, an introduction to modeling

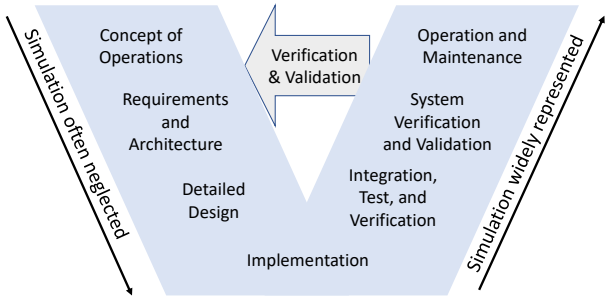

Fig. 1. Example of a V cycle. Very often there is a lack of simulation methods in the decision-making phase.

from a system engineering point of view, particularly in the design phase, supplemented by the principles of discrete event modeling is provided with the emphasis on the SimEvents tool. The body of the paper is then devoted to the description of the implementation of the RAM elements into a case study model applied to a high-performance process installation. Finally, the study is put in context with the implementation of the RAM analysis in early stages of a project as a decision-making support at the overall architect level to aim to improve the system design, to perform RAM requirements allocation based on the sensitivity of the subsystems, in order to put the efforts on important subsystems and then minimizing the design and operational costs over the system's life cycle.

II. Modelling of a Complex System

As mentioned above, there are simulation methods, tools implementing these methods, but they are generally not applied correctly. Many system engineers start modeling with the goal of modeling, overdoing it, going too deep (with the focus more on using simulation to verify or validate performance) or simply to model the bad things. Model-based systems engineering, and model-based safety assessment are essential in the efforts to design increasingly complicated and complex systems in an era of unprecedented change. However, it is a tool and a technique in many systems engineering toolbox. Nevertheless, the focus should be on applying systems engineering to deliver the required value to the customer and stakeholders effectively and efficiently.

Detailed models are not always desirable, especially in the early phases of the life cycle (the pre-study phase and the descending phase of the V-cycle). At this stage, the systems are not yet fully defined. The contribution of the models should provide a broader insight into the behavior of the system, giving directions and insights. The interest is not to chase the precise values as the result of the model (which is already quite difficult given the complexity of the systems). In many cases, orders of magnitude are enough to understand it. At the same time, the models must be able to provide the flexibility to model various scenarios and possibilities of the system configuration.

The objective of this paper is to demonstrate whether it is possible to build a recursive development of modeling as it is done elsewhere in system engineering on design in general by building a Simulation Breakdown Structure (SBS) and a corresponding Simulation Architecture in such way

- From the most general (system performance) to the specific (component performance) while changing the paradigm on precision. To favor the approximately right at the general level in the upstream phase to the detriment of the precisely wrong.
- Use this approximative modeling to develop and enrich the system engineering requirements model.

III. Discrete Event Simulation

Discrete-event simulation is a method that allows to simulate the behavior and to quantify the performance of a process consisting of a series of ordered sequences. Unlike continuous systems whose state variable(s) change continuously over time, the state variable(s) of a discrete system varies only at a discrete set of points in time. A discrete system can be imagined as a set of entities that are connected and communicate when an event or activity occurs. Each entity is characterized by a set of properties (attributes) that describe their current state and affect their behavior. Entities representing components of the system under investigation must be explicitly modeled in order to capture the behavior of the system in relation to the simulation study. As the simulation time evolves, entities can change their state as a consequence of activities that happened during a given simulation period. The time in the simulation at which such a state change occurs is called an event. The relationship between events and activities are defined by user and are based on the objectives of the studied system [14]-[17]. This includes in particular the specification of activity durations, which can be modeled as deterministic and based on stochastically influenced parameters (simulation of failures of the system's elements).

Discrete event simulation methods can be used in different fields of application. In this paper, the focus area is related to a logistics process. The history of discrete event simulations goes back to the early 1960's, but it was only towards the end of the last century that the DES application spread widely as information technologies boomed [20]. Currently, there exist many discrete event simulation tools [18][19].

IV. Simulation Tool

Over the years, several MBSA methods have been developed and adopted by the industry which yielded into a development of specialized MBSA languages like Alta Rica [5] and dedicated software like GRIF© [6] or Simfia© [7]. These tools are well adapted to evaluate the RAMS indicator, generate FMEA (Failure Mode and Effects Analysis) to some extent or construct a fault tree. Although they are adapted to model discrete event systems, these tools are not directly and efficiently adapted to study the propagation of flows, which is the main point of interest in the case study presented here.

For the case study presented in this paper, it was decided to use the SimEvents software of the MathWorks company, because the SimEvents provides a wide range of predefined tools for the DES and in combination with Matlab Simulink allows the user to develop and customize the model according to his needs. Although Matlab does not dispose of a specialized RAMS module, it allows within the Simulink environment to develop a customized object. Some hybrid techniques have been proposed to integrate the HiP-HOPS

with Matlab Simulink [10] or to implement the failure logic modelling methodology in the context of Matlab Simulink and Stateflow [2].

In the frame of the presented case study, the aim was not to provide a detailed safety analyses but rather to simulate whether a processing line under consideration would be capable to satisfy the required performance objectives. Standard MBSA tools might be able to cover a part of the processing line if kept simple, but they fail when the entities need to be introduced into the system in a particular manner, combined with other entities to create new ones, to be multiplied or to be decomposed into several other entities. The attributes of the entities change constantly affecting the flow priorities which in consequence drive the flow. In addition, the interest was also the logistic support systems, especially to the management of the resources, the spare parts and the waste.

From a practical point of view, while the SimEvents library contains all the crucial types of blocks needed for basic simulations, it is obvious that the development of add-on blocks might be unavoidable if the system becomes more complicated, for example, by adding the possibility of machinery breakdowns in a manufacturing/processing line.

However, in addition to the SimEvents library, Matlab contains a few exemplary models that users can test. The Matlab 2019a release provides an example of a Machine block that attempts to incorporate principles of failures, reliability, and maintainability into SimEvents simulations.

The main downside to the native form of the Machine block is that it simply does not model properly the case of a failure. Firstly, the random failures are characterized by an exponential law rather than a gaussian law. Another problem was related to the management of the resources needed to maintain a failed block that were not used at all causing serious consequences for the optimization studies of the installation resources. The last limitation of the default version of the machine block was related to the preventive maintenance. The theory of dependability studies considers the components undertaken a maintenance task as new which translates in simulation by resetting the internal component clock to zero. The default block configuration omitted this fact yielding the maintenance as it has never happened. One by one, all these issues have been treated by authors and the associated problems eliminated as well as some additional features were added allowing to switch properly between the block's internal states.

V. Illustration Case Study

This section is devoted to a demonstration of how a real system (which is in the design phase) can be modeled using SimEvents. The case study concerns the modeling of the flow of storage / removal of items of a future installation in order to be stored. For the policy reason, some details of the system cannot be shared. However, the aim is to focus on the benefits of the simulation method as a decision-making support at the overall architect level in order to improve the design and minimize the operational costs over the system's life cycle.

The items storage process can be viewed as a manufacturing process that transforms an object into another object. To take this into account, and due to certain Matlab limitations (existing pre-programmed objects), a single type of entity is used to represent the different objects appearing in the process (convoys with full or empty items, wagons, the

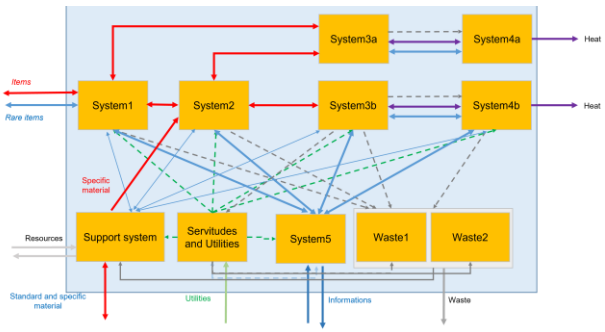

Fig. 2. Illustration of the main elements of the storage facility.

secondary elements to be stored, or waste products. The entities themselves have no graphical representation. Their flow is represented by arrows connecting the blocks. On the other hand, entities can transport data in the form of attributes. The attributes are used as a decisional key to take actions with the entities like orienting their flow through various gates for example.

The installation is composed of several systems (that are denoted for simplicity as System1, System2 etc.) and some auxiliary systems to support the main systems (Fig. 2). As the items advance through the storage line, each item undergoes a process of manipulation and transformation until it is temporarily stored in the System3 (red lines).

Each system is composed of several subsystems (workstations) each having associated a group of available resources. At the design phase, the composition of the subsystems is not yet defined, and the performance allocations are made at the subsystems and system levels (level 0 and level 1 of a traditional Product Breakdown Structure - PBS). The items to be stored arrive at System1 by a convoy at a predefined rate and are processed through the installation line. Inversely, the stored items can also be evacuated as the new ones arrive. The requirements impose a certain number of items to be stored over a given period. As can be seen, no detailed input data are provided and in fact they are not needed at this stage to model the system and its capabilities as is described next.

A. *Model Description*

In the model, each of the subsystems is represented by the improved block Machine which can transition between four predefined states that are defined in the operational phase of the considered system lifecycle

- Operation state
- Maintenance
- Breakdown
- Idle

In the regular operation state, the machine acquires a working resource and processes a given task. In the scheduled maintenance state, the machine switches to service mode and the processing of a task is interrupted. After a fixed service time, the machine resumes regular operation. The machine can also break down sporadically following an exponential probability law and enter a breakdown state. The time to repair is defined by a mean time to repair and the machine resumes

regular operation once the repair is completed. If there is no task to be done, the machine stays in an idle state.

The main structure of the flow model in the SimEvents interface is shown in Fig. 3. As can be seen, the model contains several distinct blocks. The blocks are built according to the principle of Russian dolls which has the advantage of being able to look at and analyze the model at different scales. It also allows building a progressive validation.

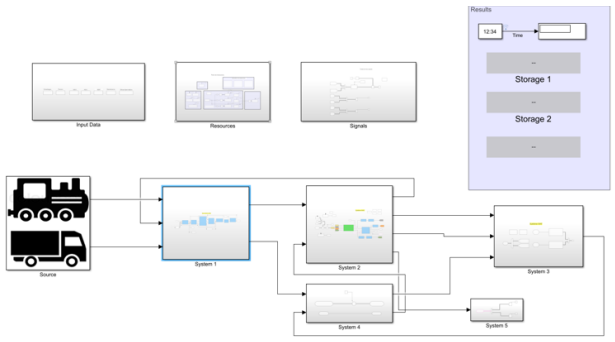

Fig. 3. Main structure of the model in the SimEvents interface.

As the entity representing the treated object advances in the process, different events occur and trigger another subsystems and elements like resources. The propagation of the entities through the system is governed by a set of rules controlling the flow e.g., an entity cannot move forward if the next workstation is occupied etc.

B. Model Validation

The good performance of the model was systematically and incrementally validated. In addition, non-regression checks were carried out each time the model was upgraded (implementation of a new block, a control function, etc.): an immediate check in the simulation data inspector was performed in order to check if the behavior of the model was correct.

To demonstrate that the flow inside the installation is correct, the case of a convoy transporting three objects arriving at the installation was considered. The objects are unloaded and introduced into the System1 one by one according to the availability of the workstations. The Gantt chart representing the progress of the transformation of the arriving items into their stored location is shown on the Fig. 4.

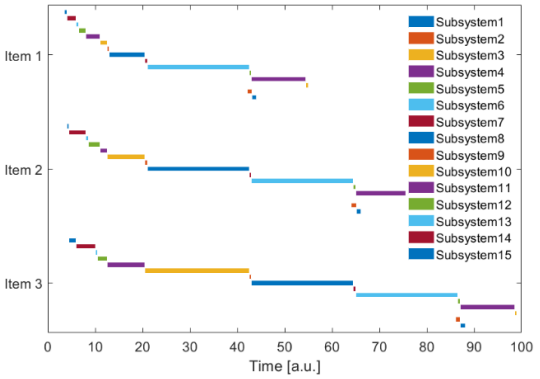

Fig. 4. Gantt diagram of the flow of three items.

Fig. 4 shows that the flow of the objects inside the installation is continuous and in order. This shows that the flow is controlled meaning that the entity (object) cannot advance if the next workstation is occupied by treating another object.

In principle such a simple case can be modelled manually if some average values were considered reinforcing the model validation. However, if random failures are considered, the situation changes dramatically, and manual prediction is not applicable. Fig. 5 gives an example of the delay necessary to treat several convoys. As can be seen, the delays are now becoming irregular as some workstations break down and processing is delayed due to the time required for repair. The modularity of the model allows to modify it by adding additional workstations, modifying the delays and generally to optimize the installation performance capabilities.

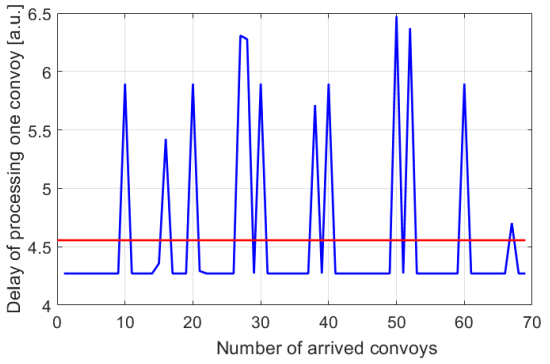

Fig. 5. Example of the delays necessary to treat the content of one convoy. Red line represents the mean delay.

In addition to the possibility to model the random breakdown of the various subsystems, the model allows to manage and to size the need of the resources. Several types of resources are defined for each of the systems: operator, technician, supervisor and so on. Some of the workstations require a different number of resources for their operation. Similarly, each time a workstation breaks down, a maintenance resource represented by a technician (of limited pool) is called to perform the maintenance task so if multiple failures occur at the same time, some workstation must wait

until a technician is liberated from another task. With regard to the resources involved in the operation of the installation, their rate of use is shown on Fig. 6. Information on the rate of use of the resources allows to optimize the pace of work. For example, it does not appear necessary for the teams to always be present at the installation. In addition, their presence must be carefully planned, because some subsystems may break down, which would delay the flow. The model as it stands assumes that resources are available when needed but can be easily adopted to consider the work on shifts or mutualization of the resources during less occupied periods which are defined by the convoy's programmable arrival schedule.

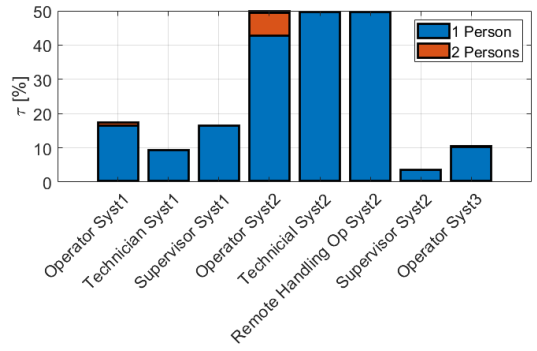

Fig. 6. Resources use rate for various subsystems.

VI. Summary

The model presented here was developed to simulate the storage process of a future installation. Model validity was performed systematically and incrementally. Non-regression checks were carried out at each evolution of the model in order to check whether the behavior of the model was correct. The model is based on decision parameters and model version for a given configuration state of the design with its associated definition artefacts (functional architecture, physical architecture, etc.)

The line process is not difficult to imagine for simple cases, but the model becomes very useful when the introduction of workstations availability is applied. The model clearly shows that it can be considered as a reliable source for predictions of various scenarios.

The results obtained with the model show that the installation is able to meet the requirement of receiving a given number of objects even if considering the availability of workstations and associated resources. Failure modeling is completely random, which makes each simulation unique from the standpoint of workstation availability. The impact of the reliability of workstations on the flow can be visible over longer periods (mitigated by regular maintenance).

The paper presents only certain scenario, but due to the wide variety of model parameters, the model allows the study of other scenarios as well. The modeling strategy adopted (starting from the most general or more specific) will allow to refine the model and gradually allocate performances / objectives to lower-level subsystems / components in the physical breakdown structure as the progress of the design advances and thus be able to simulate more precisely and progressively what is happening inside each subsystem up to the desired level of detail.

VII. Conclusion

Although models are not a perfect representation of a system, MBSE and MBSA can provide an insight and feedback earlier and at lower cost than implementation alone. This approach tested in this case study has clearly shown that the use of the simulation method is suitable for managing the development of the system by showing the weak points of the system and therefore the points for its improvement and adaptation. It has been shown that there is not always a need to go into deep detail when modeling a system. Parameters describing performance could be sufficient and subsystems could be treated as black boxes. In addition, when such a model is deployed early in the design phase, it can always be updated as the design studies progressively refine the physical breakdown structure and can finally serve as a validation / verification tool in the ascending phase of the V-cycle.

Regarding the perspectives, it would be interesting to find out how accurately SimEvents can manage the management of human resources, especially if the pace of work is not continuous. Currently, the "Resource Pool" block does not offer the option to specify whether resources will only be available for a fraction of the time. On the other hand, it allows the use of a fraction of a resource. However, further investigation would be required to answer this question.

References

[1] INCOSE, "Systems Engineering Handbook, A Guide for System Life Cycle Processes and Activities", 4th edn. Wiley, New York, 2015.

[2] O. Lisagor, T. Kelly, and Ru Niu, "Model-based safety assessment: Review of the discipline and its challenges", In the Proceedings of 2011 9th International Conference on Reliability, Maintainability and Safety, pages 625-632, June 2011.

[3] J.D. Andrews and T.R. Moss. "Reliability and Risk Assessmen", John Wiley & Sons, 1993.

[4] M. Walker and Y.Papadopoulos, "Qualitative temporal analysis: Towards a full implementation of the Fault Tree Handbook", Control Engineering Practice, 17:1115-1125, 2009.

[5] T. Prosvirnova et al., "The AltaRica 3.0 Project for Model-Based Safety Assessment", 4th IFAC Workshop on Dependable Control of Discrete Systems, University of York, September 4-6, 2013.

[6] GRIF – Graphical Interface for reliability Forecasting, www.grifworkshop.com

[7] SIMFIANEO, https://apsys-safetysecurity.com/solutions/simfianeo-software/

[8] W. J. Stewart, "Introduction to the Numerical Solution of Markov Chains", Princeton University Press, 1994.

[9] J.-P. Signoret, "Dependability & Safety Modeling and calculation: Petri Nets", In Proceeding of the 2nd IFAC Workshop on Dependable Control of Descrete Systems, DCDS 2009, Bari, Italy, June 2009.

[10] Y. Papadopoulos, M. Maruhn, "Model-Based Synthesis of Fault Trees from Matlab-Simulink Models", in International Conference on Dependable Systems and Networks (DSN), p. 77-82, 2001.

[11] A. Hazle, J. Towers, "Good Practice in MBSE Model Verification and Validation", INCOSE UK Annual Systems Engineering Conference (ASEC), 2020.

[12] D. Cook, W. Schindel, "Utilizing MBSE Patterns to Accelerate System Verification", In: IS2015 INCOSE, 2015.

[13] M.J. Simões-Marques, "Modeling and Simulation in system life cycle", Procedia Manufacturing, vol. 3, pp. 785 – 792, 2015.

[14] O. Ullrich, D. Lückerath, "An Introduction to Discrete-Event Modeling and Simulation", Simulation Notes Europe SNE 27(1), pp. 9-16, 2017.

[15] D. Antonelli, P. Litwin, D. Stadnicka, "Multiple System Dynamics and Discrete Event Simulation for manufacturing system performance evaluation", Procedia CIRP, Volume 78, Pages 178-183, 2018.

[16] G.S. Fishman, "Discrete-Event Simulation: Modeling, Programming, and Analysis", Springers series in operations research, 2001.

[17] B. Babulak and M. Wang, "Discrete Event Simulation: State of the Art", Int. Journal Online Engineering (iJOE), vol. 4, No 2, 2008.

[18] L.M.S. Dias et al., “Discrete Simulation Software Ranking – A Top List of the Worldwide Most Popular and Used Tools”, Proceedings of the 2016 Winter Simulation Conference, 2016.

[19] J.J. Swain, “Simulation Software Survey”, OR/MS Today magazine from Institute for Operations Research and the Management Sciences (INFORMS). Lionheart Publishing. 1991-2009.

[20] R.E. Nance, “A history of discrete event simulation programming languages”, In: Proceedings of the 2nd ACM SIGPLAN conference on History of programming languages, pp. 149-175,1993.

[21] A. Villemeur, ”Sûreté de fonctionnement des systèmes industriels », Collection de la Direction des études et recherches d'Electricité de France, 1997.

[22] R. F. Stapelberg, “Handbook of reliability, availability, maintainability and safety in engineering design”, Springer-Verlag London Limited, 2009.