XXVI IJCIEOM 2020

International Joint Conference on Industrial Engineering and Operations Management- ABEPRO-ADINGOR-IISE-AIM-ASEM (IJCIEOM 2020)

A Modelling Tool for Teaching and Learning Simulation

Guilherme A. B. Pereira[1], Ricardo J. Machado[1], Bruno L. S. Fernandes[1], Luís M. S. Dias[1], José A. Oliveira[1], António A. C. Vieira[1, 2]

[1] ALGORITMI Research Centre, University of Minho, Portugal

[2] CEFAGE Research Centre, University of Évora, Portugal

Abstract Teaching strategies at universities, regarding discrete event simulation, tend to solely focus on commercial tools, resulting on too much focus being applied on syntax and semantics of a particular tool, rather than on simulation basic concepts. In this context, a tool is proposed, which allows the user to model a real system using ACD (Activity Cycle Diagrams) diagrams – and automatically creating the corresponding simulation program. The effort needed to develop ACD requires a complete understanding of the real system behaviour, and simultaneously favours a good structure of that knowledge and a full comprehension of the simulation foundations. The aim of this paper is therefore to explore ways of focusing on learning simulation basics through a great effort on modelling issues.

Keywords: Teaching/learning strategies; activity worldview; discrete-event simulation; activity cycle diagrams (ACD).

1 Introduction

Simulation teaching strategies at universities, particularly regarding discrete event simulation, tend to only use commercial simulation tools. Because of this, teaching and learning basic simulation concepts, such as queues, resources, entities, activities, states, among others, may be neglected. In fact, due to the extensive use of such commercial tools at universities, the learnt/taught concepts many times consist on syntax and semantics of the particular tool being used, including programming languages. There is risk in associating simulation teaching to the use of commercial simulation tools that, in fact, do not pay adequate attention to the foundations of simulation. This could be one of the reasons to justify why simulation has not yet reached levels of professional utilization that the inherent applicability potential suggests (Altiok et al., 2001; Aquere et al., 2012).

Modern simulation tools, through the graphical representation of a real system, are already capable of automatically generating a simulation program. However, the great level of sophistication of these tools usually leads to representing system mod-el twice. First, the model is drawn, usually on a sheet of paper, in a simple formal-ism in order for the client to clearly understand. Then, another model is developed within the simulation tool, through the tool formalism that really depends on the tool used. Furthermore, modern simulation tools that generate simulation programs through the interpretation of system models, usually have their own representation formalisms, more concerned with the easy development of the program than with the comprehension of the system. Thus, the need for a common language is obvious, enabling a better understanding of the system and supporting the communication with the client.

Due to this need for a common representation language that would lead to a full comprehension and discussion about the system, and also based on the above considerations found in the literature, Activity Cycle Diagrams (ACD) could be the correct choice – in fact, ACD has been the formalism used on the development of the first simulation tool generating simulation programs (CAPS/ECSL – Computer Aided Programming of Simulation / Extended Control and Simulation Language). This tool is already discontinued and this is then a great opportunity to revisit Activity Cycle Diagrams. The effort needed to develop Activity Cycle Diagrams requires a complete comprehension of the detailed real system behaviour, and simultaneously favours a good structure of that knowledge, however the utilization of a simulation language to the development of a simulation program would become easier, after a previous construction of an Activity Cycle Diagram, for the system under analysis. Its pictorial structure enables the correct comprehension of its logic and then facilitates its communication and discussion (Freimer et al., 2004; Kang and Choi, 2011).

In this context, the aim of this paper is to present a tool that uses activity based models, interprets them, and executes it (Fernandes et al., 2014). This simulation tool would enable that the main simulation teaching focus would not be neither programming issues nor syntax or semantic aspects of the computer tool. Thus, this simulation tool would contribute to much better practices as far as simulation teaching/learning process is concerned and, therefore, would also be an important tool so as to new progress to the process of creating new simulation professional tools.

This document is structured as follows: the next section discusses the simulation tool proposed; the third presents a practical example, found on literature, which will be used in the proposed tool; and lastly, in the last section, conclusions of the con-ducted work will be presented, as well as some future work items that would be worth considering.

2 Proposed Strategy

The tool is named VBS (Visual Basic for Simulation). Following the idea dis-cussed in Pereira et al. (2009) and Pereira et al. (2009), the relevancy of creating a simulation software tool capable of generating a simulation program is based on the following general principles/premises:

1. Activity world view definitely contributes to a better understanding of the foundations of simulation, i.e., the basic concepts of simulation;
2. VBS would emphasize the utilization of a visual approach to deal with the representation of the real problem through the activity paradigm;
3. VBS would literally enable the automatic generation of a simulation program (VBA program) thus with no programming effort at all.

In addition, this new tool, including both event and activity paradigms, tries to correspond to main conclusions discussed by Stahl and his co-authors (2000; 2003). The authors stress that, the following are characteristics for teaching/learning simulation tools, specially related to criteria for simulation software to be used in education: ease of learning, ease of input, ease of reading output, ease of doing replications and experiments, safe programming, efficiency, availability and advancement potential.

This tool, as in the previous works mentioned, would again use the well-known graphical editor Microsoft Visio for incorporating the Activity Cycle Diagrams, representing a system behavior. This task would be accomplished through the creation of Visio Shapes that would reflect the different actions involved in each system simulation entity identified. Thereafter, VBA (Visual Basic for Applications) would interact with Visio, interpreting the shapes and actions associated, as well as the sequence of shapes to be "executed". At the end, this means that the effort to run simulation experiments through this tool would be equivalent to building Activity Cycle Diagrams on a piece of paper. These tasks would be performed under a newly created add-in (VBS) within Microsoft Visio – see Figure 1.

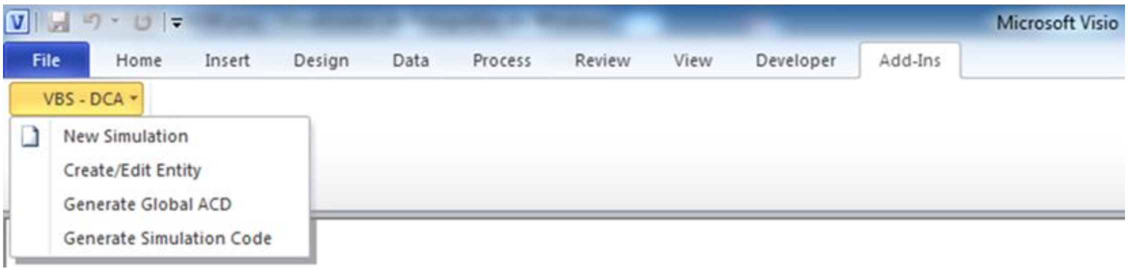

Fig. 1 New Microsoft Visio Add-In (Menu)

For this purpose, and according to the ideas presented above, the software tool would have to implement the actions for each entity. In fact, through this tool, an Activity Cycle Diagram for each entity will be created and a global Activity Cycle Diagram for the full system will be automatically generated. This global ACD will coordinate the simulation execution.

The software tool would then include a new Visio Stencil (Figure 2) with two new Visio Shapes – a Shape for defining Activities and a Shape for defining Queues.

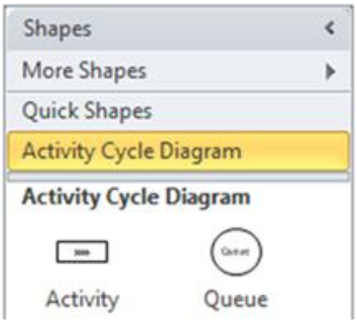

Fig. 2 New Microsoft Visio Stencil

To define an activity (see Figure 3), a Visio Shape was created, and it includes the parameters:

- Priority – if two or more activities are ready to start, this defines a priority order
- Duration – defines the duration of an activity, being deterministic or stochastic.

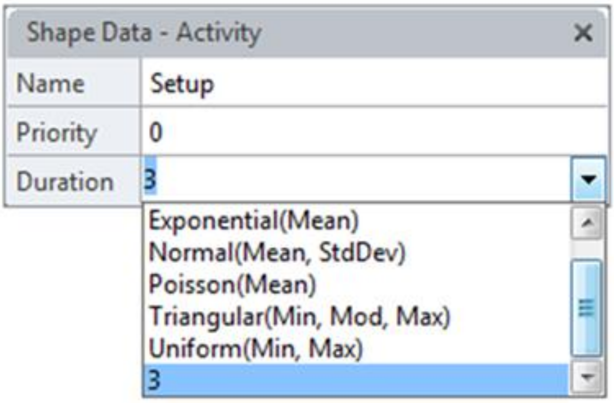

Fig. 3 Parameters for Shape Activity

To define a queue (see Figure 4) a Visio Shape was created, and it includes the parameters:

- Type – defines the queue philosophy

- Quantity – defines the number of entities in the queue when simulation starts (initial conditions)

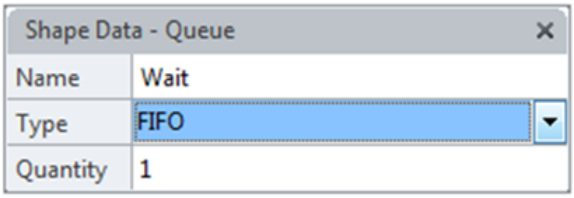

Fig. 4 Parameters for Shape Queue

The developed tool promotes the interaction between Visio shapes and VBA, which interprets each shape and corresponding parameters, interprets each Activity Cycle Diagram and generates a global Activity Cycle Diagram for the whole system, bearing in mind all interactions between each individual Activity Cycle Diagram corresponding to each type of system entities. At the end, a whole computer program is generated, which can be compiled and executed. Next section presents an application example, the corresponding interaction with the software tool.

3. Example Case

In this section, the example provided by Hutchinson (1975) will be considered. In it, three semiautomatic machines need the action of an operator to perform the machine setup and, therefore, the machine works autonomously. There is a unique operator working on the three machines and the challenge is to evaluate if this system does need another operator.

For this simple example, two entities will be considered – operator and machine (see Figure 5). Within Visio, there is a small wizard to define each necessary entity. The user can also define if the entity is external or internal – if the entity is external, the tool automatically creates a virtual ACD for a virtual entity called Door, representing the arrival process for that entity into the system, throughout the simulation.

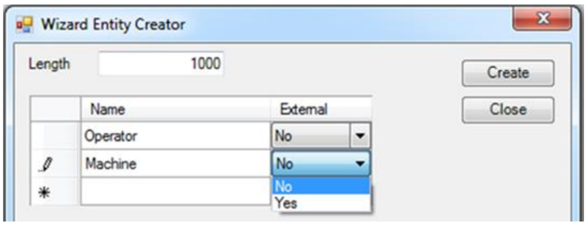

Fig. 5 VBS Wizzard

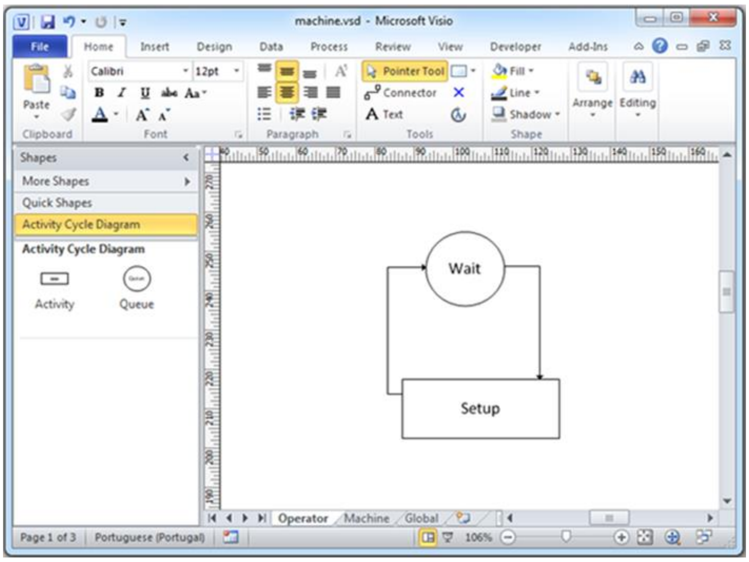

Fig. 6 ACD for Entity Operator

Thereafter, in each of the Visio sheets that the tool has included in the model (one for each entity), the user needs to develop the Activity Cycle Diagram for each entity. Having performed these tasks, the user has to go back to the VBS menu and select the option Create Global ACD. Thus, the software tool presented, based on the activity world view, makes the VBS tool more robust – incorporating both opportunities to work under event and activity paradigms.

4. Conclusions and Future Work

Simulation teaching and learning strategies to university students have not been focusing on the comprehension of simulation basics. In fact, instead, of this, they have been concentrating on syntax and semantics of the simulation tools used. In this context, the aim of this paper was to tackle this gap, by presenting a tool that is capable of: interpreting an Activity Cycle Diagrams (ACD), which represents the sys-tem to be modelled and run experiments.

In fact, authors do believe the use of this tool in a preparatory phase of a simulation course could be essential for the students to comprehend the fundamental concepts of simulation, thus students could be much better prepared to better engage in a commercial software tool and being able to get the most out of the tool. The foundations of simulation are in fact crucial for the students to develop consistent simulation models regardless of the complexity of the real problem they would face. The tool presented in this paper, based on simple activity cycle diagram concepts, could definitely contribute to a better understanding of modelling and simulation – thus making students appreciate simulation through the teaching-learning process.

As future developments for this simulation tool, the authors think that an animation interface, together with a customized output on relevant statistical performance indicators would be appropriate. At the end of this task, any activity-based simulation model constructed through this tool will integrate with animation capabilities and will produce, at the end of each simulation replication, the adequate statistical performance indicators. This could be an important step forward in this teaching/learning simulation tool. Another topic for future development is concerned with the assessment of this teaching strategy in engineering courses. In this regard, evaluation methods could be employed in order to assess the performance of students of engineering courses with and without the hereby-proposed tool.

Acknowledgements

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

References

T. Altiok et al., “Panel: academic perspectives: various ways academics teach simula-tion: are they all appropriate?,” in Proceedings of the 33nd conference on Winter simu-lation, 2001, pp. 1580–1591.

A. L. Aquere, D. Mesquita, R. M. Lima, S. B. S. Monteiro, and M. Zindel, “Coordina-tion of student teams focused on project management processes,” 2012.

M. Freimer, L. W. Schruben, T. M. Roeder, C. R. Standridge, C. M. Harmonosky, and I. Stahl, “You are going to teach simulation-now what?: Tips and strategies,” in Proceed-ings of the 2004 Winter Simulation Conference, 2004., 2004, vol. 2, pp. 2057–2065.

D. Kang and B. K. Choi, “The extended activity cycle diagram and its generality,” Sim-ul. Model. Pract. Theory, vol. 19, no. 2, pp. 785–800, 2011.

F. O. Soares, M. J. Sepúlveda, S. Monteiro, R. M. Lima, and J. Dinis-Carvalho, “An integrated project of entrepreneurship and innovation in engineering education,” Mech-atronics, vol. 23, no. 8, pp. 987–996, 2013.

S. Fernandes, D. Mesquita, M. A. Flores, and R. M. Lima, “Engaging students in learn-ing: findings from a study of project-led education,” Eur. J. Eng. Educ., vol. 39, no. 1, pp. 55–67, 2014.

G. A. B. Pereira, L. M. S. Dias, and B. L. S. Ferreira, “Flowchart simulation–a tool for the automatic generation of simulation programs,” 2009.

L. M. S. Pereira, Guilherme A. B., Dias and B. L. S. Ferreira, “Teaching Simulation Basics through Flowchart Simulation the Event Scheduling World View,” in European Modeling and Simulation Symposium, 2009.

I. Stahl, “How should we teach simulation?,” in Proceedings of the 2000 Winter Simu-lation Conference, 2000, vol. 2, pp. 1602–1612.

I. Ståhl, R. R. Hill, J. M. Donohue, H. Herper, C. M. Harmonosky, and W. D. Kelton, “Teaching the classics of simulation to beginners, panel: teaching the classics of simula-tion to beginners (panel),” in Proceedings of the 35th conference on Winter simulation: driving innovation, 2003, pp. 1941–1951.

G. K. Hutchinson, “Introduction to the use of activity cycles as a basis for system’s decomposition and simulation,” ACM SIGSIM Simul. Dig., vol. 7, no. 1, pp. 15–20, 1975.