Journal of Simulation

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tjsm20

Activity based simulation – A modeling tool for new teaching-learning strategies

Guilherme A. B. Pereira, António A. C. Vieira, Ricardo J. Machado, Bruno L. S. Ferreira, Luís M. S. Dias & José A. V. Oliveira

To cite this article: Guilherme A. B. Pereira, António A. C. Vieira, Ricardo J. Machado, Bruno L. S. Ferreira, Luís M. S. Dias & José A. V. Oliveira (2022): Activity based simulation – A modeling tool for new teaching-learning strategies, Journal of Simulation, DOI: 10.1080/17477778.2022.2032431

To link to this article: https://doi.org/10.1080/17477778.2022.2032431

Published online: 12 Feb 2022.

JOURNAL OF SIMULATION
https://doi.org/10.1080/17477778.2022.2032431

ORIGINAL PAPER

Activity based simulation – A modeling tool for new teaching-learning strategies

Guilherme A. B. Pereira[a], António A. C. Vieira[a,b], Ricardo J. Machado[a], Bruno L. S. Ferreira[a], Luís M. S. Dias[a] and José A. V. Oliveira[a]

[a]ALGORITMI Research Centre, University of Minho, Braga, Portugal; [b]CEFAGE Research Centre, University of Évora, Evora, Portugal

ABSTRACT

Teaching and learning Discrete Event Simulation (DES) at universities within the context of undergraduate studies in industrial and engineering management is not thoroughly covered in the literature. In fact, most strategies tend to solely address commercial tools, resulting on too much focus on syntax, semantics and getting used to the interface of a particular tool, rather than on simulation fundamental concepts. In the light of this, this paper proposes a tool, which allows students and professors to use Activity Cycle Diagrams (ACDs) to model systems with a comprehensive approach, focusing on the fundamental elements of simulation, allowing thereafter the corresponding simulation code to be extracted and experiments to be conducted. The tool is described and its applicability is demonstrated in some example cases. The effort needed to develop ACDs requires a complete understanding of the real system and simultaneously favours a full comprehension of the simulation fundamental elements, hence portraying an adequate teaching and learning strategy to be pondered.

ARTICLE HISTORY
Received 10 March 2021
Accepted 12 January 2022

KEYWORDS
Teaching/learning strategies; activity world view; Discrete Event Simulation; Activity Cycle Diagram

1. Introduction

With the wide use of Discrete Event Simulation (DES) commercial tools, in both academia and industry (Lang et al., 2021), teaching and learning strategies many times consist of overtaking barrier such as syntax and semantics of the particular tool being used, or even getting the student used to the particular interface (Schriber & Brunner, 2007). Hence, fundamental simulation concepts, such as queues, resources, entities, activities, states, including others queueing theory concepts may not be efficiently assimilated, or the teaching strategy may not be the most efficient (Frantzén & Ng, 2015; I Jové et al., 2014; Kang & Choi, 2011).

There is a considerable benefit when students need to work in professional contexts with a given tool (since they are already used to it); However, there is also the risk of associating simulation teaching to the use of commercial simulation tools that, in fact, do not pay adequate attention to the fundamental elements of simulation. This could be one of the reasons to justify why simulation has not yet reached levels of professional utilisation that its potential suggests. Similar issues occur, for instance, when students learn a particular tool in their classes, but in their professional context have to use a different tool, requiring them to get familiar with a new syntax and interface. Hence, the study of simulation centres on different syntaxes and getting used to different interfaces, rather than assimilating simulation concepts (Frantzén & Ng, 2015).

A well-known method to conceptually model dynamic and discrete event systems is using Activity Cycle Diagrams (ACDs; Kang & Choi, 2011). In fact, the effort needed to develop ACDs requires a complete comprehension of the detailed real system behaviour, and simultaneously favours a good structure of that knowledge. Therefore, it has been postulated that ACD is an adequate paradigm for teaching and learning simulation, due to its inherent syntax simplicity and semantic richness (Kang & Choi, 2011; Pidd, 2004).

Even in the context of using ACDs, models are usually represented twice. First, the model is conceptually drawn, using formalisms such as ACDs, in order to have a thorough overview on what will be modelled. Then, another model is developed in the simulation tool, following this tool's specific syntax. Figure 1 (top) illustrates a simple example with the basic elements that comprise an ACD. In this case, a single server system consisting of a teller that helps customer in a banking service. As customers arrive to the bank, they are either helped by the teller (if the teller is idle), or must wait. When service finishes, customers leave the system and the teller is now idle, meaning is available to help other customers on the queue.

CONTACT António A. C. Vieira avieira@uevora.pt CEFAGE Research Centre, University of Évora, Evora, Portugal

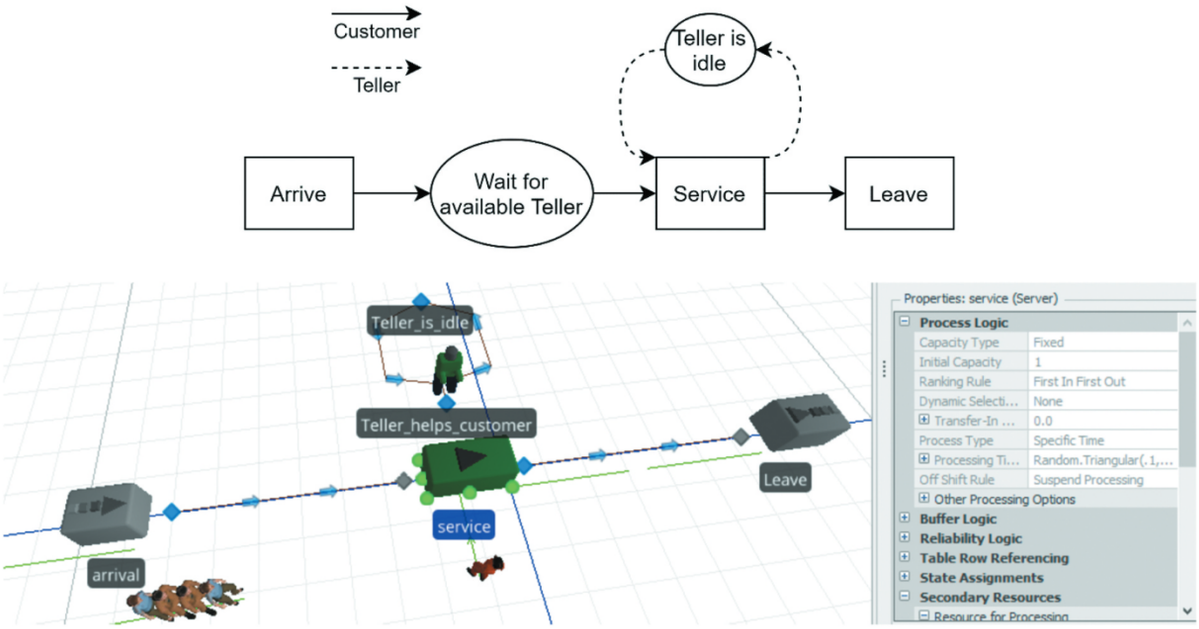

Figure 1. Example of a simple system modelled with an ACD (top) and using SIMIO (bottom).

The same example can thereafter be modelled using a commercial tool, which is illustrated in Figure 1 (bottom), in this case using SIMIO.

Notwithstanding the benefits of using ACDs to have a better understanding of the fundamental simulation elements, and a thorough vision (from a simulation perspective) of a system, the fact is that ACDs have two main limitations. First, as the size and complexity of systems increase, modelling them with ACDs also becomes complicated (Kang & Choi, 2011). Second, while they can be used to perform manual simulations (Hutchinson, 1975), they do not provide results that can be analysed. Thus, there is a need for an approach that, at the same time, allows the conceptual specifications of the model to be drawn (using ACDs), while also allowing such a model to be transcribed for a simulation tool that can conduct experiments and obtain results. Only this way shall a comprehensive simulation approach be achieved, enabling a better understanding of the system and supporting a thorough teaching/learning strategy.

In this context, the aim of this paper is as follows. It proposes a tool oriented towards allowing users, i.e., students and professors, to respectively learn and teach simulation fundamental concepts, by focusing on simulation elementary concepts, rather than in specific simulation tools' particular aspects, e.g., interface, syntax or certain workarounds. To do so, the tool allows users to model the system – using ACDs – and automatically creates the corresponding simulation program. The paper also describes the tool being used in some examples, allowing simulation models to be created from ACDs and results to be analysed.

This simulation tool would enable the main simulation teaching focus to shift away from programming issues or syntax and 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.

In addition, it is expected that this work contributes to: (i) overcoming traditional difficulties in the act of teaching and learning fundamental simulation concepts; (ii) emphasising and thoroughly understanding simulation fundamental concepts during the act of teaching and learning, rather than putting the effort on overcoming particular tools' issues; (iii) creating new opportunities for organisations to use simulation, through both motivating university students and attracting organisation's staff to simulation.

This paper is structured as follows. Next section covers the literature review conducted; the third section presents the simulation tool proposed; the fourth concerns a practical example, found on literature, which will be used in the presented tool. Fifth section provides a discussion of our experience using the proposed tool; and the last section presents the main conclusions of the conducted work, as well as some future work items that would be worth considering.

2. Literature review

It is interesting to note the rare appearance in the overall scientific community of studies mentioning ACDs, as searching for this term in Scopus results in only 73 results, without applying any other filters. Engineering education should engage students of

engineering courses (Fernandes et al., 2014) and help them to develop the required skills to confidently and successfully handle problems and design effective solutions (Aquere et al., 2012; Fernandes et al., 2012; Soares et al., 2013). However, regarding simulation, there is a lack of teaching/learning tools that would help students to achieve the intended success.

Schriber and Brunner (2007) convincingly support that a "black box" approach is often taken in teaching and learning simulation software, wherein teaching and learning usually tend to neglect the full comprehension of simulation fundamental concepts. Schriber and Brunner (2007) also state that the foundation on which the software is based is ignored or is touched on only briefly, resulting in the modeller possibly not being able to efficiently approach complex modelling situations. From the cited authors, it can be argued that simulation teaching strategies should use tools that allow modellers to have a full comprehension of systems to be modelled.

Simulation tools rankings as in Dias et al. (2016), concerning commercial simulation tools, could be found in the literature. However, it is not the case for teaching/learning simulation tools. In fact, the main purpose of teaching/learning simulation is quite different from the main issues covered by commercial simulation tools, traditionally used while teaching simulation to university students (Freimer et al., 2004). Commercial tools, dedicated to solve heterogeneous problems that organisations are facing, are full of complex mechanisms that do not favour the learning process and do not focus on the basics of the simulation technique (Herper & Stahl, 1999). Thus, one of the main reasons concerning the lack of popularity of simulation among university students is related to the high emphasis that is put on programming knowledge in simulation disciplines, sometimes programming skills are even pre-requisites to work with simulation (Frantzén & Ng, 2015; Stahl et al., 2003). Moreover, many times the emphasis is put on the simulation tool, rather than on understanding the problem itself (I Jové et al., 2014). Therefore, in order to give the simulation technique new perspectives and challenges, it is crucial to attract potential simulation users. However, engineering and management students usually do not have the required programming competences, therefore contributing to the lack of engineers working on this field. This further culminates in a lack of capability to comprehend the strengths and limitations of the simulation technique (Stahl, 2000).

The process of teaching simulation usually follows two different approaches: one emphasises theoretical aspects and the other focuses on practical simulation issues. In the literature one would not be able to find a magic formula to better balance these two approaches. However, Altiok et al. (2001), Nance (2000), and Stahl et al. (2003) do present and discuss advantages and disadvantages for both approaches. It is evident from these authors that, besides the recognition that the practical aspects of simulation are very important, the simulation fundamental elements are crucial, so that students become capable of modelling complex problems and also for them to be able to develop future simulation tools.

As found in the literature, few simulation tools would have this capability for teaching purposes. General Purpose Simulation System (Stahl et al., 2011) is based on the process paradigm. In this process-oriented tool users can, in fact interact with some of the fundamental elements of any simulation (e.g., simulation clock). However, it often requires the already mentioned, and frequent, use of programming code to produce simulations.

In Dias et al. (2008), an interesting event approach was presented, revisiting the traditional Basic Simulation Facility (BSF) simulation tool, which is a tool that employs the event paradigm. This tool uses the creation of flowcharts (focused on the event paradigm) with simple options to edit some of the main parameters of each block of the flowchart. Thereafter, the corresponding simulation program can be compiled. This is, in fact, an interesting tool for teaching/learning purposes; however, it does not allow the possibility of modelling systems based on ACDs, which is a commonly used approach in the simulation field (Kang & Choi, 2011). Another interesting approach using event graphs to conceptually model systems is SIGMA, proposed by Schruben (1983).

Despite the different paradigms that exist within the available simulation Pidd (1992, 2004), Dias et al. (2006) and Kang and Choi (2011), clearly state that the activity-based paradigm is the most adequate paradigm for teaching simulation purposes due to the inherent simplicity and syntax and semantic richness. It is a concept historically poorly explored by simulation tools manufacturers (Nance, 1995). In the light of this and the fact that the herein proposed tool allows the creation of ACDs, and the respective conversion into simulation programming code, establishes the benefits said tool in advancing the State of the Art, regarding adequate strategies for teaching and learning simulation.

3. Developed simulation tool

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

Activity world view definitely contributes to a better understanding of the fundamental concepts of simulation;

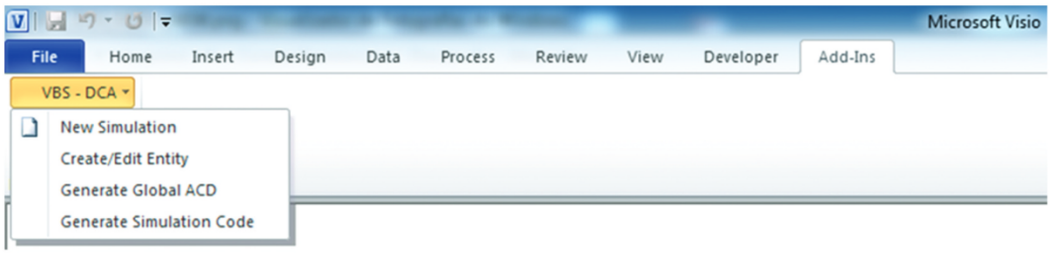

Figure 2. New Microsoft Visio Add-In (Menu).

VBS would emphasise the utilisation of a visual approach to deal with the representation of the real problem through the activity paradigm;

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 includes both event and activity paradigms, and incorporates some relevant features that had been discussed by Stahl (2000), namely: 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 ACDs, representing a system behaviour. 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 ACDs on a piece of paper. These tasks would be performed under a newly created add-in (VBS) within Microsoft Visio – see Figure 2.

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 ACD for each entity will be created and a global ACD 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 3) with two new Visio Shapes – a Shape for defining Activities and a Shape for defining Queues.

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

Priority – if two or more activities are ready to start, this parameter defines a priority order

Duration – defines the duration of an activity, being either deterministic or stochastic.

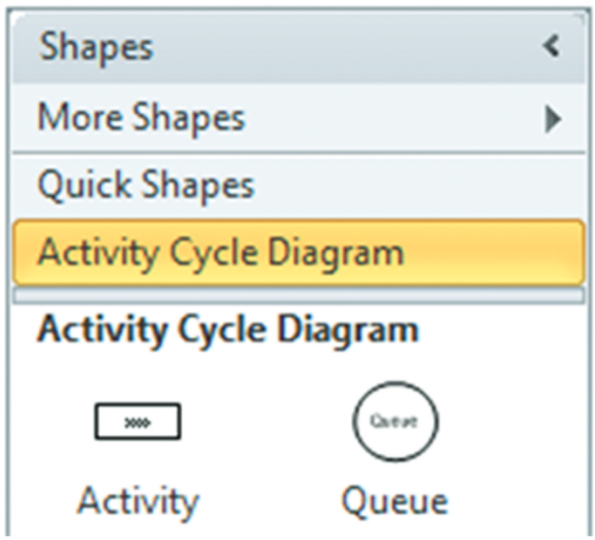

Figure 3. New Microsoft Visio Stencil.

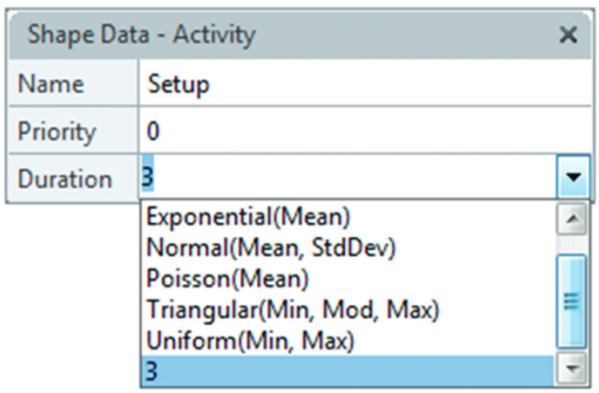

Figure 4. Parameters for shape activity.

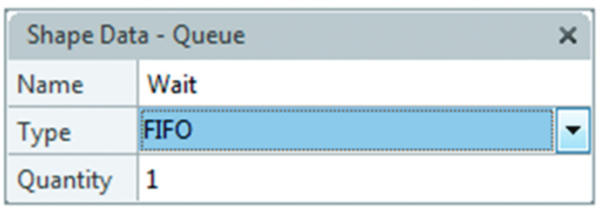

Figure 5. Parameters for shape queue.

To define a queue (see, Figure 5) 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)

Users (students or professors) need to create an ACD for each type of entity in the system, so as to specify the activities performed by each one. After the user has placed and connected the shapes and edited its properties for each ACD, the developed tool uses this information to create a single ACD using the elements (e.g., queues or activities) in common in each ACD developed by the user. Having created this global ACD, the tool can generate the respective simulation program that can run simulations and retrieve results. Next section presents an application example, the corresponding interaction with the software tool, as well as the generated computer code (VB).

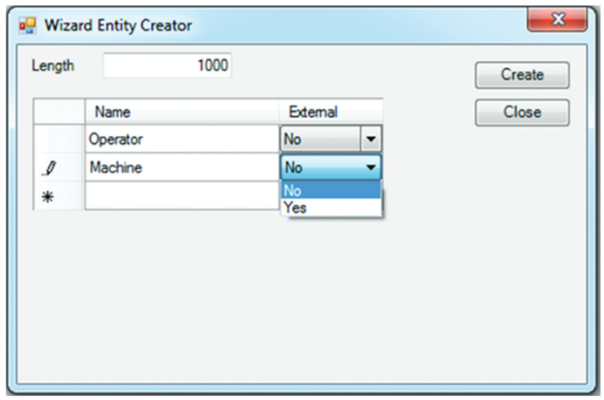

Figure 6. VBS wizard.

4. Application example

This section described the application of the developed tool when considering an adaptation of the case provided by Hutchinson (1975). Thus, the next subsection briefly describes the case to be considered in the application example, while the second subsection describes how to use the tool to model this system using the ACDs approach. Finally, the last subsection describes the type of analysis that is possible to perform with this tool.

4.1. Example description

The system in question consists of a semi-automatic production environment, where there are 3 machines available to perform a given operation, although they require a manual setup task to be performed by an operator. Material arrives following a negative exponential with an average of 1 material every 2 minutes. As materials arrive requiring the intended operation to be performed on them, the operator prepares one of the 3 machines – whichever is available at the time – taking 3 minutes to do so (in this case this was the considered time, albeit different values can be considered, including statistical distributions to generate random numbers, similarly to the interarrival time). When the selected machine is ready, it autonomously operates on the material during 10 minutes. When the operation of a machine ends, the material has finished its production flow. The next subsection describes the approach that models this system in the proposed tool.

4.2. Modelling approach

This subsection describes the approach that the tool employs to model the system described in the previous subsection. For this simple example, two entities will

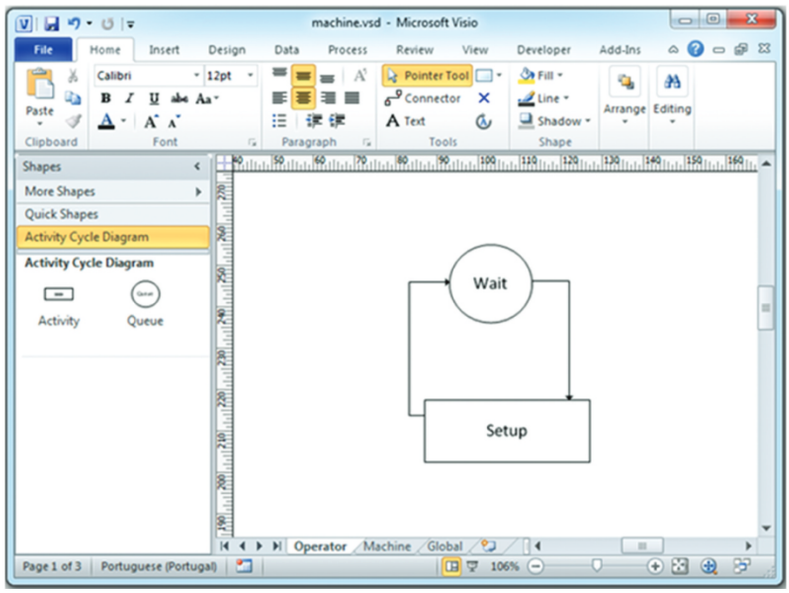

Figure 7. ACD for entity operator.

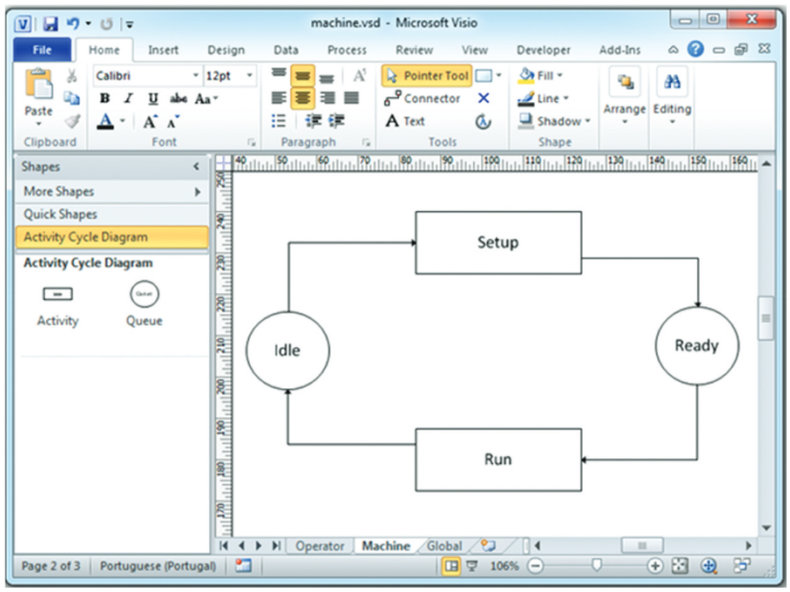

Figure 8. ACD for entity machine.

be considered – operator and machine (see Figure 6). 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 (meaning that it originates outside the system, i.e., it is periodically created), the tool automatically creates a virtual ACD for a virtual entity named Door, representing the arrival event for that entity into the system, throughout the simulation.

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 ACD for each entity – see below, Figures 7 and 8, for both ACDs for the mentioned entities.

Having performed these tasks, the user just has to go back to the VBS menu and select the option Create Global ACD. The result would then be the one described in Figure 9.

At this stage, the user can again come back to the VBS menu and select the option Generate Simulation Code. The tool thereafter automatically creates VB code for this system – as shown in Figure 10 – and also interprets that code, runs it (performing the simulation for the previously defined simulation time), and finally presents the simulation results, which are covered in the next subsection.

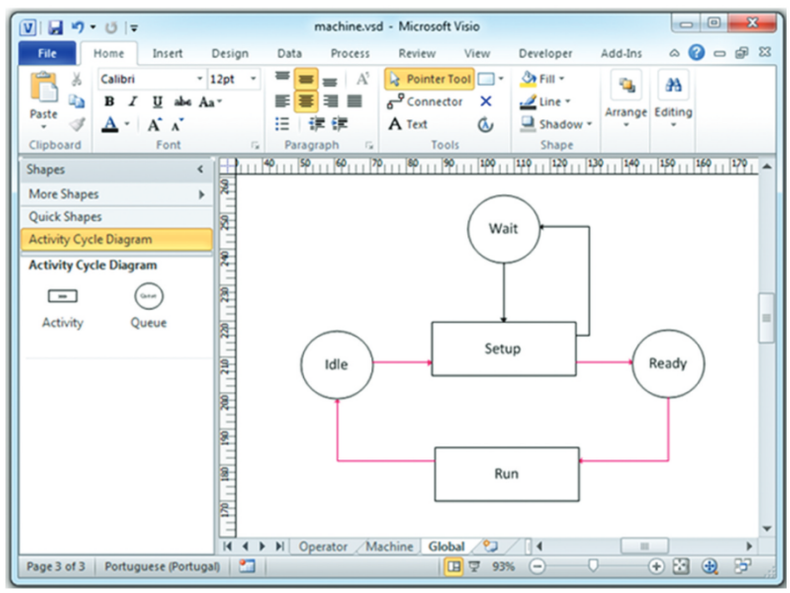

Figure 9. Automatically generated global ACD.

Source Code Generated:

```
Module Module1
    Public simulation As Simulation = Nothing
    Public entityOperator As EntityType = Nothing
    Public entityMachine As EntityType = Nothing
    Public WithEvents activitySetup As Activity = Nothing
    Public WithEvents activityRun As Activity = Nothing
    Public queueWait As Queue = Nothing
    Public queueIdle As Queue = Nothing
    Public queueReady As Queue = Nothing

    Sub Main()
        simulation = New Simulation()
        simulation.Length = 1000
        entityOperator = simulation.CreateEntity("Operator")
        entityOperator.Attributes.Create("IsExternal", 0)
        entityMachine = simulation.CreateEntity("Machine")
        entityMachine.Attributes.Create("IsExternal", 0)
        activitySetup = simulation.CreateActivity("Setup")
        activityRun = simulation.CreateActivity("Run")
        queueWait = simulation.CreateQueue("Wait", entityOperator)
        queueIdle = simulation.CreateQueue("Idle", entityMachine)
        queueReady = simulation.CreateQueue("Ready", entityMachine)
        queueWait.Insert(entityOperator.Generate(1))
        queueIdle.Insert(entityMachine.Generate(3))
        activitySetup.Resources.AddDependency(entityOperator, 1)
        activitySetup.Resources.AddDependency(entityMachine, 1)
        activityRun.Resources.AddDependency(entityMachine, 1)
        activitySetup.Resources(entityOperator).AddSource(queueWait)
```

Refresh Code | Run Code | Export Code | 0 Errors | 0 Warnings

Figure 10. Partial automatically generated computer code (VB).

Simulation Report When Time = 997

Statistics | History

Id	Activity	ScheduledAt	ExecutedAt	TimeDiff	Resources
1	Setup #1	0	3	3	Operator1,Machine1
2	Setup #2	3	6	3	Operator1,Machine2
3	Run #1	3	13	10	Machine1
4	Setup #3	6	9	3	Operator1,Machine3
5	Run #2	6	16	10	Machine2
6	Run #3	9	19	10	Machine3
7	Setup #4	13	16	3	Operator1,Machine1
8	Setup #5	16	19	3	Operator1,Machine2

Figure 11. Historical data presented.

4.3. Results analysis

This subsection addresses the results that can be obtained with the tool. In this regard, and having conducted the steps described in the previous subsection, the user (i.e., student or professor) can either analyse historical data or simulation results. The former can be useful to interact with students (or for students to analyse) so that the performed actions by the simulator and the role of each fundamental element (i.e., queues, simulation clock, resources and others) is thoroughly assimilated. See, Figure 11 for a sample of the historical data generated by the automatically created simulation program.

Simulation Report

Statistics | History

Name	Type	IN	OUT	Now	AvgStay	AvgLen
op_queue	Queue	320	319	1	19,0878	6,1081
mch_queue	Queue	318	298	20	34,6913	11,0318
operator	Resource	319	318	1	3	0,957
machine	Resource	298	295	3	10	2,98

Figure 12. Simulation outputs.

Apart from the historical data, the tool also allows simulation outputs to be analysed, i.e., after the execution of the simulation time that was defined by the user, in this case 1000 minutes. The respective outputs obtained in this case are displayed in Figure 12.

The results are displayed using the developed tool's interface, which, for each defined resource and queue, displays the following outputs:

- IN: represents the total number of entities that entered each queue or resource throughout the simulation un;
- OUT: represents the total number of entities that exited each queue or resource throughout the simulation un;
- Now: represents the total number of entities that remained on each queue or resource at the end of the simulation run;
- AvgStay: represents the average number of entities that occupied the queues and resources throughout the simulation run;
- AvgLen: for the queues represents the average size that was used throughout the simulation, and, for the resources, represents the average number of resource units that were occupied throughout the simulation run.

Thus, another major and commonly used output – resource utilisation – can be easily calculated through the following way:

$$Resource\ utilization = \frac{average\ number\ of\ utilized\ resource\ units}{total\ number\ of\ resource\ units}$$

Hence, for this case, the resource utilisation of the operation is 95.7%, since only 1 operator was considered, and the resource utilisation of the machines equalled 99.3%, as 3 machines existed. Naturally, these excessive resource utilisation rates are the result of the interarrival time that was considered, which also affects the average queue length. Finally, it should also be noted that these performance indicators are static for all models, meaning that students do not need to implement them. While this also means that the set of performance indicators that can be used is limited, the purpose of the tool is to help students and professors in the simulation teaching endeavour, namely for the modelling of the first systems, which should not be too complex.

5. Discussion

The tool presented in this paper was developed by the authors, for the purposes described in this paper. In other words, to propose an alternative tool and approach to teach simulation. In fact, we have used it in our classes for such purposes, namely we let our students explore this tool during the initial weeks, to model systems with the emphasis on exploring the fundamental concepts of simulation.

Our experience using this tool consisted of using it as the first and only simulation tool that students had contact with, during the initial 4 to 5 weeks of classes, in curricular units where simulation of systems is the main subject. After this initial contact, students would develop a project, consisting of modelling a problem and analysing the respective results. Afterwards, students would migrate to a commercial tool, which in our case is Simio, as we have extensively used it in classes and in research projects at our University. Before this approach, Simio was the first real contact that students had with the simulation concepts in practice. Hence, they would need to understand such concepts, while also getting familiar with the tool's interface and syntax.

The above-described shift in the teaching methodology revealed interesting results, i.e., considering the feedback from our students, the results and their involvement in assessments. As such, and given these interesting results, the authors are keen on continuing this project, namely by extending this paper to demonstrate the interesting results we have already accomplished.

However, and despite the encouraging results, the tool has not yet been publically available for download; however, we do intend to do so in the near future, hence allowing the community to try, and explore the tool and the approach described in this paper.

6. Conclusions

Discrete Event Simulation (DES) teaching and learning strategies for university students have not been focusing on the comprehension of simulation basics. In fact, instead, of this, they have concentrated 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 system to be modelled; run the model; and present results. This section presents the main conclusions of this research. Thus, the first subsection discusses the findings and main implications, while the last subsection presents the major future research that can derive from this work.

6.1. Implications

The software tool presented, based on the activity world view, incorporates opportunities to both work under event and activity paradigms, maintaining three particularly interesting features, namely:

It is based on simple diagrams;

It automatically generates a VB computer program to perform the mimic of the system under analysis;

It runs the model directly over the automatically generated global system diagram, producing debugging trace files.

These features, together, would still contribute to:

the generalisation and a better understanding of the use of simulation;

the full comprehension of the basics of simulation;

the automatic generation of simulation programs.

In brief, it can be argued that the generalisation and better understanding of the use of simulation would have been accomplished, since this new tool would only require expertise on a basic simulation approach – activity-based approach – to describe real system behaviours.

Furthermore, these diagrams, apart from providing an understanding of the behaviour of the system, also contribute to the comprehension of fundamental simulation concepts, such as entities, queues, activities, resources and also to a very important simulation concept – the evolution of the state of the system over time.

Finally, these simple diagrams are translated into the software tool by means of an automatic generation of a computer program that performs the mimic of the system and evaluates the corresponding efficiency measures. The simulation runs over the automatically generated global ACD, step by step, enabling the user to gradually assimilate concepts. While commercial simulation tools also allow simulation models to be built by using small components (e.g., building blocks or process steps/activities), this tool uses ACDs as the basis, which allows students and professors to focus on the fundamental concepts of simulation, rather than in specific and tool-oriented interfaces, syntaxes or other elements.

In fact, the 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. These fundamental concepts 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 ACD concepts, could definitely contribute to a better understanding of modelling and simulation – thus making students appreciate simulation through the teaching-learning process. In the light of this, the approach herein proposed could be used in real contexts, e.g., by starting to blend the traditional approach, of using a commercial tool, with the herein proposed approach.

6.2. Limitations and future research

The case that was addressed in this paper represents a dynamic system, where the fundamental elements of simulation can be analysed. However, such case is, indeed, not too complex. In fact, as this tool is oriented towards teaching/learning purposes, its purpose is not to model and analyse real and complex cases, but rather in academic contexts. In such contexts, the authors believe it should be used in a first approach (during the first weeks), before migrating to another more advanced tool (e.g., a commercial tool) to model more advanced systems. At this stage, students and professors shall inevitably concentrate also on syntax issues (perhaps particular to the specific tool in question), however the fundamental simulation concepts would have been assimilated in a solid way.

The major limitation of this is regarding the simulation outputs that can be extracted from this tool. As discussed, these are limited to a set of standard performance indicators. While different outputs can be implemented, the ones that are already available allow students to venture on their initial systems and assimilate the fundamental elements of simulation.

As future developments for this simulation tool, an animation interface could be developed. The importance of animation has been extensively recognised by the community, and students tend to be motivated by such features. Thus, apart from generating the simulation code, the tool could also allow the user to visualise an animated environment, separating the three layers – ACD, simulation code and animation environment. This could be an important step forward in this teaching/learning simulation tool. Apart from this, other elements could be developed. Nevertheless, a major step shall remain, consisting of providing the means to, from the output provided by this tool, produce a simulation model in a commonly used simulation tool, such as commercial ones, like Arena, Simio, AnyLogic, and so on.

Another topic for future development is concerned with the assessment of this teaching strategy in engineering courses. In fact, despite the potential benefits, empirical evidence would be needed. In this regard, evaluation methods could be employed in order to assess the performance of students of engineering courses, perhaps with and without the herein proposed approach.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work has been supported by FCT (Fundação para a Ciência e Tecnologia) within the R&D Units Project Scope [UIDB/00319/2020], and in the scope of the project [UIDB/04007/2020]..

ORCID

António A. C. Vieira http://orcid.org/0000-0002-1059-8902

References

Altiok, T., Kelton, W. D., L'Ecuyer, P., Nelson, B. L., Schmeiser, B. W., Schriber, T. J., Schruben, L. W., & Wilson, J. R. (2001). Panel: Academic perspectives: Various ways academics teach simulation: Are they all appropriate? *Proceedings of the 33rd Winter Simulation Conference*, Arlington, VA, USA, (IEEE).

Aquere, A. L., Mesquita, D., Lima, R. M., Monteiro, S. B. S., & Zindel, M. (2012). Coordination of student teams focused on project management processes. *International Journal of Engineering Education*, 28(4). https://www.researchgate.net/publication/256733052_Coordination_of_Student_Teams_Focused_on_Project_Management_Processes .

Dias, L., Pereira, G., & Oliveira, J. A. (2006). Activity based modelling with automatic prototype generation of process based arena models. *2nd European Modeling and Simulation Symposium*. Barcelona, Spain, (CAL-TEK SRL). https://www.proceedings.com/20406.html .

Dias, L., Pereira, G., & Oliveira, J. A. (2008). Event scheduling made easy: Basic simulation facility revisited. *20th European Modeling and Simulation Symposium*, Campora San Giovanni, Amantea (CS), Italy, CAL-TEK SRL. https://www.proceedings.com/20408.html .

Dias, L., Vieira, A. A. C., Pereira, G., & Oliveira, J. A. (2016). Discrete simulation software ranking – A top list of the worldwide most popular and used tools. *Proceedings of the 2016 Winter Simulation Conference*, Washington, DC, USA, IEEE.

Fernandes, S., Flores, M. A., & Lima, R. M. (2012). Students' views of assessment in project-led engineering education: Findings from a case study in Portugal. *Assessment & Evaluation in Higher Education*, *37*(2), 163–178. https://doi.org/10.1080/02602938.2010.515015

Fernandes, S., Mesquita, D., Flores, M. A., & Lima, R. M. (2014). Engaging students in learning: Findings from a study of project-led education. *European Journal of Engineering Education*, *39*(1), 55–67. https://doi.org/10.1080/03043797.2013.833170

Frantzén, M., & Ng, A. H. C. (2015). Production simulation education using rapid modeling and optimization: Successful studies. *Proceedings of the 2015 Winter Simulation Conference*, Huntington Beach, CA, USA, IEEE.

Freimer, M., Schruben, L. W., Roeder, T. M., Standridge, C. R., Harmonosky, C. M., & Stahl, I. (2004). You are going to teach simulation: Now what? Tips and strategies. *Proceedings of the 36th Winter Simulation Conference*, Washington, DC, USA, IEEE.

Herper, H., & Stahl, I. (1999). Micro-GPSS on the Web and for Windows: A tool for introduction to simulation in high schools. *Proceedings of the 1999 Winter Simulation Conference*, Phoenix, AZ, USA, IEEE.

Hutchinson, G. K. (1975). Introduction to the use of activity cycles as a basis for system's decomposition and simulation. *ACM SIGSIM Simulation Digest*, *7*(1), 15–20. https://doi.org/10.1145/1102722.1102725

I Jové, J. F., Petit, A. G., Casas, A. P., & Casanovas-Garcia, J. (2014). Teaching system modelling and simulation through petri nets and arena. *Proceedings of the 2014 Winter Simulation Conference*, Savannah, GA, USA, IEEE.

Kang, D., & Choi, B. H. (2011). The extended activity cycle diagram and its generality. *Simulation Modelling Practice and Theory*, *19*(2), 785–800. https://doi.org/10.1016/j.simpat.2010.11.004

Lang, S., Reggelin, T., Muller, M., & Nahlas, A. (2021). Open-source discrete-event simulation software for applications in production and logistics: An alternative to commercial tools? *Procedia Computer Science*, *180*(2021), 978–987. https://doi.org/10.1016/j.procs.2021.01.349

Nance, R. E. (1995). Simulation programming languages: An abridged history. *Proceedings of the 27th Winter Simulation Conference*, Arlington, VA, USA, IEEE.

Nance, R. E. (2000). Simulation education: Past reflections and future directions. *Proceedings of the 2000 Winter Simulation Conference*, Orlando, FL, USA, IEEE.

Pereira, G., Dias, D., & Ferreira, B. L. S. (2009a). Flowchart simulation – A tool for the automatic generation of simulation programs. *IX Congreso Galego de Estatística e Investigación de Operacións, SGAPEIO2009*, Ourense, Espanha, November. 12-14. pp 129–134. https://dialnet.unirioja.es/servlet/libro?codigo=777642 .

Pereira, G., Dias, L., & Ferreira, B. L. S. (2009b). Teaching simulation basics through flowchart simulation the event scheduling world view. *EMSS 2009 - European Modeling and Simulation Symposium, part of IMMM09*. Tenerife, Espanha, Set., 2009. pp 1–8. https://www.scimagojr.com/journalsearch.php?q=21100229189&tip=sid&clean=0 .

Pidd, M. (1992). *Computer simulation in management science* (3rd ed.). Wiley.

Pidd, M. (2004). Simulation worldviews: So what? *Proceedings of the 36th Winter Simulation Conference*, Washington, DC, USA, IEEE.

Schriber, T. J., & Brunner, D. T. (2007). Inside discrete-event simulation software: How it works and why it matters. *Proceedings of the 39th Winter simulation Conference: 40 years! The best is yet to come*, Savannah, GA, USA, IEEE.

Schruben, L. (1983). Simulation modeling with event graphs. *Communications of the ACM*, *26*(11), 957–963. https://doi.org/10.1145/182.358460

Soares, F. O., Sepúlveda, M. J., Monteiro, S., Lima, R. M., & Carvalho, J. D. (2013). An integrated project of entrepreneurship and innovation in engineering education. *Mechatronics*, *23*(8), 987–996. https://doi.org/10.1016/j.mechatronics.2012.08.005

Stahl, I. (2000). How should we teach simulation? *Proceedings of the 2000 Winter Simulation Conference*, Orlando, FL, USA, IEEE.

Stahl, I., Born, R. G., Henriksen, J. O., & Herper, H. (2011). GPSS 50 years old, but still young. *Proceedings of the 2011 Winter Simulation Conference*, Phoenix, AZ, USA, IEEE.

Stahl, I., Hill, R. R., Donohue, J. M., Herper, H., Harmonosky, C. M., & Kelton, W. D. (2003). Teaching the classics of simulation to beginners, panel: Teaching the classics of simulation to beginners (panel). *Proceedings of the 35th Winter simulation Conference: driving innovation*, New Orleans, LA, USA, IEEE.