Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 180 (2021) 978–987

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Industry 4.0 and Smart Manufacturing

Open-source discrete-event simulation software for applications in production and logistics: An alternative to commercial tools?

Sebastian Lang[a,b], Tobias Reggelin[a,b], Marcel Müller[b], Abdulrahman Nahhas[b]

[a]*Fraunhofer Institute for Factory Operation and Automation IFF, Sandtorstr. 22, 39106 Magdeburg, Germany*
[b]*Otto von Guericke University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany*

Abstract

Discrete-event simulation is an established method to support decision making for planning tasks in production and logistics. However, there are still many enterprises, especially smaller companies that do not use discrete-event simulation because of the high costs associated with buying and maintaining commercial simulation tools. The question is whether or not free discrete-event simulation software is an alternative to commercial tools for solving typical planning tasks in production and logistics. The paper analyzes the modeling process with the three free and open-source discrete-event simulation tools Salabim, JaamSim and CloudSim and compares them with the two standard commercial simulation packages Arena and Plant Simulation. JaamSim provides everything which is necessary to model typical planning tasks in production and logistics and proves as a real alternative to commercial discrete-event simulation tools.

Keywords: discrete-event simulation, open-source, production, logistics

1. Introduction

Discrete-event simulation (DES) is a well-known method to support decision-making in production and logistics. According to our experience, DES is mostly established in large enterprises, but only rarely utilized by small- and medium-sized companies. The high investment and maintenance costs for a commercial discrete-event simulation (C-DES) tool might be one reason that smaller companies hesitate to consider DES for their production and logistics planning. Depending on the features required, a C-DES tool can cost anywhere from a few thousand up to ten thousands of US dollars. In addition, there are usually annual costs for service and maintenance, which also often

10.1016/j.procs.2021.01.349

amount to several thousand US dollars per year. The high costs are not only a barrier for companies, but also for universities that want to include modeling and simulation in their curriculum. Although many providers of C-DES tools offer a free version for personal learning, the license agreement often prohibits the use on several computers for teaching classes.

Free and open-source discrete-event simulation (OS-DES) software could be an interesting alternative for both small- and medium-sized companies and educational institutions. While the use of open-source software is quite common in other areas of engineering, our experience indicates that there is not much knowledge about OS-DES software for production and logistics applications. Our assumption is supported by the fact that the ranking of the worldwide most popular DES tools does not contain any OS-DES software [1]. The contribution of this paper is to fill this gap of knowledge by providing an exemplary evaluation of free OS-DES software in terms of utility for industrial and academic applications. For this purpose, we compare three OS-DES tools (Salabim, JaamSim and CloudSim) and two C-DES (ARENA and Plant Simulation) based on the implementation of a reference problem case. In the conclusions, we summarize our experience by evaluating the tools in terms of their capabilities and usability.

2. Related research

To the best of our knowledge, there is only one publication that provide an overview about OS-DES software [2]. These authors analyze 44 OS-DES tools in terms of their suitability to support decision-making in operations research, respectively manufacturing, services, supply chain management and logistics. The identified tools are classified in terms of programming language, license type, version control system and means of communication between developers and the user community. The authors exclude 34 of the 44 OS-DES tools, for instance due to the lack of project maintenance or because of their inadequacy for problems in operations research. The remaining tools, which the authors analyze in detail, are DESMO-J, JaamSim, JAPROSIM, NS-3, OMNeT++, PowerDEVS, SharpSim, SimPy and URURAU. The authors draw some interesting conclusions. OS-DES software seems to be widely used to simulate computer networks and intercommunication of computer systems. In contrast, the circle of users for applications in the field of operations research seems to be very small. Furthermore, there are many papers describing OS-DES software in detail, but only a small number of articles describe the application of OS-DES on industrial problems. The findings of the literature review also indicate that OS-DES software is mainly used by computer scientists and not by industrial engineers, who in turn are a major user group of C-DES tools. Thus, the results of confirm our assumption that the existence of OS-DES software is virtually unnoticed by production and logistics planners.

3. Design of the comparative study

We planned our study in three steps. First, we decided about the OS-DES and C-DES tools to be compared. Second, we designed the reference problem case. Third, we defined evaluation criteria for the comparison of the different tools.

3.1. Selection of the OS-DES and C-DES tools

Several criteria were taken into account for the selection of the OS-DES tools. In the following, we want to list some of the major criteria that influenced our decision:

- capabilities to model production and logistics problems
- support and maintenance of the software
- existence of a software documentation
- size of the user community
- tools for animation

Furthermore, it was important to us that the selected OS-DES tools differ in terms of their individual strengths and weaknesses. We finally chose Salabim, JaamSim and CloudSim for our comparison. We provide a short explanation for the selection of each OS-DES tool in the further course of this article.

The primary criterion for the selection of the first C-DES tool was the number of users as we wanted to compare the OS-DES software packages with at least one commercial tool, which is widely known in the DES user community. We therefore decided for ARENA, which has still the top rank in the list of the worldwide most popular and utilized DES tools [1]. For the selection of the second C-DES tool, it was important for us that it differed from ARENA in terms of the process of modeling. We selected as second C-DES tool Plant Simulation, which is the standard tool for DES in the German automotive industry. While modeling in ARENA relies on the business process modeling notation, the modeling elements in Plant Simulation are based on material handling technology. Thus, complex modeling logic's in ARENA are inherent to the structure and connections of modeling elements. In Plant Simulation, on the other hand, the programming of scripts is preferred for the design of complex process controls.

3.2. Conception of the reference problem case

The reference problem case is a modeling task that requires the utilization of all modeling features common to DES and usually provided by any C-DES tool, such as:

- time-flexible and type-flexible creation of entities
- setting of entity specific attributes
- queuing of entities
- delaying of entities
- seizing and releasing of process resources
- consideration of sequence depending setup times
- customization of model control logics

Based on these considerations, we decided for a manufacturing system with three levels as modeling task. Figure 1 shows the conceptual model of the system. Entities generated by the source are to 70 percent of type "A" and to 30 percent of type "B". The setup times of the machines on the main processing level are sequence-dependent and in a range between 0 and 4.5 minutes. The upper workstation on the last stage does only process entities of type "A", while the lower workstation only receives entities of type "B". A workstation requires an operator during the complete process of one entity. However, there is only one operator for both workstations.

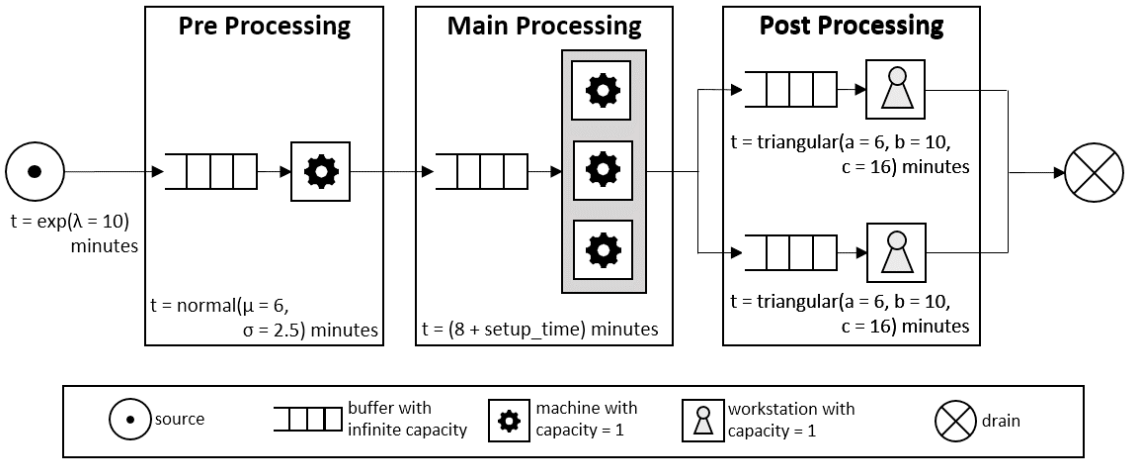

Fig. 1. Conceptual model of the reference problem case.

3.3. Definition of evaluation criteria

Any DES tool for industrial applications has to fulfill certain requirements in terms of capabilities, usability and computational efficiency. We assessed the capabilities of the selected DES tools based on their modeling libraries and their features for:

- evaluating simulation runs
- analyzing input data
- conducting optimization studies
- programming custom model and control logic
- debugging simulation models
- animation / visualization of models
- exchanging data with other applications

The usability of a DES tool has a significant impact on the time a modeler requires to implement a particular problem as a simulation model. Therefore, the usability directly affects the costs for conducting a simulation study. The following aspects were important to us to assess the usability of each DES tool:

- simplicity and comprehensibility of the modeling process
- availability of a graphical user interface
- scope and level of detail of the documentation
- availability of user support (mailing list, online community)
- maintenance of the software

Furthermore, there are additional criteria that only apply to O-DES or C-DES tools. For the selection of an O-DES tool, it is important to know about the license terms, respectively if they allow the creation of simulation models for industrial and commercial purposes. For the selection of a C-DES tool, the investment and maintenance costs are of particular importance.

4. Conducting of the comparative study

4.1. Arena

Arena is currently the most popular DES software [1]. Arena is especially common in North America and some Latin American countries such as Colombia and Peru. The Systems Modelling Corporation and Rockwell Automation developed Arena. Although Arena is a commercial software, the company also offers a free trial version for students that offers full functionality without time limit. Only the model size is limited and any commercial use is strictly prohibited.

We used Arena in the version 15.10.00004. The development of the example model in Arena took only a few hours. The model structure is based on the classical material flow from left to right. Figure 2 shows the structure of the model with its branches. Some of these branches are only necessary because Arena is not specifically based on production logistics and therefore setup processes had to be modeled with general logic blocks and references to global variables.

In detail, for example, the assignment of product types was easy to implement using the "Assign" block. The material flow had to be divided up first, in order to allow the right product mix at the given arrival rate. Other simulation software manufacturers offer much more customization options in such a "create" block. Even more complicated was the modeling of the setup process. Here several "Decide" blocks were needed: First, it was checked whether a conversion is necessary at all. If this was the case because the product type did not match, it was checked to see if the triggering object was product A and then whether main processing was equipped for product B or not for any product yet. Due to the different setup times such a cumbersome single consideration was necessary, while other software provides easy handling for such cases by means of tables.

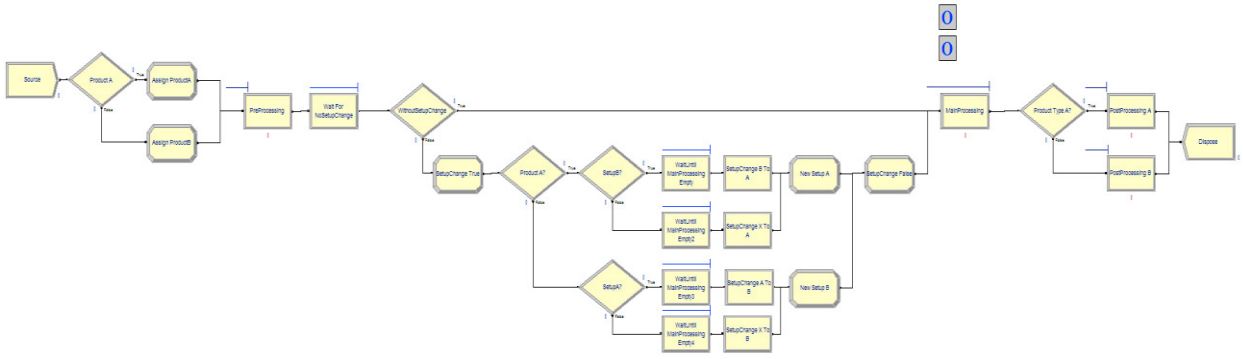

Fig. 2. Overview of the model structure in Arena.

Despite this partially cumbersome modeling, the reference situation could be modeled without encountering further difficulties. It should be noted that as the number of product types increases, the modeling effort would increase significantly.

4.2. Plant Simulation

We decided for Tecnomatix Plant Simulation from Siemens in version 14.0.6 as second C-DES tool. The software is widely known in Germany as standard simulation tool for analyzing production and logistics processes in automotive industry, mainly due to a modeling toolbox add-on provided by the German Association of Automotive Industry [3]. Similar to ARENA, models are created graphically via dragging-and-dropping components from the modeling libraries and connecting them to each other. Figure 3 presents the structure of the Plant Simulation model.

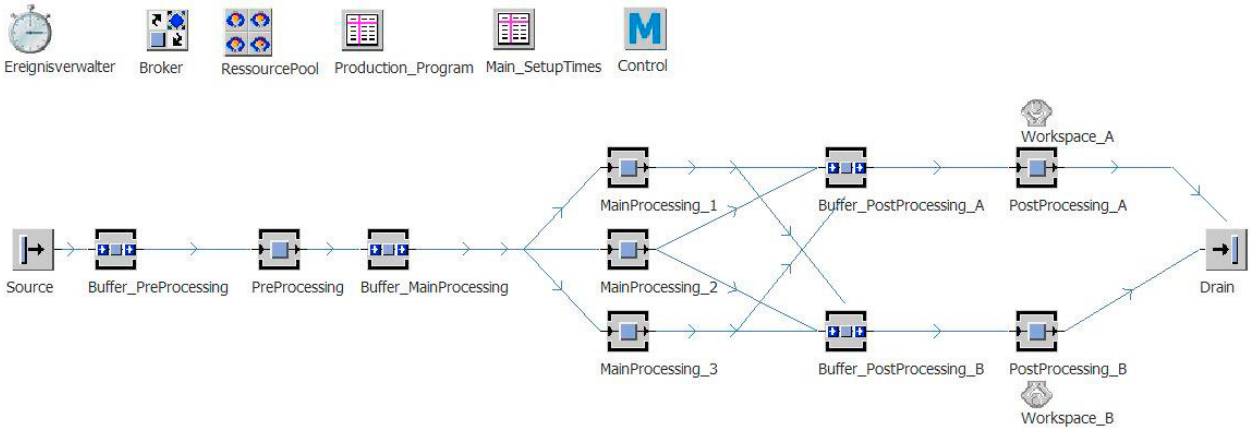

Fig. 3. Overview of the model structure in Technomatrix Plant Simulation

As discussed in section 3.1, the significant difference between the model structure in Plant Simulation and ARENA is that Plant Simulation strictly separates in material-flow related and non-material-flow related modeling components. Thus, the model structure in Plant Simulation is very close to the system layout of the problem case, while the structure in ARENA shows the entire logic of the modeled process. Plant Simulation provides extensive options for every model component to adjust its input, processing and output behavior. Furthermore, a user is able to develop custom process logic of any complexity by writing methods with the tool-integrated scripting language SimTalk. Especially due to the very extensive and detailed online help, but also due to the many example models and the large active user community, new users quickly learn how to create models in Plant Simulation. In Summary, we required 3 hours to

implement the reference problem case in Plant Simulation. Although the modeling was mostly pretty intuitive, the implementation of resources appeared slightly cumbersome and was the only time we were forced to use the online help. Furthermore, there was another obstacle during modeling concerning the implementation of the main processing stage. Since material flow library of Plant Simulation offers a single process and a parallel process, we assumed that it is convenient to consider the three main processing machines as one parallel process with a capacity equal to three. However, it was not obvious from the beginning that a parallel process can only consider one setup type at the same time. Consequently, all machines always had the same setup type at the same time. We then rebuilt the model and replaced the parallel process with three individual processes.

4.3. Salabim

We decided early to search for a python-based DES software package, since Python is considered as easy to learn, because of its clear structure and syntax. Furthermore, due to its popularity in data science and machine learning, the user community of Python is still rapidly increasing. SimPy is probably the oldest and best-known DES package for Python [4]. However, our first choice was initially the package ManPy [5; 6] - a wrapper for SimPy to facilitate the creation of manufacturing simulation models. It turned out that ManPy is not compatible with the latest Python version, because the software received its last update in 2016. We finally found Salabim, which met our requirements the most. Discovering Salabim was rather a coincidence than result of a systematic search as we found out about the software by an internet podcast [7]. Even thereafter, we could only investigate one scientific article about Salabim with a scope of one page [8]. Salabim is not based on SimPy and comes with its own event scheduler. The main benefit compared to SimPy is that Salabim supports some additional modeling concepts of the programming language SIMULA [9]. Thus, processes and entities can be activated, passivized and hold as preferred, which simplifies the implementation of processing logic.

Salabim 20.0.3 is compatible with the latest Python version 3.7. Furthermore, the latest version of the documentation is dated on August 08, 2020 [10]. Therefore, Salabim seems to be continuously updated, which makes its installation uncomplicated for users with first knowledge about Python. Since Salabim is a conventional library for Python, the creation of models is only possible by writing Python code. So far, Salabim provides no graphical user interface for modeling as it is common for most C-DES tools. However, the comprehensive documentation (211 pages) with many modeling examples allows the creation of complex custom models after very short time. The implementation of the reference problem case required around 6.5 hours after completing the installation of Salabim. Salabim provides some powerful evaluation features to verify and validate a simulation model and to conduct performance measurements. For instance, Salabim allows to trace every event in a python output console and to print statistics reports about the utilization of each queue. Furthermore, the built-in animation tools enable the real-time visualization of queue states while running a simulation.

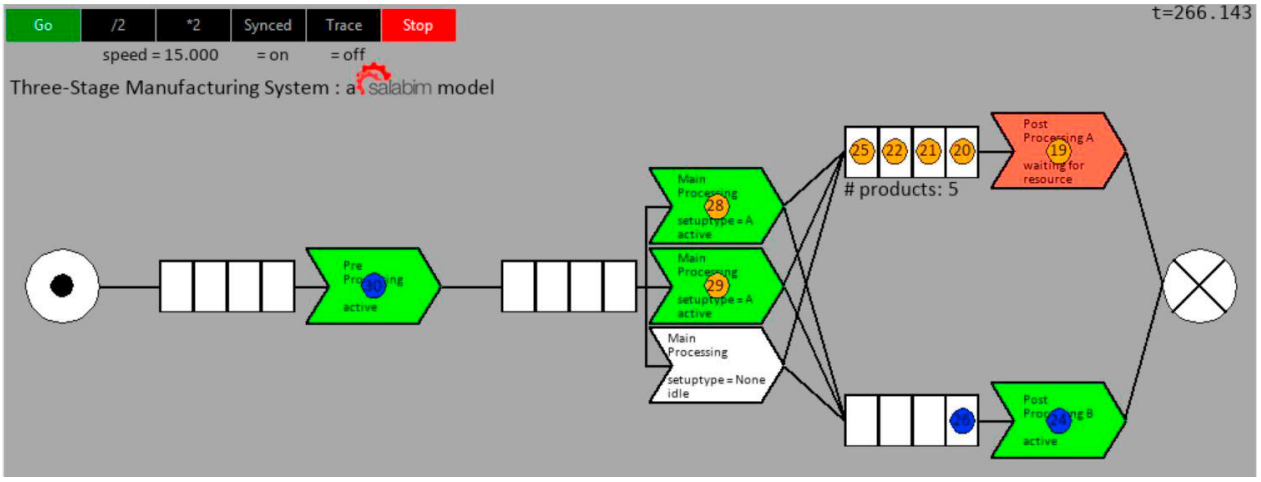

Fig. 4. Model animation in Salabim.

The animation capabilities of Salabim even allow more complex visualizations. Figure 4 shows a custom created animation, which represents the reference system as kind of process chain similar to figure 1. While running a simulation experiment, each process adapt its color and caption depending on its current state (idle, active, setting up, waiting for resource) and each queue always shows the first four entities stored. However, the creation of all animations from scratch required much more time than creating the model itself. It took almost a week until all animations worked properly. However, it seems likely that the effort for animating future models will be considerably lower, since many code fragments can be easily reused.

4.4. JaamSim

JaamSim is a free open source simulation software with a drag and drop interface. JaamSim requires Java 7 or higher and a graphics driver supporting OpenGL 3.0. JaamSim has been in continuous development since 2002. The current JaamSim Version is 2019-03. We used the JaamSim Version 2018-09, licensed under the Apache License, Version 2.0 [11]. JaamSim provides an extensive user manual which explains all the necessary concepts of JaamSim required to implement the reference task described in section 3.2 [12].

The program file has a size of only 8 MB. An installation is not necessary. JaamSim has a GUI which allows a comfortable modeling of the structure of a production and logistics systems. Figure 5 shows the structure of the reference model in JaamSim. Rules for the control of the production flows can be modelled with the help of attributes and expressions.

There is no object orientation like in Plant Simulation with inheritance of objects through the GUI. A JaamSim model is saved as a text file and can be audited and edited in the text file. In the model file, JaamSim allows for grouping of objects and assigning attributes to these groups.

In addition to the manual JaamSim has a good mouse over help. Furthermore, a JaamSim Users Discussion Group shows solutions to different kind of modelling problems. User question are answered immediately by the developers of JaamSim. Another very good source of material for modeling production and logistics systems with JaamSim is the “The Big Lean Simulation Library” with example models, videos and instructional documents for all models [13].

From the experience of the modeled reference case and the examples provided by the Big Lean Simulation Library the authors can state, that JaamSim is a good alternative companies to model production and logistics systems when they do not have access to commercial simulation packages. Descriptions JaamSim’s main functions can be found in [14; 15].

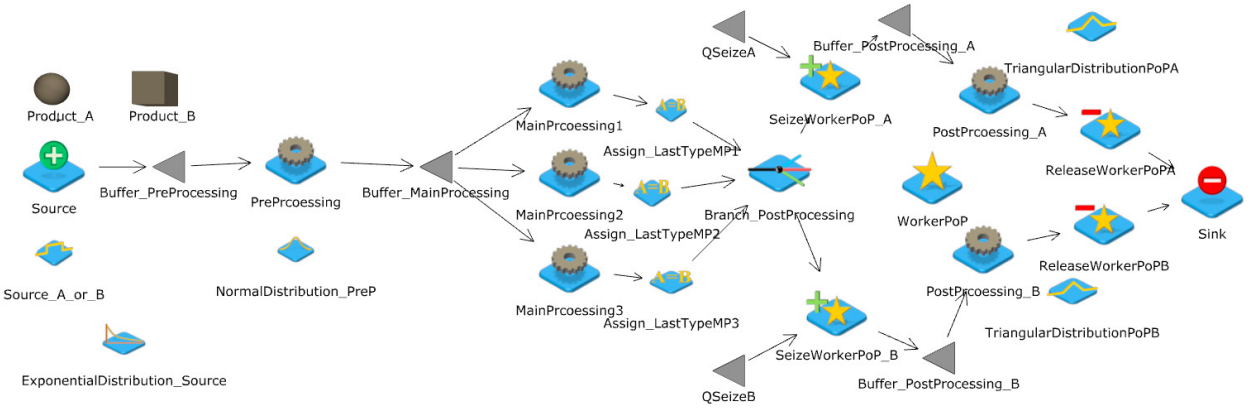

Fig. 5. Overview of the model structure in JaamSim.

4.5. CloudSim

Another OS simulator is the CloudSim toolkit [16], which is widely known in the computer science research community. CloudSim framework is built in Java using the discrete-event engine of SimJava simulator [17], which posed one of the main reasons to take a look into it since Java programming language is one of the most popular programming languages in the last two decades. The CloudSim framework is initially developed by the GRIDS laboratory of the University of Melbourne and was released in 2010. Thereafter, many extensions have been presented using the CloudSim as the core simulator, as for instance, CloudAnalyst. Most recently, a another extension named CloudSim Plus is presented by the Federal Institute of Education, Science and Technology of Tocantins [18]. Our initial analysis on google scholar is quit surprising, due to the tremendous number of found articles based on this simulator. The findings on the relevant research databases such as, SpringerLink, ScienceDirect and IEEE confirmed the ones done on google scholar. The number of found publications using "CloudSim" as a search string ranged between 400 in IEEE and 915 in SpringerLink. Those results motivated us to take a general insight into this simulation framework and found out that it has been widely adopted to address scheduling problems in the context of cloud computing. Scheduling is the deployment of resources in order to complete a set of tasks during a determined time span [19]. Scheduling has an essential role in different manufacturing as well as service sectors. In spite of the operative nature of scheduling tasks, they have a critical impact on most of the strategic decision-making processes [20]. Therefore, we modelled the presented reference problem to investigate whether this simulator can be adapted for normal industrial problems in addition to modeling cloud computing problems and infrastructure.

We relied on CloudSim Plus version to model the reference problem case. We were able to model the problem entirely. However, we faced some challenges in modelling the incoming jobs and their nature since jobs in a cloud computing context are usually processed on a single machine. On the contrast, the completion of jobs in a manufacturing environment usually requires several processing steps on different processing stages. One can extend the "Cloudlet" class of CloudSim to generate and model jobs with additional attributes that are required to be processed on multiple processing stages. The implementation time took roughly around 15 hours. The developer needs, however, a basic understanding of polymorphism in programming languages to be able to extend classes and leverage the capabilities of inheritance principles in object oriented programming. The CloudSim plus can be installed and integrated as a java project in Eclipse or Netbeans. In addition to the model examples provided in the CloudSim package, many substantial examples are provided in the CloudSim Plus. The examples are designed in manner to demonstrate the functionalities of the simulator and help the developer to grasp an insight on their usage. The documentation provided within the source code is comprehensive, in which the source classes and their methods are explained. In addition, a 604 pages substantial documentation was released for the CloudSim Plus that contains code documentation, frequently asked questions and answers and information about the relevant publications [21]. In this context, the development of is possible through writing the model in source code in Java. However, some limited functionalities of CloudSim can be leveraged using the CloudAnalyst package, which was developed on the top of CloudSim to provide a user interface [22]. CloudSim does not contain any animation packages. However, Java provides powerful toolkits for the development of user interfaces and animation such as Java Swing, which can be used to develop custom animation. The simulation toolkit provide a comprehensive library to collect and analyze simulation results

5. Conclusions

The authors could model the reference problem with all of the three used free and open source discrete-event simulation tools. Table 1 on the next page summarizes some key findings of the comparative study. JaamSim seems to be the best alternative among these three tools to commercial DES tools for a typical industrial engineer applying DES simulation to support planning tasks in production and logistics, since it comes with a GUI and with a lot of basic modeling objects that users already know from commercial tools. Advanced users can develop their own new simulation components for JaamSim as described in the Programming manual of JaamSim.

Salabim is also a very good tool for DES, which seems to comprise more modeling features than other text-based DES environments. Since Salabim do not provide a GUI for modeling, we believe that it is mostly suitable for

advanced users, who are proficient in structured and organized programming and process oriented thinking. However, due to the really good documentation with a variety of tutorials, modeling with Salabim appeared pretty intuitive for us and the implementation of the model was done in short time. Since Salabim is a library for python, it is easily possible to integrate any Salabim model in any python application and vice versa. Due to the huge variety of OS python libraries, for instance for machine learning, optimization, visualization and data exchange, we believe that an ambitious developer can create Salabim models, which are in any way inferior to C-DES models.

CloudSim probably requires the most advanced programming skills. Since it was designed to simulate cloud computing application, the tool had to be significantly expanded with custom functions to implement the reference model. Therefore, CloudSim is probably only interesting for IT specialists.

Table 1. Results of the comparative study.

	Arena	Plant Simulation	Salabim	JaamSim	CloudSim
Modeling					
Modeling Libraries	yes	yes	no	yes	yes
Attribution of Flow Objects	values and expressions	values and methods	values and methods	values and expressions	values and methods
Resources	yes	yes	yes	yes	yes
Control Logics	limited by modeling libraries	programming of custom methods	programming of custom methods	programming of custom methods	programming of custom methods
Usability					
Installation	easy	easy	advanced	easy	medium
GUI for Modeling	yes	yes	no	yes	no
Documentation	online help, books	online help, books	PDF slides	PDF slides	PDF slides
Modeling Tutorials	yes	yes	yes	yes	yes
Support	customer service, internet forum	customer service, internet forum	Google group, mailing list	Google group, mailing list	Google group, mailing list
Evaluation					
Model Animation	medium	professional	basic	advanced	non-existent
Input Data Analysis	Features to search for the best fitting probability distribution that describe a set of input data		variety of OS libraries	no	variety of OS libraries
Simulation Results	extensive statistical reports, diagrams	extensive statistical reports, diagrams	trace file, queue statistics and diagrams	customized reports, diagrams	trace file, queue and utilization statistics
Debugging	limited, mainly based on model animation	extensive debugging features	strongly depends on external libraries and coding editor	property viewer to assess model states	very good, due to the strong capabilities of Java for debugging
Additional Features					
Optimization	OptQuest engine	genetic algorithm library	variety of OS libraries	no, but planned for future releases	variety of OS libraries
Data Interfaces	limited	extensive	variety of OS libraries	limited	variety of OS libraries

References

[1] Dias, L. M. S.; Vieira, A. A. C.; Pereira, G. A. B.; Oliveira, J. A. Discrete simulation software ranking — A top list of the worldwide most popular and used tools. In Simulating complex service systems. Proceedings of the 2016 Winter Simulation Conference, Washington, DC, USA, 11-14 December; Roeder, T. M., Frazier, P. I., Szechtman, R., Zhou, E., Eds.; IEEE: Piscataway, NJ, 2016; pp. 1060–1071.
[2] Dagkakis, G.; Heavey, C. A review of open source discrete event simulation software for operations research. Journal of Simulation 10, 3; pp. 193–206.
[3] Mayer, G.; Pöge, C. The Road to Standardisation – from the Idea to the Realisation of the VDA Automotive Toolkit. In Integrationsaspekte der Simulation: Technik, Organisation und Personal; Zülch, G., Stock, P., Eds.; KIT Scientific Publishing: Karlsruhe, Germany, 2010; pp. 29–36.
[4] Introduction to Discrete-Event Simulation and the SimPy Language. Available online: http://heather.cs.ucdavis.edu/~matloff/156/PLN/DESimIntro.pdf (accessed on 29.03.2019).
[5] Dagkakis, G.; Heavey, C.; Robin, S.; Perrin, J. ManPy. An Open-Source Layer of DES Manufacturing Objects Implemented in SimPy. In Proceedings of the 8th EUROSIM Congress on Modelling and Simulation, Cardiff, UK, 10-13 September; Al-Begain, K., Ed.; IEEE: Piscataway, NJ, 2013; pp. 357–363.
[6] Olaitan, O.; Geraghty, J.; Young, P.; Dagkakis, G.; Heavey, C.; Bayer, M. et al. Implementing ManPy, a Semantic-free Open-source Discrete Event Simulation Package, in a Job Shop. Procedia CIRP 25; pp. 253–260.
[7] Salabim: Logistics Simulation with Ruud van der Ham. Episode 151. Available online: https://www.podcastinit.com/salabim-with-ruud-van-der-ham-episode-151/ (accessed on 01.09.2020).
[8] van der Ham, R. salabim: Discrete event simulation and animation in Python. JOSS 3, 27; pp. 767–768.
[9] Dahl, O.-J.; Nygaard, K. SIMULA: an ALGOL-based simulation language. Commun. ACM 9, 9; pp. 671–678.
[10] salabim Documentation. Release 20.0.3. Available online: https://rawgit.com/salabim/salabim/master/salabim.pdf (accessed on 01.09.2020).
[11] JaamSim: Discrete-Event Simulation Software. Version 2018-09. Available online: https://jaamsim.com (accessed on 13.10.2020).
[12] JaamSim User Manual. Software Version: 2020-12. Available online: https://jaamsim.com/docs/JaamSim%20User%20Manual%202020-12.pdf (accessed on 13.10.2020).
[13] The Big Lean Simulation Library. Available online: https://thebigleansimulationlibrary.wordpress.com/ (accessed on 09.09.2020).
[14] King, D. H.; Harrison, H. S. JaamSim" Open-source Simulation Software. In Proceedings of the 2013 Grand Challenges on Modeling and Simulation Conference (GCMS 2013), Toronto, ON, CA, 7-10 July; Vakilzadian, H., Huntsinger, R., Crosbie, R., Cooper, K., Eds.; Society for Modeling & Simulation International: Vista, CA, 2013; 1:1-1:6.
[15] King, D. H.; Harrison, H. S. Open-source Simulation Software "JaamSim. In Simulation: Making Decisions in a Complex World. Proceedings of the 2013 Winter Simulation Conference, Washington D.C., USA, 8-11 December; Pasupathy, R., Kim, S.-H., Tolk, A., Hill, R., Kuhl, M. E., Eds.; IEEE: Piscataway, NJ, USA, 2013; pp. 2163–2171.
[16] Calheiros, R. N.; Ranjan, R.; Beloglazov, A.; Rose, C. A. F. de; Buyya, R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw: Pract. Exper. 41, 1; pp. 23–50.
[17] Howell, F.; McNab, R. simjava: A Discrete Event Simulation Library For Java. In Simulation and Modeling Technology for the Twenty-First Century. Proceedings of the 1998 International Conference on Web-Based Modeling & Simulation, San Diego, CA, USA, 11-14 January; Smith, R., Hill, D. R., Fishwick, P. A., Eds.; Society for Computer Simulation International: San Diego, CA, USA, 1998; pp. 51–56.
[18] Filho, M. C. S.; Oliveira, R. L.; Monteiro, C. C.; Inacio, P. R. M.; Freire, M. M. CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In Proceedings of the 2017 IEEE International Symposium on Integrated Network Management. May 8-12, 2017, Lisbon, Portugal, Lisbon, Portugal, 8-12 May; Chemouil, P., Ed.; IEEE: Piscataway, NJ, 2017; pp. 400–406.
[19] Baker, K. R.; Trietsch, D. Principles of sequencing and scheduling; John Wiley: Hoboken, NJ, USA, 2009.
[20] Pinedo, M. L. Scheduling. Theory, Algorithms, and Systems. 4th ed.; Springer US: Boston, MA, 2012.
[21] CloudSim Plus Documentation. Release 1.0.0. Available online: https://buildmedia.readthedocs.org/media/pdf/cloudsimplus/latest/cloudsimplus.pdf (accessed on 08.04.2019).
[22] Wickremasinghe, B.; Calheiros, R. N.; Buyya, R. CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications. In Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, Australia, 20-23 April; Chang, E., Ed.; IEEE: Piscataway, NJ, 2010; pp. 446–452.