Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

LEARNING ENVIRONMENT FOR INTRODUCTION IN DISCRETE-EVENT SIMULATION FOR DESIGN AND IMPROVEMENT OF NEW AND EXISTING MATERIAL FLOW SYSTEMS

Bastian C. Schumacher
Holger Kohl

Technische Universität Berlin
Chair of Sustainable Corporate Development
Straße des 17. Juni 135
Berlin, 10623, GERMANY

ABSTRACT

In this paper, ways are shown how students can be familiarized with executing simulation studies for the design and improvement of new and existing material flow systems using flexible discrete-event simulation (DES) tools. The prototypical app “Production Simulation Application” is described. It combines learning-conducive components that are used to familiarize users with objects, the graphical model buildup, and the use of programming language. Game elements such as levels, badges, and points are shaped to motivate learners to interact frequently. They enable immediate feedback. A test shows that the app has been used repeatedly at short intervals beyond the course. A procedure for experience-based learning for conducting simulation studies is developed, in which a so-called learning factory enables learners to complete a simulation study. It is shown that the developments can contribute to the dissemination of DES and to increasing the planning quality in times of rising complexity of production systems.

1 INTRODUCTION

Simulation has the potential to support people in decision-making, by analyzing more alternatives in a shorter time. The use of discrete-event simulation (DES) in a planning project for material flow systems can lead to savings through the early identification of errors (Gutenschwager et al. 2017). Consequently, these errors do not have to be corrected in later phases of the project.

DES can be used for decision support for the design and improvement of new and existing material flow systems. The complexity of material flow systems results from branching and reverse material flows (Lödding 2016). Product variants, dynamic behavior, and stochastic influences increase this complexity. Material flow systems are complex systems (ElMaraghy et al. 2012). DES has been proven to be a suitable tool for decision support in such complex environments (Law 2015). Five levels of planning of material flow systems in decreasing aggregation can be distinguished: Production network, plant, building, segment, and workplace (VDI 2008). DES tools should be flexible enough to be used for more than one level and should be able to model new paradigms such as cyber-physical systems, e.g. as multi-agent systems.

“Raising awareness of models’ potential among the wider engineering community and [to] equip engineers with methods and tools for using appropriate models to depict real-world systems in the virtual world” is a great challenge (Kagermann et al. 2013). “Modeling and simulation can only be carried out by qualified experts” (Kagermann et al. 2013). Complex and comprehensive simulation tools hinder a broad use of the software (Byrne et al. 2015; Padilla et al. 2014). Conducting simulation studies is often considered to be complex and requires experience (Banks et al. 2010; Shannon 1998).

Mobile and game-based approaches as well as learning factories can improve learning and teaching productivity for DES. Thus, they have the potential to reduce the before mentioned barriers. These approaches can be a suitable means of instruction, especially for young learners.

This paper is intended to contribute to the dissemination of DES as a decision support tool for the design and improvement of new and existing material flow systems. The aim is to improve the education and training of engineers in their ability to plan material flow systems. A learning environment is designed and prototypically implemented to enable beginners to become acquainted with DES tools for conducting simulation studies.

Chapter 2 documents the state of the art for conducting simulation studies using DES tools in the application area of material flow systems. In chapter 3 the learning environment is designed. The concrete implementation and a test are described in chapter 4. Chapter 5 contains a summary of this paper.

2 STATE OF THE ART

The following sections document the state of the art for planning material flow systems using DES and a discussion of the necessity of qualified simulation users for different DES tools. Procedures for conducting simulation studies, potentials for increasing learning and teaching productivity for DES such as gamification, learning factories, and visual block-based programming language and related work form the knowledge base for the development of the learning environment.

2.1 Qualification of Simulation Users

The required knowledge of users can be considered in relation to DES tools available on the market. DES tools can be classified according to their application relevance (Noche et al. 1993; Noche and Wenzel 1991; Schmidt 1988; Wenzel 2008).

- Programming languages with basic aspects in simulation, e.g. class concepts or event management.
- General DES tools: Like the previous one, extended by components specific to model classes.
- DES instruments: Like the previous one, with specialized components in terms of specific application areas, e.g. production and logistics, computer systems and networks (Banks et al. 2005).
- Specific DES instruments with components for modeling and simulation of aspects from a part of an application area. Example: Instruments for the planning of automated guided vehicles.

With an increasing level, the DES tools gain in application relevance and lose in flexibility and universality. This is contrasted with the required qualification of the user. With the increasing universality of the DES tools, specific knowledge of simulation technology and IT is required, and vice versa.

The DES instruments focused on in this paper are a compromise between flexibility on the one hand and application relevance and technical language on the other hand. The DES instruments allow the flexible modeling and simulation of different aspects of material flow systems and the evaluation of concepts such as multi-agent systems. They are suitable for the simulation of different planning levels of material flow systems.

Most DES instruments support a graphical model buildup and model building using a programming language or provide access to so-called programmed modules (Swain 2015, 2017). In summary, it can be stated that new DES users need to be familiarized with the graphical model buildup using objects. The use of a programming language is essential for the flexible implementation of simulation models (Schumacher 2020).

2.2 Simulation Studies

A simulation study is a project for the simulation-based investigation of a system (VDI 2013). It consists mainly of the preparation, execution, and evaluation phase (Kühn 2006; Law 2015; VDI 2014). During the first phase, the problem and objective of the investigation are specified and relevant data are collected. In the execution phase, an initial simulation model must be created, verified, and validated. Verification is a process in which the correctness of programs or program parts is formally proven (VDI 2013). It can be carried out by syntax checks or consistency checks (VDI 2013). Validation targets to analyze the suitability of a model concerning a given task and the sufficiently accurate modeling of the system under consideration (Rabe et al. 2008). After the simulation experiments have been carried out according to an experiment plan, the results of the experiments must be analyzed and the changes to the material flow system must be made in the evaluation phase.

Especially abstraction by a reduction in the sense of omitting the representation of details or idealization in the sense of simplifying real conditions as well as the interpretation of simulation results requires experience (Banks et al. 2010; Shannon 1998; VDI 2013).

In the literature, a big number of procedures for conducting simulation studies can be found. In the following, Kühn's so-called simulation cycle is used as a basis (Kühn 2006). This cycle can be represented graphically in a compact way. It is visible in the inner part of Figure 2. A digital simulation model is abstracted from a real material flow system. With this simulation model, experiments are conducted based on a previously defined plan. In the next step, the results of the experiments are analyzed and interpreted by the user considering the objective of the investigation. Since the digital model represents an abstraction of reality, it is necessary to contextualize the simulation result data on the material flow system and thus derive improvements. This means that the user checks whether abstracted aspects have an effect on a possible implementation in reality. The improvements are finally implemented in reality. The cycle is repeated.

2.3 Increasing Learning and Teaching Productivity

Gamification is the "use of game design elements in non-game contexts" (Deterding et al. 2011). Such game design elements, in the short term referred to as game elements, can for example be levels, points, or badges. Levels represent short closed learning units. A level comprises a problem to be solved by the learner. Points document the progress of an individual gamer for a group of gamers. Badges make the gamer's achievements visible for the gamer and others. In the case of obvious badges, the criteria for awarding them are known to the gamer. In contrast to this are the non-obvious badges. Manrique et al. (2015) recommend the following steps in the game development process. First, the team of game developers should be formed. They formulate the problem and/or the objective. The why what and who is visualized: They analyze the target group and define how the gamified experience should influence them. The theme and a story are selected and game elements such as levels, or badges, and game mechanics are chosen and designed. After the graphics are improved, repeated game tests take place. Games can positively influence the teaching and learning productivity of students through motivation (Müller et al. 2015). In most cases, the literature confirms the learning benefit. However, it can be assumed that negative results may be published less frequently (Despeisse 2018). After an analysis of several game examples, Despeisse (2018) concludes that digital games are well suited to develop technical skills and achieve cognitive learning outcomes. Non-digital methods and collaborative game mechanics are more aimed at developing soft skills. Consequently, it seems attractive to combine both digital and non-digital game approaches. Learning factories are a promising non-digital approach.

The definition of the term learning factory includes a learning environment specified by authentic processes, multiple stations, and technical aspects as well as organizational aspects, a setting that is changeable and resembles a real value chain, a physical product being manufactured, and a didactical concept with formal, informal, and non-formal learning, enabled by trainees' actions (Abele 2014; Abele et al. 2015). Learning factories are a place where engineers learn experience-based (Abele et al. 2017; Kolb et al. 1971; Müller et al. 2017a). Through active experimentation, learners gain a concrete experience (Kolb et al. 1971). This experience is followed by observation and reflection, which leads to the formation of an abstract concept, which in turn leads to an experience in future experiments. Learning factories are particularly suitable for the training of engineers. Engineers spend their time in a familiar environment and can experiment without the risk of safety or damage (Müller et al. 2017a).

Visual block-based programming has the potential to introduce beginners in programming by combining blocks by Drag and Drop to get executable programs. Blocks are stored for the learner in a block library. The blocks are instantiated by the user in the block editor. Blocks represent program elements such as functions or variables (Kurihara et al. 2015). These blocks have special visual characteristics. The user knows intuitively through colors and shapes where a particular block can be placed. Usually, existing visual block-based programming languages include blocks that can represent conditions, loops, the declaration of variables, and pre-defined functions. An example of a visual block-based programming tool is Scratch, a project of the Lifelong Kindergarten Group at the Media Lab of the Massachusetts Institute of Technology (Scratch 2020). It is a visual block-based programming language designed especially for eight to 16 year-olds. Blocks can be connected by Drag and Drop to create simple games or animations. A similar example of a visual block-based programming tool is Google Blockly (2020). In summary, visual block-based programming languages help to introduce beginners to programming languages. It is possible for learners with little previous knowledge to quickly build correct programs or program parts. Visual block-based programming languages appear attractive to introduce beginners to programming in DES instruments.

2.4 Related Work

Padilla et al. (2014, 2015 and 2016) describe the use of simulation games for teaching and learning DES as part of the tool ClouDES. The authors propose to use simulation games as an entertaining learning activity first. In this activity, students shall learn so-called DES concepts through reduced technical language in the form of objects, to subsequently switch to a more serious game mode. The authors have named the two games Medieval Wars and Dystopian City, which are implemented in the DES tool ClouDES (Padilla et al. 2015). The goal of both game approaches is to build simulation models by instantiating prepared objects according to a game description. Concerning the goal of the respective game, users must select and parameterize the correct objects.

Currently, training in the context of DES is often based on textual tasks. Collective works are available in bookshops, for example Bangsow (2010, 2015), Banks et al. (2010), or Kelton et al. (2010, 2015). Hypothetical cases and problems are described textually, which provide the learner a basis for conducting simulation studies for problem-solving. Usually, the publications contain a step-by-step tutorial with an exemplary solution of the task in a specific DES instrument. In most cases, the learners complete the tasks in the books “linearly”. The books usually refer to a specific DES instrument.

New approaches that focus on building up experience in conducting simulation studies could not be identified in the literature. There is little research available where game elements are used for introduction into flexible DES instruments. No approaches focus in this context on the design and improvement of new and existing material flow systems. The existing approaches hardly concentrate on introducing into a graphical model buildup using objects and programming language for DES instruments. The abstraction and interpretation of results are hardly focused. Little importance is attached to high teaching and learning productivity and a motivating design of a learning environment.

3 LEARNING ENVIRONMENT

In the following sections, the learning environment for DES is designed based on the findings from the previous chapters. Learning conducive elements and a procedure for experience-based learning to conduct simulation studies (ELSS) are structured in a game- and learning path (Schumacher 2020).

3.1 Learning Conducive Elements

Incrementally more Complicated Simulation models (ICS) and Visual Block-based Programming language for the Implementation of Simulation models (VBPIS) are designed as part of a mobile app.

3.1.1 Incrementally More Complicated Simulation Models

ICS are used as a step-by-step introduction to graphical model buildup using objects. Considering the level of difficulty of the task and the abilities of the individual gamer, simulation models are to be created that include a small number of previously unknown objects (Müller et al. 2017b). These consist, for example, of a source, a drain, and a workstation. Level-based, the difficulty is increased by introducing new objects, e.g. an assembly station. For each level, one task must be solved by the learner.

The ICS are structured in levels. Each level contains a task to be solved by the learner. Learners build a simulation model at each level. In this way, they solve a task assigned to them. Similar levels are grouped in levelpacks. These are mainly formed according to aspects of material flow systems, like personnel, information, production equipment, material flow equipment, information flow equipment, energy, and infrastructure (Schmidt and Schneider 2008).

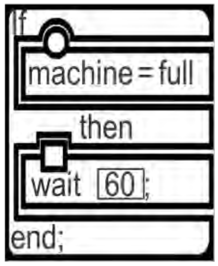
- Levelpack 1 Material Flow: Introduction to objects to model the material flow systems. Learners are enabled to create simulation models with pre-configured objects.
- Levelpack 2 Information Flow: The focus is on mapping the information flow and information flow objects. The learner is trained in programming using the VBPIS.
- Levelpack 3 Personnel and Energy: Similar to levelpack 1 and 2, the levelpack 3 is aimed at mapping employees and energy.
- Levelpack 4 User-defined objects: The focus is on using user-defined objects. Example: Modeling of multi-agent systems.
- Levelpack 5 Story mode: Performing sensitivity analyses. The possible changes in the material flow are limited by a story that is told to the learner.
- Levelpack 6 Level design: Users can implement task descriptions in the app and make them available to others for solving.

The learning environment invites the learner to use it occasionally and frequently. Game elements and mechanics are designed to reward users for their activity and success. A point is awarded upon successful completion of a level. Points in turn increase the gamer's rank in the leaderboard. Other gamers can view this leaderboard. By reaching a certain number of points, the gamer is awarded a badge. In addition to awarding points, badges should motivate gamers to work towards goals. These goals result from the criteria known to the gamer. Badges are in the simplest case a graphic that is visible to the gamer or the username of the gamer that is displayed publicly. For example, a badge can be awarded when levelpack 1 has been completed. Gamers always know the steps they need to take to advance in the game. They should also be able to see which badge has already been given to another gamer. This way the own progress can be evaluated in comparison to the others. Striving for progress should be generated.

3.1.2 Visual Block-based Programming Language for the Implementation of Simulation models

The introduction into the specific programming language for the implementation of simulation models which is supported by the DES instrument is part of the learning process. To solve more difficult tasks, learners must formalize programs with such a language. The VBPIS combines the graphical properties of visual block-based programming languages with functions of object-oriented programming languages for the implementation of simulation models. Programming languages for the implementation of simulation models support typical functions in DES instruments such as event handling or data collection (VDI 2014). Usually, these languages are used if individual processes are to be represented in a simulation model that can hardly be described by pre-configured objects.

The VBPIS is inspired by Scratch in terms of its visual and block-based features (Scratch 2020). The goal is to familiarize the future simulation user with programming considering the syntax of a programming language for implementing simulation models. Figure 1 shows conceptually the VBPIS on the left, consisting of a block “If” with blocks that have been inserted such as the condition and a “wait” function (Schumacher 2020). On the right side of Figure 1, the code has been translated into an exemplary programming language for the implementation of simulation models. A compiler is required to translate the program developed by the learner in the block editor of an app into the programming language for the implementation of simulation models, which is supported by the DES instrument.




```
if machine = full

then

wait 60;

end;
```

Figure 1: Example for the VBPIS (left), translation in exemplary programming language for the implementation of simulation models (right).

Compared to Scratch or Google Blockly, VBPIS has the characteristics of object-oriented programming languages for the implementation of simulation models. Blocks are used to define conditions, loops, variable declarations, object paths and operators. Simulation-specific characteristics are added: Objects, both temporary and permanent, stationary and mobile, must be addressable by the programmer. Manipulating attributes of objects is an important prerequisite for changing the behavior of objects by program code. The execution of preconfigured procedures such as the transfer of a temporary object must also be supported.

3.2 Procedure for Experience-based Learning to conduct Simulation Studies

The procedure consists of Kühn's simulation cycle in the center, surrounded by Kolb's experiential learning cycle, see Figure 2 (Kolb et al. 1971; Kühn 2006; Müller et al. 2017a). It shows the procedure for ELSS. The procedure is to be applied to a learning factory. Simulation models are abstracted digital representations of the learning factory's information and material flows, including resources. Outcomes refer to the results of the simulation experiments. Improvements are outcomes, interpreted by the user concerning the applicability in the learning factory. In section 4.2 a concrete application case is illustrated.

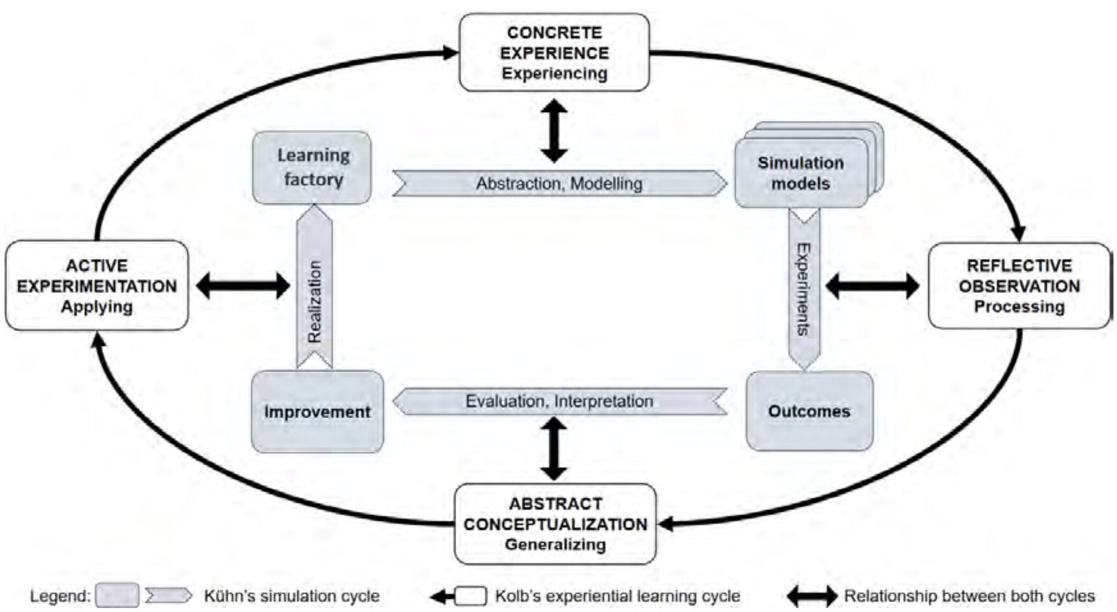


Figure 2: Procedure for experience-based learning to conduct simulation studies.

3.3 Game- and learning path

The ICS, VBPIS, and the procedure for ELSS are structured as a game- and learning path in Figure 3 (Schumacher 2020). This path represents the learning environment for training in DES in the application area of material flow systems. The learning environment is structured according to the phases Onboarding, Midgame, and Endgame (Manrique et al. 2015). After completing the game, learners should be able to set up simulation models in a DES instrument and conduct simulation studies. They should be able to explore further objects in the DES instrument independently.

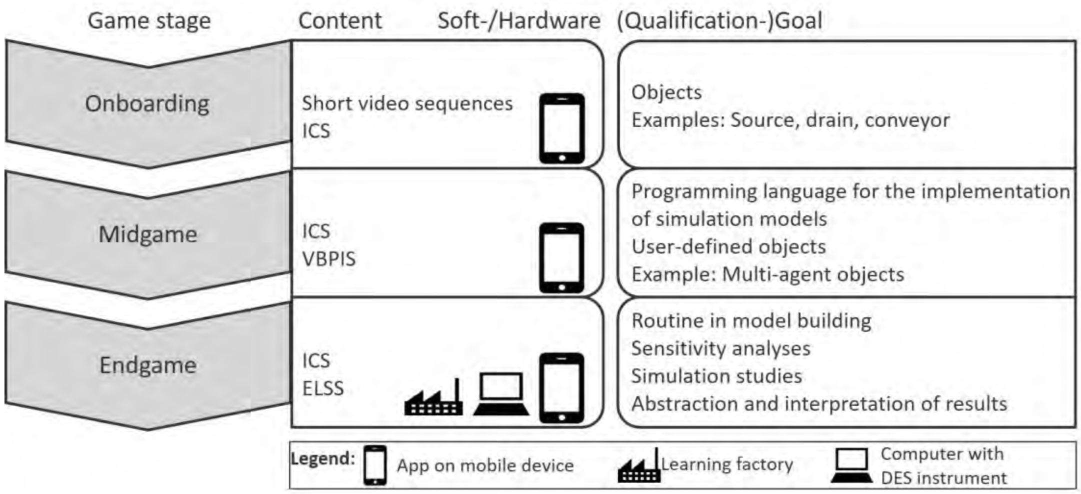


Figure 3: Game- and learning path.

The game is intended to introduce in modeling simple and complicated material flow systems. A game environment, in which the complexity of DES instruments is simplified, allows a quick introduction of new users into the application of DES instruments. Complex simulation models are created in the DES instrument after the user has applied the learning environment.

The goal of onboarding is to enable users to easily enter the learning environment. This includes attracting the user's initial attention. Furthermore, they are introduced to the operation of the app. The important objects of the DES instrument are introduced in the form of videos.

The essential part of gaming and learning takes place in the midgame. To be able to continue collecting points and unlock badges, users must implement individual procedures using a programming language. For this, they use VBPIS. First, the learners replicate the code shown in the task description, then they create it based on pseudocode and finally on a textual description. In the next step, the learners can design tasks for other learners to further intensify the course and strengthen group interaction. The animation of the models built by users in the sense of a stereoscopic/immersive projection serves in this phase of the game to increase motivation and support validation/verification.

In the end game phase, the learner uses user-defined objects in the app or DES instrument. In practice, such objects are often grouped in libraries and refer to a concrete application area. Examples are the object library of the German Association of the Automotive Industry or objects for the simulation of multi-agent systems (Mayer et al. 2010). In the final game, the learners are practically trained in the execution of simulation studies using the procedure for ELSS.

4 IMPLEMENTATION

The concept was realized with the DES instrument Tecnomatix Plant Simulation. In the following section 4.1, the exemplary implementation of the app Production Simulation Application (PSIMA) including the

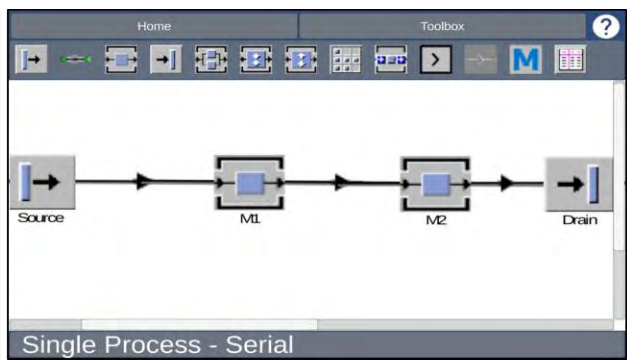
ICS and VBPIS is described. The application of the procedure for ELSS is presented in section 4.2. The conclusion is documented in section 4.3.

4.1 Mobile and Gamified App Production Simulation Application

A Cloud-IT-architecture was created and the app PSIMA including ICS and VBPIS was prototypically implemented using the widely used DES instrument Tecnomatix Plant Simulation as an example (Müller et al. 2017b). The cloud-IT-architecture consists of PSIMA, a web server, a database, and the DES instrument. The learners are building simulation models with PSIMA in a game-based way and provide input and experiment data. The modeling environment of the app is graphically similar to the DES instrument. Attributes of objects such as type, position in x, and y-direction and processing time are transferred to a local database located on the mobile device. Attributes of the simulation model, such as the simulation period, are also treated in the same way. The local database is synchronized in a defined time interval with a globally accessible database via the internet using scripts stored on a web server. An algorithm in the DES instrument automatically generates a simulation model based on the data stored in the database and starts a simulation run. At the end simulation results are transferred to the database. Learners can access this data to evaluate it in PSIMA.

This cloud-IT-architecture enables realizing a cost-effective mobile learning environment, that is scalable, easy to maintain, and develop. Challenges are the implementation of animation and mobile data use. Figure 4a shows an example of the description of level 1-2. The goal of level 1-2 is that the learner connects several objects sequentially and knows that these can be renamed. Figure 4b shows for this case the modeling environment in which the learner already instantiated the required objects by Drag and Drop. In this modeling environment, learners build material flow systems.

a) Description of level 2 in levelpack 1



b) Exemplary sim. model in modeling environment

c) VBPIS exemplary program

d) Hall of Fame showing rewards

Figure 4: Exemplary screenshots of the prototypical app PSIMA.

The VBPIS has been implemented using the language SimTalk of Tecnomatix Plant Simulation. The most important structures like if-then functions or loops are implemented in the app PSIMA. Figure 4c shows the block library at the top and the block editor in the middle. In the block editor, an exemplary program was created in the new VBPIS. A compiler is required to translate the VBPIS used in PSIMA into SimTalk. Finally, Figure 4d shows in the background the badges acquired by the learner. In the front details about the badge being touched are shown. In the illustrated case the learner has earned the badge "Persistent" by opening the app ten times. Exemplarily, the animation has been implemented in a virtual reality (VR) environment.

In a 19-day experiment, PSIMA was tested by students regarding motivation, usage behavior, and learning success (Müller et al. 2017b). The feeling of motivation was confirmed by two-thirds of the test persons. After using PSIMA, one respondent was able to build simulation models directly in the DES instrument and simulate them with little help. Overall, 377 models were submitted by 19 active gamers. The term active gamers refer to students who submitted at least one model. This high number of submitted models shows the high motivation to use PSIMA and to achieve many points. The evaluation of the usage data shows that PSIMA was mainly used for the preparation and follow-up of an accompanying lecture on DES. The data indicate that PSIMA was frequently used at short intervals. In the survey, the students stated that they used the app mainly at home. The second-highest number of responses was while traveling on public transportation.

4.2 Learning factory

In the following, the testing of the ELSS procedure in learning factories in terms of the teaching is described. The testing took place in the course Simulation of Production Systems in the production engineering laboratory of the Vietnamese-German University (VGU). Five variants of so-called trolleys are assembled on seven workstations (OPs). The OPs are arranged in a U-layout and linked by roller conveyors. The trolleys are assembled in a One-Piece Flow (Müller et al. 2017a).

A bottleneck in the learning factory should be identified and expanded by the students using DES. The ELSS procedure was tested at VGU with 30 students, divided into five groups. The learning unit lasted five days and was conducted as the final activity of a course on DES. The students were already familiarized with the DES instrument Tecnomatix Plant Simulation. Table 1 shows the workflow.

Table 1: Workflow for the test of the EEDS procedure in the learning factory.

Day	Workflow
1	Randomly selected students operate the learning factory for 45 minutes. All other students are responsible for data collection by measuring times and by observation.
2	Students are not allowed to operate the learning factory physically. Students conduct simulation experiments for increasing profitability.
3	Same workflow as on day two.
4	Each group of students has one hour in the learning factory: 30 minutes for the reconstruction of the factory according to the improvements identified on days two and three and 30 minutes for the operation of the improved learning factory.
5	Presentation of the results about the simulated improvements, their implementation in the learning factory, and a comparison between the simulated expected result and the result determined from the operation of the learning factory.

Four of the five groups of students were not able to satisfy the customer demand as expected through simulation after improving the physical learning factory. The four groups found that the workers were producing defective products or assembling them more slowly than expected. Also, one group reported insufficient communication with the workers. Another reported problem was that the batteries of electric screwdrivers were running low and had to be replaced during operation.

Only one of the five groups of students achieved the target values that they expected based on their simulation study. This indicates that most students made wrong or too general assumptions in the modeling phase concerning the relevant parameters that influence the system behavior. Examples are the error rate of workers, and rejects = 0%, no communication problems, and availability of tools = 100%. As a result, students noticed during the operation of the improved learning factory that they had abstracted reality too much. Students were able to gain an awareness of the consequences of insufficient abstraction of reality and interpretation of the simulation result data. Students voluntarily determined further improvements by DES after the course. This suggests a high level of motivation.

5 CONCLUSION

A learning environment to introduce in a DES instrument for conducting simulation studies was developed. The learning environment contributes to the dissemination of DES as a decision support tool for the design and improvement of new and existing material flow systems.

Game-based approaches could be identified as promising due to their learning conducive and motivational effects. Games increase the learning and teaching productivity, student motivation, positively influence group work, and allow experimentation in a safe environment. Mobile devices can be used independently of time and place. DES instruments that support objects and a graphical model buildup, as well as a programming language for the implementation of simulation models, have been identified as common in the market for the design and improvement of new and existing material flow systems. So far, there were hardly suitable approaches to introduce in the use of DES instruments.

The learning environment is described in the sense of a game and learning process in the paper. The learner is introduced to the graphical model buildup by using objects and a programming language for the implementation of simulation models using a game-based app. Incrementally more Complicated Simulation models (ICS) and Visual Block-based Programming language for the Implementation of Simulation models (VBPIS) were designed. The app Production Simulation Application (PSIMA), comprising ICS and VBPIS was prototypically implemented. PSIMA was tested with students. They used the app frequently at short intervals during an accompanying lecture on DES.

To make students aware of the importance of abstraction and interpretation of simulation results, the procedure for Experience-based Learning to conduct Simulation Studies (ELSS) was developed. It allows students to train to conduct a simulation study in a safe environment. In a learning factory, the ELSS procedure was tested. The students were able to experience that careful preparation, execution, and interpretation of simulation studies are prerequisites for satisfactory results.

The learning environment contributes to reducing the complexity of DES instruments through step-by-step training of the user, thus making the instruments easier and quicker accessible to people. A one to two weeks training, as offered by consulting companies, for example, is obsolete. The risk of a bad investment is reduced because a costly DES instrument does not have to be procured to find out whether the question of the project can be answered at all. The game-based and motivating aspects of the learning environment have the potential to use working time effectively and efficiently. Engineers build up experiences in abstraction and interpretation of simulation results in an environment they are familiar with.

Through the accompanying training, there is a chance to shorten the duration of the planning project. The “up-to-date” learning environment has a motivating effect on the planner of material flow systems, who regards the use of DES in the project as useful and meaningful rather than as an additional stressful task. Stochastic and dynamic effects can become more easily assessable for material flow systems planners. Thus, time and financial advantages can be exploited in the design and improvement of new and existing material flow systems.

A virtual reality environment will be developed as a follow-up to the work, which will enable learners to more easily analyze the simulated material flow.

REFERENCES

Abele, E. 2014. "Learning Factory - Living reference work entry - 07.03.2016". In *CIRP Encyclopedia of Production Engineering.* edited by G. Reinhart, and L. Laperrière, Springer: Heidelberg, New York, Dordrecht, London, 1–5, http://link.springer.com/referenceworkentry/10.1007/978-3-642-35950-7_16828-1, accessed 20th April 2020.

Abele, E., J. Metternich, M. Tisch, G. Chryssolouris, W. Sihn, H. ElMaraghy, V. Hummel, and F. Ranz. 2015. "Learning Factories for Research, Education, and Training". *Procedia CIRP* 32:1–6.

Abele, E., G. Chryssolouris, W. Sihn, J. Metternich, H. ElMaraghy, G. Seliger, G. Sivard, W. ElMaraghy, V. Hummel, M. Tisch, and S. Seifermann. 2017. "Learning Factories for future oriented research and education in manufacturing". *CIRP Annals – Manufacturing Technology* 66(2):803–826.

Bangsow, S. 2010. *Manufacturing Simulation with Plant Simulation and SimTalk: Usage and Programming with Examples and Solution.* Berlin, Heidelberg: Springer.

Bangsow, S. 2015. *Tecnomatix Plant Simulation: Modeling and Programming by Means of Examples.* Heidelberg, New York, Dordrecht, London: Springer.

Banks, J., J.S. Carson, B.L. Nelson, and D.M. Nicol. 2005. *Discrete-Event System Simulation.* 4th Ed. Upper Saddle River: Pearson Prentice Hall, Inc.

Banks, J., J.S. Carson, B.L. Nelson, and D.M. Nicol. 2010. *Discrete-Event System Simulation.* 5th Ed. Upper Saddle River: Pearson Prentice Hall, Inc.

Byrne, J., P. Liston, D.C. E. Ferreira, and P.J. Byrne. 2015. "Cloud based data capture and representation for Simulation in Small and Medium Enterprises". In *Proceedings of the 2015 Winter Simulation Conference Social and Behavioral Simulations,* edited by L.Yilmaz, W. Chan, I. Moon, T. Roeder, C. Macal, and M. Rossetti, 2195-2206. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Deterding, S., D. Dixon, R. Khaled, and L. Nacke. 2011. "From Game Design Elements to Gamefulness: Defining Gamification". In *Proceedings of the 15th International Academic MindTrek Conference Envisioning Future Media Environments,* edited by A. Lugmayr, H. Franssila, I. Hammouda, and C. Safran, 9–15. New York: Association for Computing Machinery.

Despeisse, M. 2018. "Games and Simulations in Industrial Engineering Education: A Review of the cognitive and affective learning outcomes". In *Proceedings of the 2018 Winter Simulation Conference Simulation for a Noble Cause,* edited by M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 4046-4057. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

ElMaraghy, W., H. ElMaraghy, T. Tomiyama, and L. Monostori. 2012. "Complexity in engineering design and manufacturing". *CIRP Annals - Manufacturing Technology* 61(2):793–814.

Google Blockly. 2020. Google for Education Blockly: Try Blockly. https://developers.google.com/blockly, accessed 20th April 2020.

Gutenschwager, K., M. Rabe, S. Spieckermann, and S. Wenzel. 2017. *Simulation in Produktion und Logistik: Grundlagen und Anwendungen.* Berlin, Heidelberg: Springer.

Kagermann, H., W. Wahlster, and J. Helbig. 2013. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Final report of the Industrie 4.0 Working Group.* Acatech – National Academy of Science and Engineering. https://en.acatech.de/, accessed 20th April 2020.

Kelton, W.D., R.P. Sadowski, and N.B. Swets. 2010. *Simulation with Arena; McGraw-Hill International Editions.* 5th ed. Boston: McGraw-Hill Higher Education.

Kelton, W.D., R.P. Sadowski, and N.B. Zupick. 2015. *Simulation with Arena.*6th ed. New York: McGraw-Hill Education.

Kolb, D.A., I.M. Rubin, and J.M. McIntyre. 1971. *Organizational Psychology: an experiential approach.* New Jersey: Prentice-Hall, Englewood Cliffs.

Kühn, W. 2006. *Digitale Fabrik: Fabriksimulation für Produktionsplaner.* München, Wien: Carl Hanser Verlag.

Kurihara, A., A. Sasaki, K. Wakita, and H. Hosobe. 2015. "A Programming Environment for Visual Block-Based Domain-Specific Languages". *Procedia Computer Science* 62:287–296.

Law, A.M. 2015. *Simulation Modeling and Analysis.* 5th international student Ed. New York: McGraw-Hill.

Lödding, H. 2016. *Verfahren der Fertigungssteuerung: Grundlagen, Beschreibung, Konfiguration.* 3rd ed. Berlin, Heidelberg: Springer.

Manrique, V., I. Rodrigo, O. Garcia-Panella, E. Labrador, A. Marczewski, F. Escribano, C. Solarski, C. Pagés, and G.M. de Polavieja. 2015. Iversity Gamification Design. https://iversity.org/en/courses/gamification-design-2015, accessed 20th April 2020.

Mayer, G., C. Pöge. 2010. "Auf dem Weg zum Standard – Von der Idee zur Umsetzung des VDA Automotive Bausteinkastens". In *Integrationsaspekte der Simulation: Technik, Organisation und Personal,* edited by G. Zülch, and P Stock, 29–36. Karlsruhe: Arbeitsgemeinschaft Simulation-Mitteilung; KIT Scientific Publishing.

Müller, B.C., C. Reise, and G. Seliger. 2015. "Gamification in Factory Management Education – A Case Study with Lego Mindstorms". *Procedia Manufacturing,* 26:121–126.

Müller, B.C., J. P. Menn, and G. Seliger. 2017. "Procedure for Experiential Learning to Conduct Material Flow Simulation Projects, Enabled by Learning Factories". *Procedia Manufacturing* 9:283–290.

Müller, B.C., M. Kühn, M. Schele, and G. Seliger. 2017. "Mobile Lernumgebung zur simulativen Gestaltung von Produktionssystemen". In *Simulation in Produktion und Logistik,* edited by S. Wenzel, and T. Peter. kassel university press, Kassel: 345–354.

Noche, B., and S. Wenzel. 1991. *Marktspiegel Simulationstechnik in Produktion und Logistik.* Köln: TÜV Rheinland.

Noche, B., W. Bernhard, J. Krauth, R. Meyer, and S. Wenzel. 1993. "Simulationsinstrumente im Überblick". In *Handbuch Simulationsanwendungen in Produktion und Logistik,* edited by A. Kuhn, A. Reinhard, and H.-P. Wiendahl. 267–308. Braunschweig, Wiesbaden: Friedr. Vieweg & Sohn Verlagsgesellschaft mbH.

Padilla, J.J., S.Y. Diallo, A. Barraco, C.J. Lynch, and H. Kavak. 2014. "Cloud-based simulators: Making simulations accessible to non-experts and experts alike". In *Proceedings of the 2014 Winter Simulation Conference. Exploring big data through simulation,* edited by A. Tolks, S. Diallo, I. Ryzhov, L. Yilmaz, S. Buckley, and J. Miller, 3630–3639. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Padilla, J.J., E. Romero-Hall, S.Y Diallo, A. Barraco, H. Kavak, C.J. Lynch, R.J. Gore, and M. Sheth-Chandra. 2015. "Modeling and Simulation As a Service (MSaaS) for Education: Learning STEM Concepts Through Simulation Use and Building". In *Proceedings of the Conference on Summer Computer Simulation (SCSC)*, edited by S. Mittal, I. Moon, and E. Syriani, 1–9. San Diego, CA: Society for Computer Simulation International.

Padilla, J.J., C.J. Lynch, S.Y Diallo, R.J. Gore, A. Barraco, H. Kavak, and B. Jenkins. 2016. "Using simulation games for teaching and learning discrete-event simulation". In *Proceedings of the 2016 Winter Simulation Conference. Simulating complex service systems,* edited by T.M. Roeder, P.I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S.E. Chick, 3375–3384. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rabe, M., S. Spieckermann, and S. Wenzel. 2008. "A new procedure model for verification and validation in production and logistics simulation". In *Proceedings of the 2008 Winter Simulation Conference,* edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, 1717-1726. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Schmidt, B. 1988. "Simulation von Produktionssystemen". In *Simulation in der Fertigungstechnik,* edited by K. Feldmann, and B. Schmidt, 1–45. Berlin: Springer.

Schmidt, A., M. Schneider. 2008. "Lager- und Materialflussprozesse". In *Handbuch Logistik,* edited by D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier, K. Furmans, 371–404. Berlin, Heidelberg: Springer.

Schumacher, B.C. 2020. *Contribution to the introduction into discrete-event simulation for the new- and re-planning of material flow systems.* Ph.D. thesis, Department of Assembly Technology and Factory Management, Technische Universität Berlin, Berlin. http://dx.doi.org/10.14279/depositonce-9586, accessed 14th April 2020.

Scratch. 2020. About Scratch. https://scratch.mit.edu/about, accessed 20th April 2020.

Shannon, R.E. 1998. "Introduction to the art and science of simulation". In *Proceedings of the 1998 Winter Simulation Conference,* edited by D.J. Medeiros, E.F. Watson, J.S. Carson, and M.S. Manivannan, 7–14. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Swain, J.J. 2015. "Simulation Software Survey: Simulated worlds". *OR/MS Today* 42(5):36–49.

Swain, J.J. 2017. "Simulation Software Survey - Simulation: new and improved reality show". *OR/MS Today* 44(5):38–49.

VDI. 2011. *VDI 5200-1: Factory planning: Planning procedures.* Association of German Engineers. Berlin: Beuth Verlag.

VDI. 2013. *VDI 3633-0: Simulation von Logistik-, Materialfluss- und Produktionssystemen: Begriffe.* Association of German Engineers. Berlin: Beuth Verlag.

VDI. 2014. *VDI 3633-1: Simulation of systems in materials handling, logistics and production: Fundamentals:* Association of German Engineers. Berlin: Beuth Verlag.

Wenzel, S., 2008. "Modellierung logistischer Systeme: Simulation Logistischer Systeme". In *Handbuch Logistik*, edited by D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier, and K. Furmans, 73–94. Berlin: Heidelberg, Springer.

AUTHOR BIOGRAPHIES

BASTIAN C. SCHUMACHER is a postdoctoral researcher and program manager of the international postgraduate master study program Global Production Engineering in the Institute for Machine Tools and Factory Management at Technische Universität Berlin. He holds a Ph.D. in production engineering from Technische Universität Berlin. To date, he held 20 teaching courses about DES for postgraduate students at different universities. His research interest is in innovative approaches for DES education, digital factory, sustainable manufacturing, production planning and control, and logistics. His email address is bastian.schumacher@tu-berlin.de.

HOLGER KOHL is a professor at Technische Universität Berlin, holding the chair for corporate sustainable development. He is deputy director of the Fraunhofer Institute for Production Systems and Design Technology IPK. He is the dean of the international postgraduate master study program Global Production Engineering at Technische Universität Berlin. Holger Kohl is chairman of the annual Global Conference on Sustainable Manufacturing. He holds a Ph.D. in production engineering from Technische Universität Berlin. His research interests include Business Excellence Methods as well as Business Process and Factory Management. His e-mail address is holger.kohl@ipk.fraunhofer.de.