
SIMULATION MODEL GENERATION FOR WAREHOUSE MANAGEMENT: CASE STUDY TO TEST DIFFERENT STORAGE STRATEGIES

Abstract: A simulation model generator was developed to help a company of the Bosch Group to reduce costs in time and space with its warehouse. Particularly, the automatically created simulation models can simulate pickers riding milk runs to collect containers from the warehouse, to satisfy the needs of production lines, enabling warehouse management by testing different storage strategies. Thus, the purpose of this paper is to present the developed generator and to use it in a case study, to test different storage policies for the organization. The generator was validated, as was the simulation model automatically created. With this tool, some suggestions could be made to the organization in question. Namely, it was shown that it would not be possible to maintain the FIFO rule and, at the same time, eliminate the upper floor of all racks for ergonomic reasons. To allow this, the rate of replacing containers should be synchronized with the needs of production lines. Alternatively, it was also shown that the performance of the warehouse would improve by dividing it in zones affected for each milk run.

Keywords: Simulation, warehouse management, milk run, picking, Object modelling, Simio.

Reference to this paper should be made as follows: Author(s) (2006) 'paper title', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. X, No. Y4, pp.000–000.

Biographical notes: (ABS)

1 Introduction

Bosch Group has been applying concepts of the Toyota Production System (TPS) (Monden, 1998) and Lean Manufacturing (Womack et al., 1990, Womack and Jones, 1996), designated as Bosch Production System (BPS). Its purpose is to eliminate waste in production and all related business processes. Thus, it provides the basis for continuous improvements in costs, quality, and supply performance (Bosch, 2014, Abidi et al., 2016).

A significant part of the costs of a company are concerned with their warehouses (Baker and Canessa, 2009), which is used to store products that are latter transported to different locations within an industrial facility, which is known as internal logistics (Alnahhal et al., 2014). In this regard, the need to study alternatives to the current storage strategy of the warehouse of the company Bosch Car Multimedia Portugal, arose. The warehouse of the case in question stores products used for the final assembly activities of finished products and is designated as supermarket (Klenk et al., 2015).

The purpose of this paper is twofold. First it presents a simulation model generator able of automatically building warehouse simulation models. This way, the generated models can be adapted to new layouts, or even test those layout alternatives. Secondly, this paper intends to use the generated simulation model in a case study where it is intended to analyse different storage strategies for the supermarket of the organization of the case study. Both the generator and the automatically built simulation model have been validated, through generating the models and running them to compare the results with observations on the field.

Ultimately, it is expected that the analysis of the results retrieved by the simulations can be used to improve the performance of the supermarket of the company. More specifically, the company expects that the FIFO (First In First Out) rule of its supermarket is maintained and wants to reduce their supermarket space.

The automatically generated simulation models of the hereby proposed tool are capable of modelling pickers riding milk runs to collect containers of products, from the supermarket, to satisfy the needs of the production lines. Apart from generating different layouts, the simulation models can also use real data to simulate the intended milk run routes, therefore these are not static (Silva et al., 2016).

Simio was chosen for this project, for many reasons. Firstly, it is becoming one of the most used tool, despite its young age (Dias et al., 2007, Dias et al., 2011, Dias et al., 2016). Secondly, its object paradigm, direct interface with Google 3D Warehouse and the built-in 3D animation allows the possibility of building complex systems that are, at the same time, similar to the real system, which enhances the communication with stakeholders of the project. Lastly, it provides an API (Application Programming Interface) to allow users to develop additional user-defined logic and add-ins that automatically build simulation models.

The next section presents a literature review. In section 3, the components of the developed simulation model generator will be presented. Section 4 presents the case study in question and the simulation experiments results obtained are discussed. The last section discusses the main conclusions and sets a future research agenda.

2 LITERATURE REVIEW

Warehouses provide time and place utility for industrial goods, raw materials or finished products, enabling companies to use this service as a dynamic value-adding competitive tool. Thus, warehouses represent a very important role on modern supply chains (Baker and Canessa, 2009).

In fact, whilst warehouses are critical to a wide range of customer service activities, they can also have a major impact, from a cost perspective. Figures for the USA suggest that the capital and operating costs of warehouses represent about 22% of logistics costs (Establish, 2005), whilst figures for Europe give a similar figure of 25% (Baker and Canessa, 2009). Moreover, it is also estimated that picking operations account for more than 55% of the total cost of warehouse operations (Bottani et al., 2015). These costs impel us to understand the problematic, namely to use god storage policies and to use the storage space as efficiently as possible, as suggested by Bartholdi and Hackman (2008).

Thus, the need to provide companies with methods capable of improving the performance of their warehouses arises. Some of these methods include simulation, analytical methods and benchmarking, the former being the most used whether in literature or in practice in problems related to logistics (Gu et al., 2010, Bottani et al., 2014), despite some of its flaws that can be identified, as Tajini et al. (2014) did.

One example is the simulation model developed by Costa et al. (2008) using Arena. The authors conducted experiments to identify changes that could be made on a material delivery system to improve the efficiency and precision of the logistic train functioning that they were modelling. The same tool was used by Penker et al. (2007) to analyse production, storage and transporting processes of an Austrian production plant. Bottani et al. (2012) used Excel to optimise the allocation of items in a warehouse, with the goal of reducing the total travel time of picking operations. In its turn,

Since the number of simulation tool options can be very high, tool comparison becomes a very important task. However, most of scientific works related to this subject analyse a small set of tools and evaluate several parameters individually, avoiding to make a final judgement, due to the subjective nature of that task (Dias et al., 2007, Dias et al., 2011, Dias et al., 2016).

Hlupic and Paul (1999) compared a set of simulation tools, distinguishing between users of software for educational purpose and users in industry. In his turn, Hlupic (2000) developed a survey on the use of simulation software of academic and industrial users, which was conducted to discover how the users were satisfied with the simulation software they used and in which ways could the software be improved.

In their turn, Dias and Pereira et al. (2007, 2011, 2016) compared a set of tools based on popularity on the internet, scientific publications, WSC (Winter Simulation Conference), social networks and other sources. According to the authors, popularity should not be used as the only criteria, otherwise new tools, better than existing ones

would never get their market share. However, a positive correlation may exist between popularity and quality, since the best tools have a greater chance of being more popular. According to their study, the most popular tool is Arena, however, the good classification of Simio is also noteworthy. Based on these results, Vieira et al. (2014) compared both tools taking into consideration several factors.

Simio was created in 2007 from the same developers of Arena and is based on intelligent objects (Sturrock and Pegden, 2010, Pegden, 2007, Pegden and Sturrock, 2008). Unlike other object-oriented tools, in Simio there is no need to write programming code, since the process of creating objects is completely graphic (Pegden and Sturrock, 2008, Pegden, 2007, Sturrock and Pegden, 2010). The activity of building an object in Simio is identical to the activity of building a model. In fact, there is no difference between an object and a model. A vehicle, a customer or any other agent of a system are examples of possible objects and, combining several of these, one can represent the components of the system in analysis. In other words, the user can use realistic representations of the objects that compose the real system being modelled and, thereafter, at a lower level, define additional logic to the model, through the development of processes for instance. This way, Simio complements the main object paradigm with other paradigms such as events and processes.

Thus, a Simio model looks like the real system, which can be useful when presenting the results to those non-familiar to simulation concepts. In Simio the model logic and animation are built in a single step (Pegden and Sturrock, 2008, Pegden, 2007), which makes the modelling process very intuitive. In addition to the usual 2D animation, Simio also supports 3D animation as a natural part of the modelling process. To switch between 2D and 3D views the user only needs to press the 2 and 3 keys of the keyboard. Moreover, Simio provides a direct link to Google Warehouse, a library of graphic symbols for animating 3D objects (Oueida et al., 2016). Despite the many aforementioned features that Simio has that were beneficial for this project, it also provides an API (Application Programming Interface) that can be used to create user-defined logic and to automatically build simulation models.

This is an important issue for several reasons. For instance, automatically generating simulation models is a way to reduce errors in simulation models, since the model is built by the computer. Rather than individually validating models, this process should focus on the generator. Therefore, by using these generators, a reduction in the validation process complexity, durability and subjectivity is expected (Popovics et al., 2016, Tajini et al., 2014). In addition, more time would be available for more added-value tasks, such as results analysis.

Más et al. (2016) used simulation to support the decision-making in layout redesign of an engine and transmission assembly plant, including warehouse, in a company of the automotive sector, using an automatic simulation model generator. In its turn, Haraszko and

Németh (2015) also developed an automatic simulation model generator in Tecnomatix Plant Simulation. Guasch et al. (2011) used Arena to test different warehouse configurations on a newspaper printing plant.

A parallelism can be traced with the history of simulation since its inception and automatic simulation model generation. A detailed history of simulation can be found in Robinson (2005). The technique started with simulation programming languages, until the advent of VIS (Visual Interactive Simulation), or just animation, around the 1970s. Thereafter, in the 1980s, commercial software tools started to group programming code lines to create simulation blocks that could be dragged, dropped and connected, forming a visual programming paradigm. This change was triggered both by the need to simplify the development phase and by the increasingly complexity of systems. Nowadays, problems are even more complex and the development time also needs to be shortened. As Robinson (2005) and Fowler and Rose (2004) agree, this can be tackled by either automatically generating simulation modes, or by creating simulation models that can be reused. The reusable models can be entire models or sub-models that represent smaller parts of a system and can be applied to other models.

3 Case Study Description

The case study consists on modelling an advanced warehouse system, i.e. a supermarket, of a Bosch plant. The supermarket is comprised of channels (shelf-like structure that stores containers in depth, in FIFO order), each one storing containers that arrive at certain periods of time – replacement of containers – which can store several product units of a single product type. Figure 1 represents the main structures and container flows associated to this system.

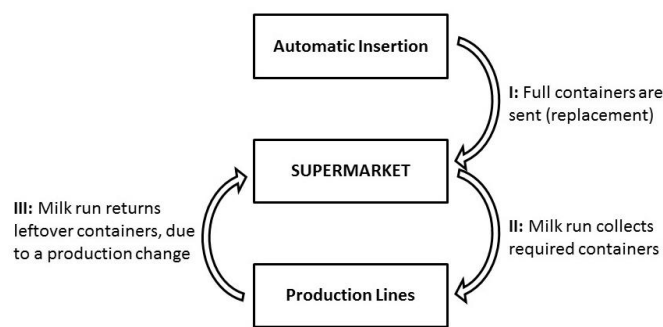


Figure 1: System description

Containers are sent to the supermarket (flow I of containers), for later being collected by pickers that travel through the supermarket, driving milk runs (flow II of containers). After collecting the intended containers, pickers deliver them to the respective production lines. These consume the required material and, when it is necessary to start consuming a different type of product, a reference change occurs. In some cases, this phenomenon can result on a container being returned to the supermarket with the leftover product units inside of it (flow III of containers).

Several storage strategies exist to determine how products are stored in a warehouse. In this case study, the

company uses a single-product strategy (dedicated), which is to the most simple one, since it consists on having channels dedicated to a single type of container (Bartholdi and Hackman, 2008). Its greatest advantage is the fact that, pickers can memorize the locations of the containers, since these do not change over time, making the picking process more efficient. Moreover, it should be expected that single-product strategies require a higher quantity of channels to work, since it does not store different types of product on the same channel. In other words, the problem with this strategy is that it does not use in an efficient way the available space.

Alternatives to this strategy would have to allow containers of different types to be mixed within the same channel (multi-product), whereby some companies oppose to its implementation. The main reason for this opposition is that the Information System (IS) would have to be much more complex in more than one way. Firstly, it would have to manage the location of containers, in order to avoid picking from the non-first position, which has negative effects on the picking operation. Secondly, it would have to guide pickers to the proper channel and to the right position, once they would no longer have the advantage of having memorized the location of the containers. If the IS cannot handle these issues, pickers must search for the container through all the positions of all the channels of the supermarket, which would also negatively affect the picking system.

Such an IS is being implemented in the organization of the case study. As such, the possibility of changing the single-product storage strategy of the company, in order to reduce the space of the supermarket, arose. Thus, the case study consisted on using the developed simulation model generator to automatically build a warehouse simulation model and use it to test different storage strategies for the Bosch organization.

Despite allowing mixture of containers of different types per channel, to comply with BPS, the proposed alternatives for the company need to ensure that containers will always be collected from the first position on each channel. In addition, a goal of the organization is to evaluate the possibility of eliminating the top floor of channels of the supermarket, as all have 3 floors. Since the supermarket has 930 channels, this would mean a reduction of at least 310 channels. Table 1 shows the considered storage strategies.

Table 1: Storage strategies description

Short name	Storage Strategy
A	Single-product channels;
A2	Single-product channels; Preferential areas for each milk run;
B	Multi-product channels; Driven by arrivals to the supermarket; Preferential areas for each milk run;
C	Multi-product channels; Driven by consumption; Only stores the containers that will be required in the current week of work Preferential areas for each milk run;

C2	Multi-product channels; Driven by consumption; Preferential areas for each milk run;
----	--

Strategy A corresponds to the one already implemented at the organization. Through field observations and analysis of the input data, provided by the company, it was found that the storage of containers on the supermarket did not consider its division in areas for each milk run. Therefore, being a variant of the previous strategy, Strategy A2 considers this division, by maintaining the standard of having dedicated channels. Figure 2 displays the simulation running, while modelling a single-product storage strategy, where different colours were assigned for each type of container. As can be seen, all containers stored within the same channel have the same colour.

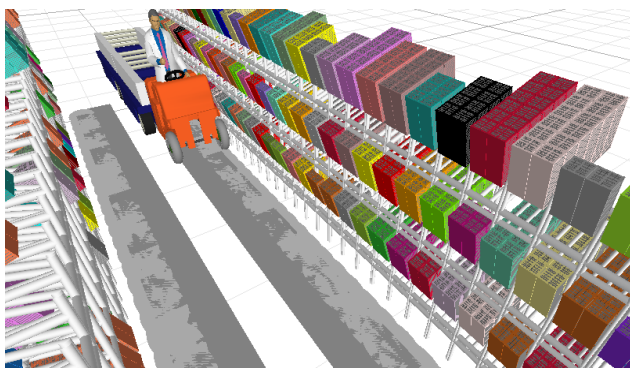


Figure 2: Single product storage strategy

Considering the above mentioned, the alternatives for the company must analyse if, by allowing mixes of containers of different types on each channel, the picking of the containers will always be made in the first position on each channel. In this sense, Strategy B consists on allowing a container to be stored on a channel if it was sent on a posterior date. This date must respect a time interval that will be addressed in further sections of this document. Moreover, a limit establishing the maximum number of different type of containers allowed per channel also must be respected. Therefore, this strategy is highly dependent on the consistency of the dates on which the containers are sent to the supermarket and required by the production lines, since it expects that if container A is sent to the supermarket sooner than container B, it will also be required by the production lines sooner.

Strategy C consists on storing containers, based on their consumption date, by giving priority to the channels that already have a container of the same type. Thus, a container can be stored in a channel with containers of other types, if the container to be stored has a posterior consumption date. Additionally, and similarly to the previous storage policy, an established limit of the number of containers of different types allowed per channel must be respected. Since consumption dates are exposed to prediction errors, the impact of eventual errors must be analysed. Lastly, this strategy considers that only the containers that will be consumed throughout the current week of work are stored on the supermarket, since the model assumes that the IS of

the company can predict the consumptions and therefore only needs to store what will be used in the current week.

Strategy C2 is a variant of strategy C, in the sense that it does not consider that all the containers that are on the supermarket will be consumed and that only the containers that will be consumed in the current week are stored in the supermarket. Thus, it is expected that more space will be needed. Figure 3 shows the simulation model in execution, while modelling a multi-product storage strategy (either B, C or C2). As can be seen, in this case, containers of different colour can be seen within a same channel.

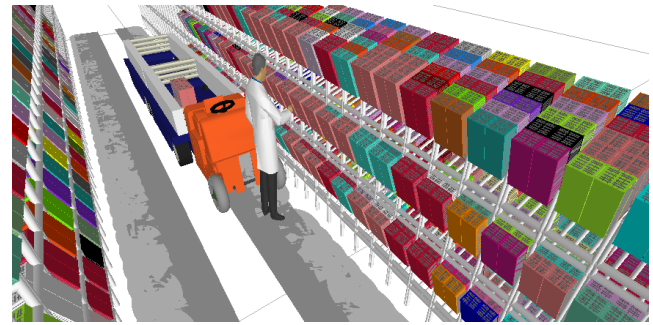


Figure 3: Multi-product storage strategy

4 Components of the Simulation Model Generator

In this section, components of the developed simulation model generator will be addressed. Figure 4 illustrates the structure of the developed simulation model generator.

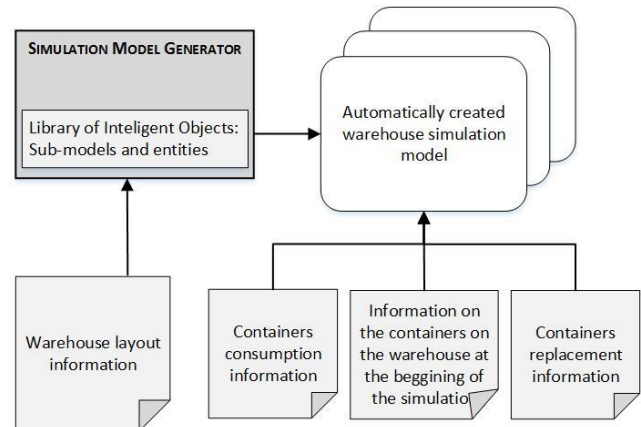


Figure 4: Structure of the developed generator





As can be seen, for the generator to automatically build a warehouse simulation model, it needs information on the layout of the warehouse in question, using a library of intelligent objects. This library comprises:

- A set of sub-models that were user-defined in the scope of this project. Sub-models can also be used to define other sub-models;
- A set of entity types that were user-defined in the scope of this project;
- And a set of standard Simio objects (e.g. paths between objects, combiners to create batches of entities, etc.) that were used to define the sub-models and in the main simulation model, e.g. to connect different sub-models.

Thereafter, the generator builds the intended simulation models, which uses data related to the containers that: exist at the beginning of the simulation, are replaced and are consumed during the simulation model run. Alternatively, random distributions can also be used. Thus, a model that represents the system in analysis is obtained without spending long times in the development phase. In the remaining of the section, some important components of the generator will be analysed.

For this simulation project, 4 types of entities and 5 models (4 sub-models and a main one) were created. The defined entity types and the corresponding symbol representation are detailed in Table 2.

Table 2: Types of entities and corresponding symbols

Type of entity	Symbol
Picker: Represents the pickers of the system. Their functions are to collect Requests at the beginning of a shift and take Containers from Channels of the Supermarket to place them on the milk run.	
Milk run: Represents the milk runs of the system. Its only purpose is to transport the Picker and the selected Containers between the Supermarket.	
Request: Represents the request of the system	
Container: Represents the containers of the system.	

To build the simulation models, a set of pre-defined sub-models, or objects, are used. These objects represent real components of the system. A complete description of all these objects is given by Vieira et al. (2016). Simio allows to build intelligent objects on two different levels. The first is the facility, where simulation objects are used to create the core logic of the model and the respective animation, in a single step. These can then be complemented at the process level with additional logic. Lastly, it is possible to assign a physical 3D representation of the sub-model. Thus, considering for instance the sub-model that represents a channel, Figure 5 shows its 3D representation. This sub-model stores containers and models the behaviour of the Pickers, when they analyse a channel to select the container they want.

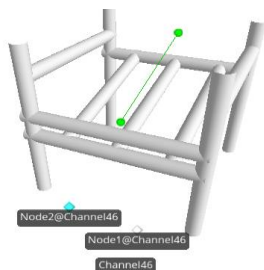


Figure 5: External view of the Channel model

By defining sub-models for the different objects that compose a warehouse system, a library is obtained, which may thereafter be used to automatically build the intended simulation model.

Apart from the Chanel sub-model and the other sub-models already presented by Vieira et al. (2016), two additional sub-models were created – DeliveryPoint and ProductionLine. These sub-models concern similar situations. ProductionLine, as the name implies, models the behaviour of pickers and milk runs when they finish a shift of collecting containers from the warehouse and must deliver them to a production line. This may involve collecting containers with leftover products, as a result of a production change in the production lines. These containers need to be returned to the warehouse by the milk run. Figure 6 shows the representation of this sub-model.

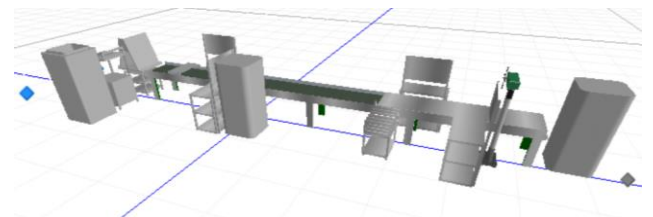


Figure 6: External view of the ProductionLine sub-model

The DeliveryPoint sub-model also models the behaviour of pickers and milk runs when they finish a picking shift. However, in this case, there is no need to model the processing time of the production lines, in order to return the leftover containers to the warehouse. The authors decided to create two separate sub-models for these similar situations because they have different physical representations and one involves processing time and other does not. Furthermore, the need to develop the DeliveryPoint only arose due to the application of this simulation model generator to another case study at a different international organization, as will be discussed further in this paper. In its turn, Figure 8 shows the objects that were used to create this sub-model. As can be seen, sub-models can also be used to define other sub-models. In this case, the GoToMilkRun sub-model was used (Vieira et al., 2016).

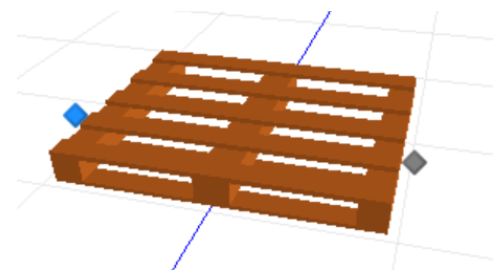


Figure 7: External view of the DeliveryPoint sub-model

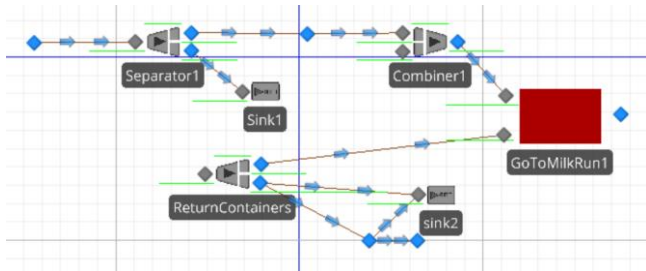


Figure 8: Facility view of the DeliveryPoint sub-model

4.1 Automatic Creation of the Simulation Model

The generator reads the layout specification data from an excel file. It was established this way, since excel is already widely used in the organization, as such it would not require any familiarization.

The layout of the supermarket can change from time to time. As such, through many meetings on the ground, it was verified what were the main parameters that could change, in order to allow the users of the generator to change them. Therefore, some rules were established for the data input in the excel file, which will be addressed in this subsection. Table 1 shows an example of the content of this file.

Table 3: Input Excel table

New corridor?	Size			Coordinates		Symbol index	Directions	Rack description	Channels per column					
	Length	Width	Height	x	y (z in Simio)				1	2	3	4	5	6
1	0,23	0,42	0,58	-50	-50	0	2	AP	3	3	3	3	3	3
0								AO	3	3	3	3	3	3
0								AN	3	3	3	3	3	3
0								AM	3	3	3	3	3	3
0								AL	3	3	3	3	3	3
0								AJ	3	3	3	3	3	3
0								AK	3	3	3	3	3	3
0								AI	3	3	3	3	3	3
0								AH	3	3	3	3	3	3
0								AG	3	3	3	3	3	3
0								AF	3	3	3	3	3	3
0								AE	3	3	3	3	3	3
0								AD	3	3	3	3	3	3
0								AC	3	3	3	3	3	3
0								AB	3	3	3	3	3	3
0								AA	3	3	3	3	3	3
2								BE	3	3	3	3	3	3
0								BD	3	3	3	3	3	3
0								BC	3	3	3	3	3	3
0								BB	3	3	3	3	3	3
0								BA	3	3	3	3	3	3
0								AZ	3	3	3	3	3	3
0								AY	3	3	3	3	3	3
0								AX	3	3	3	3	3	3
0								AW	3	3	3	3	3	3
0								AV	3	3	3	3	3	3
0								AU	3	3	3	3	3	3
0								AT	3	3	3	3	3	3
0								AS	3	3	3	3	3	3
0								AR	3	3	3	3	3	3
0								AQ	3	3	3	3	3	3

User can insert data related to a rack, per line. This way, they can specify any number of racks per corridor. Thus, they only need to signalize what is the first rack of a corridor, by entering the value 1 on the first column of this excel file. On the other hand, if the user wants to keep adding racks to a previously stated corridor, the user just needs to keep entering the value “0” on the next rows. For each corridor, the user can also set the number of directions

in which milk runs can travel. Moreover, users can also set the size of the channels (length, width and height), the position on which the corridors start to be built, the 3D representation to be assigned to the channels and the rack identifier.

The user can also set the type of corridor, i.e., if a milk run can reach a single set of racks from one of its sides, while travelling, or if it can reach two sets of racks: one from each of its sides. Lastly, the user can define any number of columns per rack and any number of channels per column, depending on the number of cells that have values and the values on each of those cells, respectively.

The meaning of the information stored in this excel file is translated into a simulation model via the developed add-in, which was created using C# code that implements the methods of the Simio API. The description of the developed add-in is given by Vieira et al. (2016).

4.2 Validation of Generated Simulation Models

Through many meetings, the authors obtained data required for the simulation model to efficiently model the system in analysis. Among others, values for the speed of the milk run, the picker, devolution rates, production times, number of shifts per day, number of production lines, time to remove containers from their channels and others were collected. This process is important, since it increases the confidence level in the developed model.

After analysing the raw data provided by the company, the authors could produce, using VBA, the required excel files that would “feed” the developed simulation model with real data. To do so, this data was imported into Simio – see Figure 4.

The validation of the developed generator was conducted in two steps. First, to validate the layout of the automatically created simulation models, these were visually compared with instances that had been manually created and validated on the field together with the responsible from the organization. The next step was to verify if all the properties of the objects comprising the generated simulation models were correctly set and if the obtained results did not have significative differences. For this reason, several metrics were defined and compared.

The created supermarket corresponds to a single corridor of two sets of channels that can be accessed by a picker who travels in between them. This supermarket corresponds to a size of 930 channels. Each channel has the capacity to hold 6 containers. More examples of automatically created simulation models, using the Simio add-in developed for this project, can be found online (Vieira et al., 2015). Figure 9 shows the automatically built simulation model, using the generator. At the left the model is displaying all objects that were created and, at the right, only those most relevant to the animation of the model are visible.

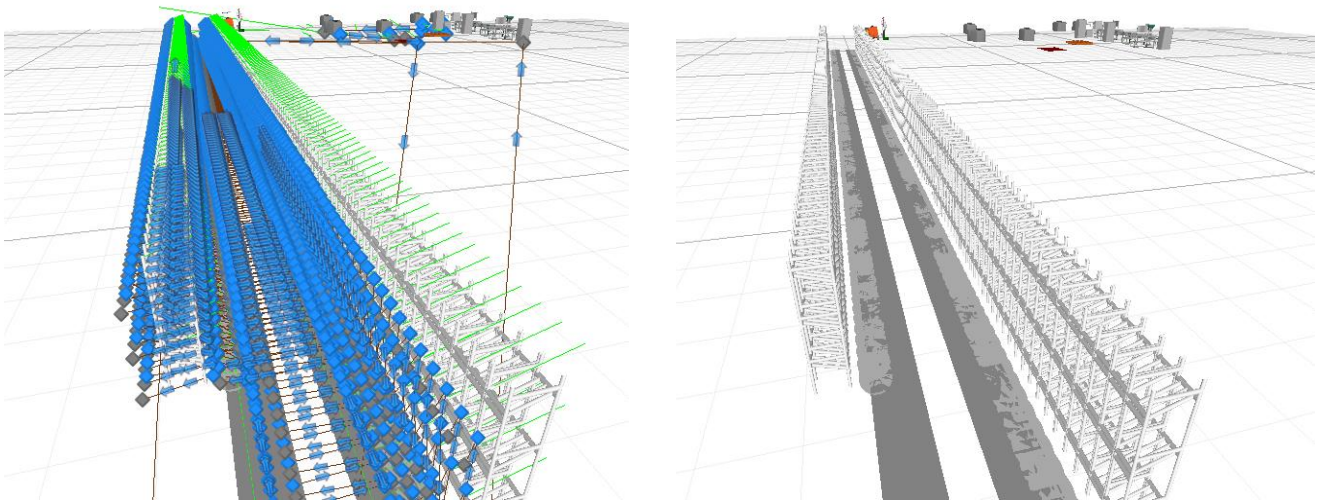


Figure 9: Simulation model created for the case study: (a) at the left shows all created objects; and (b) at the right only showing the most relevant for animation purposes

The simulation model generator is intended to be applicable to a variety of different warehouses. As such using it in other warehouses is a goal in future works, which is already starting to be done. Figure 10 shows another warehouse simulation model that was automatically created for a different case study in a different international organization of the drinks sector. This model has already been validated, which further increases the confidence level on the generator and on the simulation model used in the present case study.

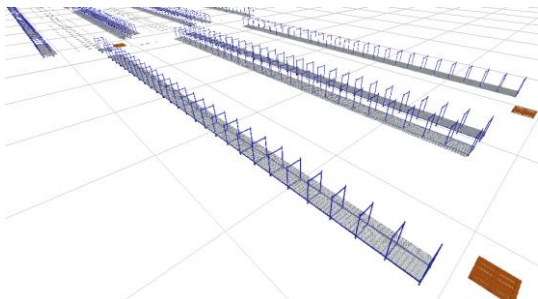


Figure 10: Simulation model generated for future study

In this figure, the DeliveryPoint sub-model can be identified. The other object is the Channel submodel, with a different 3D representation, to better illustrate the storage unit used in this second case study.

5 Simulation Experiments

In this section, the results obtained from the simulation experiments will be presented. The experiments were conducted on an automatically built simulation model that represents the system of the case study organization in question.

Hundreds of simulation scenarios were defined for this problem, however only some of them will be presented in this chapter. The conducted experiments were run with a simulation time of one week. It should be noted that the conclusions withdrawn from this comparison should not be generalized, since it corresponds to a specific studied case.

The main properties of the conducted simulation experiments were:

- The storage strategy;
- A random distribution that assigns an error to the expected consumption data of containers. This property was added, since multi-product strategies (C, C2 and B) rely on the existence of an IS capable of predicting the consumption date of the stored containers, however errors can still occur;
- Time interval to keep between containers of different types, in each channel;
- And the number of different types of containers allowed per channel;

The simulation experiments conducted in Simio considered several performance indicators: To quantify the different simulation scenarios, rather than using an explicit **multi-criteria** approach, weights were assigned to the four main KPI (Key Performance Indicators), in order to obtain a score that considers the values of the four main KPI and that relates all scenarios, resulting in a global classification ranging from 0 to 100.

Table 4: Main KPI and respective weights

Key Performance Indicator	Weight
the average total time in picking shifts in seconds	1
the average position from which containers are removed from channels (depth)	2
the total number of channels that were never used throughout the simulation (unused channels)	3
and the average number of stops per milk run per picking shift	1

5.1 Results and Analysis

Table 5 shows the obtained results for the A and A2 strategies.

Table 5: Simulation results for strategy A

Strategy	Time	Error	Different types	Total	Number of Unused	Depth	Global	
gap	of containers	time	stops	channels	Classification			
A	0	0	1	243,7	3,90	148	1	45%
A2	0	0	1	215,6	1,80	164	1	62%

As the results indicate, the pickers always collected the containers from the first position (depth), which is one of the perks of using this strategy. However, this affects the number of channels that were not used, which is lower than the same KPI on the remaining strategies, as will be shown in the next sections. Moreover, just by dividing the supermarket into preferential areas for each milk run/picker (strategy A2), it is possible to reduce the total time spent on each picking shift in about 30 seconds, which corresponds to a reduction of 12%. However, the major benefit comes from the reduction of the number of stops per milk run. In fact, as Table 5 suggests, the pickers collected the same containers with about 58% less stops (average of less 2 stops per picking shift). The reason for this is that in strategy A2 pickers can easily access all their action area with a single stop, since the containers they are supposed to collect are all located near each other, whereas in strategy A the same does not apply. Since this division of the supermarket in areas for each milk run revealed to be very profitable, it was also considered for the remaining storage strategies (B, C and C2). Regarding the classification, it is possible to see that it was possible to improve the performance by 17%. Table 6 shows the results obtained for strategy B, when analysing the impact of the different number of types of containers allowed per channel.

Table 6: Simulation results for strategy B to assess the impact of different containers per channel

Strategy	Time gap	Error	Different types of containers	Total time	Number of stops	Unused channels	Depth	Global Classification
B	24	0	1	215,4	1,78	162	1,000	62%
B	24	0	2	223,0	1,77	197	1,123	53%
B	24	0	3	227,9	1,76	197	1,122	53%
B	24	0	4	227,9	1,76	197	1,122	53%
B	24	0	5	227,9	1,76	197	1,122	53%
B	24	0	6	227,9	1,76	197	1,122	53%

As can be seen, the limitation of the number of different types of containers allowed on each channel did not affect the system. In fact, it is only when the number is reduced to 1 that significant differences can be noted and the highest global classification is achieved. However, this scenario is equivalent to the strategy A2. As such, limiting this number to 2, or 6 – the maximum capacity of the channels in the organization – produces the same results. Thus, Table 7 shows the results obtained to analyse the impact of the remaining parameters in the defined KPI.

Table 7: Simulation results for strategy B

Strategy	Time gap	Error	Different types of containers	Total time	Number of stops	Unused channel	Depth	Global Classification
B	0	0	6	241,4	1,79	256	1,257	46%
B	24	0	6	227,9	1,76	197	1,122	53%
B	48	0	6	222,4	1,76	181	1,069	57%
B	72	0	6	219,0	1,77	171	1,036	59%
B	96	0	6	216,8	1,80	162	1,012	60%
B	120	0	6	215,8	1,78	162	1,004	61%
B	144	0	6	215,7	1,78	162	1,002	61%
B	150	0	6	215,4	1,78	162	1,000	62%

When analysing these results, the first thing that should be noted is that when a time gap of 0 was modelled, the depth value was 1,257, which corresponds to an average of 185.3 containers collected from the non-first position of the

channels, per milk run. This is a very high value that affected the remaining KPI of this scenario and is something that is not allowed by the standards of the organization. It also indicates that the rates on which containers are sent to the supermarket and required by the production lines, are not fully-synchronized, since many containers are consumed before other that have been sent after to the supermarket.

Another aspect to note is that by increasing the time gap, the only KPI that seems to be unaffected by it, is the number of stops per milk run. On the other hand, as the gap is increased, the depth KPI and the number of unused channels decrease. The reason for this is that higher intervals result in reduction of the number of different types of containers allowed per channel and thus in a higher probability of having all containers collected from the first position. Hence, the space occupied in the supermarket increases. In this sense, it can be concluded that in this strategy, a balance between pros and cons needs to be pondered. Whether the goal of a company is to reduce the size of the supermarket without considering depth values, or simply to reduce the average picking time without considering the size of the supermarket, the choice of the time gap to be used depends on what the company wants. However, in this case it is mandatory to achieve a depth value considerably near to 1. In this context, it could be concluded the last scenario would be the one that best fits the requirements of the organization. Notwithstanding, this scenario was modelled with a gap so high that it became equivalent to strategy A2, as the KPI values registered and the global classification for these scenarios confirm. In the same way, scenarios modelled with gaps from 4 to 6 days represent the same situation. Thus, scenario modelled with 3 days of time gap can be selected as the best fitting scenario for strategy B, although, on average, in this scenario, milkruns collected 31,3 containers from the non-first position of the channel, during the modelled week, which is still a high value nevertheless. Table 8 shows the obtained results for the strategy C.

Table 8: Simulation results for strategy C

Strategy	Time gap	Error	Different types of containers	Total time	Number of stops	Unused channels	Depth	Global Classification
C	0	0	1	215,8	1,78	355	1	80%
C	0	0	2	214,8	1,71	431	1	87%
C	0	0	3	214,7	1,70	458	1	90%
C	0	0	4	214,9	1,72	466	1	91%
C	0	0	5	214,9	1,72	471	1	91%
C	0	0	6	214,9	1,72	473	1	91%
C	24	0	6	215,2	1,73	401	1	84%
C	48	0	6	215,9	1,78	357	1	80%
C	0	24	6	246,5	1,75	461	1,307	60%
C	24	24	6	218,0	1,82	369	1,015	79%
C	48	24	6	216,1	1,80	360	1	80%
C	0	48	6	251,0	1,80	441	1,345	55%
C	24	48	6	225,4	1,84	370	1,084	73%
C	48	48	6	219,4	1,84	336	1,025	75%
C	72	48	6	217,0	1,81	349	1,006	78%
C	96	48	6	216,1	1,79	351	1	79%
C	0	72	6	253,5	1,82	424	1,365	51%
C	24	72	6	231,8	1,83	364	1,150	66%
C	48	72	6	223,1	1,83	337	1,065	71%
C	72	72	6	219,7	1,83	337	1,031	75%
C	96	72	6	218,1	1,84	346	1,013	77%
C	120	72	6	216,2	1,79	357	1,002	80%
C	144	72	6	215,7	1,77	356	1	80%

When analysing these results, it can be seen that the gap between containers of different types stored in the same channel – only for the scenarios without prediction errors - mainly affects the number of unused channels. The consequence of this fact was already address in the previous storage policy. When analysing the impact of the property that defines the prediction errors, the data showed that, when the errors were lower than the interval gaps, the depth values were always equal to 1 and the average picking time decreased. An organization with an IS that can accurately predict the consumption date of their containers could use the scenario that obtained the best global classification – 91%. In comparison to the scenario currently in practice at the organization, this corresponds to a reduction of roughly 30 seconds per trip (12% of reduction) on the average time per picking shift, 2 stops per picking shift and per milkrun (reduction of roughly 55%) and a reduction of around 69% in the supermarket size (average difference of about 325 channels). All these gains were achieved by maintaining the rule stating that containers should be collected from the first position of any channel. Table 9 shows the results obtained for strategy C2.

Table 9: Simulation results for strategy C2

Strategy	Time gap	Error of containers	Different types	Total time	Number of stops	Unused channels	Depth	Global Classification
C2	0	0	1	216,9	1,91	158	1	60%
C2	0	0	2	220,2	2,00	262	1	69%
C2	0	0	3	219,8	2,07	279	1	70%
C2	0	0	4	219,4	2,05	288	1	71%
C2	0	0	5	221,0	2,05	283	1	71%
C2	0	0	6	220,3	2,07	284	1	71%
C2	24	0	6	221,5	2,14	120	1	55%
C2	48	0	6	220,7	2,08	139	1	57%
C2	0	24	6	236,6	2,12	266	1,150	54%
C2	24	24	6	222,9	2,12	162	1,014	58%
C2	48	24	6	220,3	2,01	145	1	58%
C2	0	48	6	247,9	2,16	252	1,260	42%
C2	24	48	6	227,7	2,14	202	1,067	56%
C2	48	48	6	222,3	2,10	175	1,012	59%
C2	72	48	6	219,9	2,05	165	1,001	60%
C2	96	48	6	219,4	2,02	164	1	60%
C2	0	72	6	256,7	2,15	248	1,343	34%
C2	24	72	6	236,1	2,12	216	1,147	50%
C2	48	72	6	226,8	2,10	187	1,064	55%
C2	72	72	6	223,5	2,11	188	1,027	59%
C2	96	72	6	219,8	2,03	167	1,005	60%
C2	120	72	6	220,0	2,02	178	1,0002	61%
C2	144	72	6	218,5	1,99	180	1	62%
C2	0	96	6	254,5	2,16	247	1,319	36%
C2	24	96	6	240,7	2,16	210	1,179	46%
C2	48	96	6	233,3	2,16	208	1,119	52%
C2	72	96	6	228,4	2,07	183	1,078	54%
C2	96	96	6	223,5	2,04	188	1,046	58%
C2	120	96	6	221,5	2,08	177	1,012	60%
C2	144	96	6	219,1	2,00	179	1,002	61%
C2	168	96	6	218,8	2,03	167	1,0002	60%
C2	192	96	6	219,2	2,01	172	1	61%

When analysing these results, it can be seen that the main conclusions from analysing the strategy C can also be observed in this strategy. Though, for strategy C2, its advantages are slightly more attenuated in comparison to the previous one. In this sense, the highest classification for this strategy was 71%, corresponding to a scenario with 0 hours for both the error and the gaps properties. By comparing the KPI values of this scenario with the values from scenario 4, a difference of more than 23 seconds per picking shift (reduction of 10%), a difference of 47% in the average number of stops (average of 2 stops less) and a difference of 48% in the number of unused channels (reduction of 136 channels) can be verified. These advantages were achieved by maintaining the depth values at 1.

5.2 Discussion

The strategies that achieved the best global classification were C and C2, i.e., ordering the containers per consumption date, regardless of the type of containers stored per channel. However, this can only be used with an IS capable of managing the flows of containers in and out of the supermarket. As it was seen, these strategies also do not achieve good results when the prediction errors are higher than the time gaps.

On the other hand, strategy B could only achieve depth values equal to 1 when the interval gap was so high that this strategy was equivalent to strategy A2. The best scenario of this strategy that was not equivalent to strategy A2 registered an average of 31,3 containers picked from the non-first position of the channels, which should have been avoided. Therefore, the authors would not advise this strategy. However, if the organization is capable of better synchronize the requests from the production lines with the dates on which containers are sent to the supermarket, this strategy should be assessed again.

There are many scenarios that comply with one of the requirements of the organization – to maintain the FIFO order on the supermarket. However, only the scenarios from the strategy C that could comply with the mentioned restriction, could also comply with the goal of reducing the space of the warehouse to a third. Thus, it is seen that this goal is highly dependent on the capacity of the IS being implemented of being able to maintain strategy C.

Regardless of the IS, the best proposition that the simulation model results indicated was that the supermarket should be divided in areas for each milkrun – strategy A2 - which was not being done at the organization in question. This still resulted in a reduction of the supermarket size in 18 channels.

6 CONCLUSIONS

Warehouses are critical to a wide range of customer service activities and yet, they are also quite significant from a cost perspective. One of the goals of the Bosch Production System (BPS), implemented at Bosch, is to provide “the basis for continuous improvements in quality, costs, and supply performance” (Bosch, 2014). Thus, the opportunity to develop a micro **simulation** model in **Simio** that could help the Bosch Car Multimedia arose.

Particularly, this tool needs to be able to design several layouts of the supermarket and use them to test different strategies for their **picking system**.

The purpose of this paper is twofold. First it presents a simulation model generator able of automatically building warehouse simulation models. Both the generator and the automatically built simulation model have been validated. Secondly, this paper intends to use the generated simulation model in a case study where it is intended to analyse different storage strategies for the warehouse of the organization of the case study.

With the developed model, the organization of the case study, benefits by being able of using it to model different types of warehouses, not only supermarkets. Since the simulation model can be automatically created, the user only needs to insert the data correspondent to the layout and generate the intended simulation model. Afterwards, the model can be used to test different scenarios for the warehouse, but also other layouts, for instance. Researchers may also benefit from the tool by using it to simulate different types of warehouses. The quality of the animation is quite appealing, as the several figures illustrated throughout the document witness. Furthermore, other authors may also benefit with the experience reported in this paper, namely regarding the utilization of the API of Simio, since a lack of information available concerning this issue can be found available. In fact, few information is present on private groups of the simulation tool, such as the facebook group, or the forum for registered members. Moreover, the amount of case studies available that work with the tool in question is not considerable, therefore this paper also contributes in that regard.

As simulation model generation is a growing trend, especially in the context of industry 4.0, this work is a continuous one, despite the fact the work on the reported case study was successfully finished. As such, a future research agenda needs to consider the usage of the proposed tool to be adaptable to many different warehouse systems. Another future research item would be to completely integrate the developed generator, for instance in the IS of the organization. Furthermore, with this integration the excel spreadsheet could be replaced with a direct interface on the IS. Lastly, the possibility of extending the generator to more relevant systems of the organization in question is already being pondered. Currently, this is being done for the production lines and, in the near-future, it is expected that these findings can be published. The goal is to have a broad generator of the most relevant systems of the organization.

References

- ABIDI, M.-A., LYONNET, B., CHEVAILLIER, P. & TOSCANO, R. 2016. Contribution of Virtual Reality for Lines Production's Simulation in a Lean Manufacturing Environment. *International Journal of Computer Theory and Engineering*, 8, 182.
- ALNAHHAL, M., RIDWAN, A. & NOCHE, B. In-plant milk run decision problems. *Logistics and Operations Management (GOL)*, 2014 International Conference on, 2014. IEEE, 85-92.
- BAKER, P. & CANESSA, M. 2009. Warehouse design: A structured approach. *European Journal of Operational Research*, 193, 425-436.
- BARTHOLDI, J. J. & HACKMAN, S. T. 2008. *Warehouse & Distribution Science: Release 0.89*, The Supply Chain and Logistics Institute.
- BOSCH. 2014. consulted online at: <http://www.bosch.com/en/com/home/homepage.html> [Online]. [Accessed].
- BOTTANI, E., CECCONI, M., VIGNALI, G. & MONTANARI, R. 2012. Optimisation of storage allocation in order picking operations through a genetic algorithm. *International Journal of Logistics Research and Applications*, 15, 127-146.
- BOTTANI, E., FERRETTI, G., MONTANARI, R. & RINALDI, M. 2014. Analysis and optimisation of inventory management policies for perishable food products: a simulation study. *International Journal of Simulation and Process Modelling* 11, 9, 16-32.
- BOTTANI, E., MONTANARI, R., RINALDI, M. & VIGNALI, G. 2015. Intelligent algorithms for warehouse management. *Intelligent Techniques in Engineering Management*. Springer.
- COSTA, B., DIAS, L. S., OLIVEIRA, J. A. & PEREIRA, G. 2008. Simulation as a tool for planning a material delivery system to manufacturing lines. *Engineering Management Conference, 2008. IEMC Europe 2008. IEEE International*.
- DIAS, L., PEREIRA, G. & RODRIGUES, G. 2007. A Shortlist of the Most Popular Discrete Simulation Tools. *Simulation News Europe*, 17, 33-36.
- DIAS, L., PEREIRA, G., VIK, P. & OLIVEIRA, J. A. 2011. Discrete simulation tools ranking: a commercial software packages comparison based on popularity. *Industrial Simulation Conference*. Venice, Italy, 6-8 June: Eurosis.
- DIAS, L. M. S., VIEIRA, A. A. C., PEREIRA, G. A. B. & 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*.
- FOWLER, J. W. & ROSE, O. 2004. Grand challenges in modeling and simulation of complex manufacturing systems. *Simulation*, 80, 469-476.
- GU, J., GOETSCHALCKX, M. & MCGINNIS, L. F. 2010. Research on warehouse design and performance evaluation: A comprehensive review. *European Journal of Operational Research*, 203, 539-549.
- GUASCH, A., PIERA, M. A. & FIGUERAS, J. 2011. Automatic warehouse modelling and simulation. *International Journal of Simulation and Process Modelling*, 6, 288-296.
- HARASZKÓ, C. & NÉMETH, I. 2015. DES Configurators for Rapid Virtual Prototyping and Optimisation of Manufacturing Systems. *Periodica Polytechnica. Engineering. Mechanical Engineering*, 59, 143.
- HLUPIC, V. 2000. Simulation software: an Operational Research Society survey of academic and industrial users. *Proceedings of the 32nd Winter Simulation*

- Conference (pp. 1676-1683 vol. 2). Society for Computer Simulation International.
- HLUPIC, V. & PAUL, R. 1999. Guidelines for selection of manufacturing simulation software. *IIE Transactions*, 31, 21-29.
- KLENK, E., GALKA, S. & GÜNTNER, W. A. 2015. Operating Strategies for In-Plant Milk-Run Systems. *IFAC-PapersOnLine*, 48, 1882-1887.
- MONDEN, Y. 1998. Toyota Production System – an integrated approach to Just-In-Time. Institute of Industrial Engineers, Norcross, Georgia.
- MÁS, A. S., GARCÍA-SABATER, J. P., LLORCA, J. M. & MAHEUT, J. 2016. Assembly plant simulation to support decision-making in Layout Design considering safety issues. A case study. *WPOM-Working Papers on Operations Management*, 7, 64-88.
- OUEIDA, S., CHAR, P. A., KADRY, S. & IONESCU, S. 2016. Simulation Models for Enhancing the Health Care Systems. *FAIMA Business & Management Journal*, 4, 5.
- PEGDEN, C. D. 2007. Simio: A new simulation system based on intelligent objects. Proceedings of the 39th Winter Simulation Conference: 40 years! The best is yet to come, pp. 2293-2300, IEEE Press.
- PEGDEN, C. D. & STURROCK, D. T. 2008. Introduction to Simio. 2008 Winter Simulation Conference, pp 29-38, IEEE.
- PENKER, A., BARBU, M. & GRONALD, M. 2007. Bottleneck analysis in MDF-production by means of discrete event simulation. *International journal of simulation modelling*, 6, 49-57.
- POPOVICS, G., PFEIFFER, A. & MONOSTORI, L. 2016. Generic data structure and validation methodology for simulation of manufacturing systems. *International Journal of Computer Integrated Manufacturing*, 29, 1272-1286.
- ROBINSON, S. 2005. Discrete-event simulation: from the pioneers to the present, what next? *Journal of the Operational Research Society*, 56, 619-629.
- SILVA, C., FERREIRA, L. M., THÜRER, M. & STEVENSON, M. 2016. Improving the logistics of a constant order-cycle kanban system. *Production Planning & Control*, 27, 650-659.
- STURROCK, D. T. & PEGDEN, C. D. 2010. Recent innovations in Simio. *Proceedings - Winter Simulation Conference*, pp. 52-62. Proceedings - Winter Simulation Conference, pp. 52-62.
- TAJINI, R., ELHAQ, S. L. & RACHID, A. 2014. Modelling methodology for the simulation of the manufacturing systems. *International Journal of Simulation and Process Modelling*, 9, 285-305.
- VIEIRA, A., DIAS, L., PEREIRA, G. & OLIVEIRA, J. 2014. Comparison of Simio and Arena simulation tools. *ISC*. University of Skovde, Skovde, Sweden.
- VIEIRA, A., DIAS, L. S., PEREIRA, A. B. P., OLIVEIRA, J. A., CARVALHO, M. S. & MARTINS, P. 2016. Automatic Simulation Models Generation of Warehouses with Milkruns and Pickers. *The European Modelling & Simulation Symposium - EMSS (part of I3M - 13th international multidisciplinary modeling & simulation multiconference)*. Winner of the best paper award of the conference. Larnaca, Cyprus.
- VIEIRA, A., DIAS, L. S., PEREIRA, G. A. B., OLIVEIRA, J. A., CARVALHO, M. S. & P., M. 2015. Using Simio to Automatically Create 3D Warehouses and Compare Different Storage Strategies. *Faculty of Mechanical Engineering Transactions*, 43, 335-343.
- WOMACK, J. P. & JONES, D. T. 1996. *Lean Thinking*. Siman & Schuster, New York, USA.
- WOMACK, J. P., JONES, D. T. & ROOS, D. 1990. *The machine that changes the world*. Rawson Associates, NY