Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

BUILT-IN LEAN MANAGEMENT TOOLS IN SIMULATION MODELING

Pawel Pawlewski

Faculty of Engineering Management
Poznan University of Technology
Ul. Strzelecka 11
Poznan, 60-965, POLAND

ABSTRACT

The paper presents an approach for integrating Lean management tools, value-added analysis and Yamazumi charts, with simulation modeling in order to analyze the behavior of certain types of production systems. This is accomplished through the development of a set of high-level, operational, task-oriented instructions that are based on production engineering means and language. This is in contrast to the more traditional simulation modeling approach that uses lower-level, logic-oriented instructions that are based on the means and language of computer programming. The paper defines the approach and its implementation and its application is illustrated through a simple manufacturing cell example.

1 INTRODUCTION

This paper presents a modeling approach that has been developed as the result of research and several applied simulation projects that have been carried out in the automotive industry over the past several years. The projects focused on material flow, line balancing, layout, and intralogistics design and redesign using simulation and Lean tools. The projects have led to a means to integrate Lean management tools with simulation modeling. Many articles are found in the literature that describe how specific Lean tools have complemented or been tested with simulation, including for example Value Stream Mapping (VSM) and Just in Time. Over the last decade, more and more authors have begun to identify the benefits that can be gained by combining Lean concepts and simulation; however, this is often not discussed in the literature (Robinson et al. 2012; Uriarte et al. 2015).

The main objective of this research is to define and develop an approach for modeling the behavior of certain types of production systems by using high-level, operational, task-oriented instructions as opposed to lower-level logic-oriented instructions that are more akin to programming. The higher-level approach is more intuitive to engineers and others who design and manage operations systems. Such an approach also should complement Lean management concepts and support such related tools as value-added analyses and Yamazumi charts.

The main contribution of this paper is the definition, development, and implementation of a task-based approach for representing production operations so that they can be modeled and simulated by factory engineers and not simulation experts. The main focus is on providing a means for describing the behaviors of a class of objects referred to as operators, which include: human workers, robots, forklifts, AGVs (Automated Guided Vehicle), etc.

The paper is divided into six sections. Following an introduction, the second section contains a literature review on Lean Management tools – especially on Value Stream Mapping and simulation. In this section the relationship between VSM, a high level approach, and simulation, a low level approach, are discussed. The third section describes the motivation for the research and provides a problem statement. The proposed high level, operational, task-oriented instructions are defined in Section 4 and its implementation is presented in Section 5. The last section presents conclusions and outlines further research.

2 LITERATURE REVIEW

As mentioned earlier, much has been written about how specific lean tools have been complemented by, or tested with, simulation. Lean is a management philosophy that focuses on customer satisfaction and the elimination of non-value-added activities via continuous improvement. It is defined as a concept that effectively eliminates, or at least mitigates, systems' waste (Womack and Jones 2003). Lean provides a set of principles and tools that are used to gain operational efficiency, reduce process waste, and increase productivity. While the majority of the studies focus on a single aspect of lean, a successful application oftentimes needs to focus on multiple aspects, such as Value Stream Mapping (VSM), Line Balancing, Single Minute Exchange of Dies (SMED), Visual Management, Production Levelling, 5Ss, LLD, etc. (Wilson 2010).

Value Stream Mapping is a key method within Lean for identifying material and information flow, as well as value-added and non-value-added activities (Rother and Shook 1999). VMS has some limitations, primarily due to its static nature; i.e., the uncertainty, variability, and the dynamic nature of a production system cannot be represented (Abdulmalek and Rajgopal 2007; Marvel and Standridge 2009; Jia 2010). Another limitation is that only the flow of one product can be analyzed at a time (Solding and Gullander 2009; Atieh et al. 2016). Also, it does not represent the complete reality, just a two-dimensional simplification ("paper and pencil" map) (Solding and Gullander 2009; Helleno et al. 2015). Therefore, VSM is not effective for mapping complex systems and the interactions between its components, (Marvel and Standridge 2009; Jia 2010; McDonald et al. 2002). Generally, VSM is used to improve existing production system. In order to optimally design lean manufacturing processes, some Operations Research techniques are used (Dolgui 2007), such as advanced line-balancing methods introduced by (Baybars 1986; Scholl 1999).

Simulation has been recognized by many authors as a good complement to VSM and a good way to overcome the aforementioned limitations. By simulation, we mean discrete-event simulation (DES) - one of the most-applied simulation techniques for studying and improving manufacturing systems (Negahban and Smith 2014).Some example case studies where simulation and VSM have been employed together include: (McDonald et al. 2002; Marvel and Standridge 2009; Solding and Gullander 2009; Gurumurthy and Kodali 2011; Schmidtke et al. 2014; Helleno et al. 2015; Atieh et al. 2016). Some authors present tools that generate a simulation model based on a VSM (Lian and Van Landeghem 2007; Jia 2010). Such tools are now available in some commercial simulation software packages (Uriarte et al. 2015). Also, some papers describe the generation of a VSM from a simulation model (Solding and Gullander 2009).

Therefore, VSM can be viewed as a high-level approach for describing a value stream; whereas, simulation can be considered a low-level approach because of the details needed to build a simulation model. As such VSM and simulation can be considered complementary approaches.

The approach offered in this paper is more of a middle layer between high- and low-level approaches. Operations of manufacturing and logistics systems are defined, and behaviors are described, by a set of parameterized instructions that are more operationally-relevant than lower-level, more abstract, logic-based instructions inherent in most commercial, general-purpose simulation software. The operational instructions are closely aligned with basic operations terminology and Lean thinking rather than flowcharting and computer programming.

3 MOTIVATION AND PROBLEM STATEMENT

Typically, simulation models are built by selecting pre-built objects from a library, dragging them to a work area, and defining relationships (connections) between these objects according to the real system's flow. Many simulation tools use a process-driven approach, where the flows of items, e.g. parts in a manufacturing system, create demands on resources, e.g. workstations and operators, in order to complete a planned operation. While this approach works well for many applications, it does not allow resources to complete tasks which are not flow related. For example, an operator (mobile resource) might perform a set of inspections on idle equipment when not engaged in process work. This is referred to as a task-based approach, where a task executor (e.g., an operator) creates a set of tasks or activities that are independent of the flow-processing activities. Typically, such tasks require travel and the acquisition of other tools or resources. As such, the task-driven approach incorporates "intelligence" into resources by enabling them to decide what jobs to do and when. Simulation tools offer means for defining the task-executor logic. For example, in *AnyLogic* this is done through flowcharts and *Java* scripting; similarly, in *FlexSim* this is done graphically through ProcessFlow, a flowcharting type of tool, and FlexScript, a C++-based language. Other simulation software available in the market work similarly. In all of these cases, the logic is built at a low-level using programming-like instructions that are contained in libraries of hundreds of functions. This low-level approach makes it difficult and cumbersome to perform value-added and Lean analyses. Therefore, a higher-level approach is needed that is more accessible to factory engineers; i.e., closer to the operations language used by these engineers.

This notion is explained through the simple example presented in Figure 1, which shows a welding cell from factory that produces automotive parts. A few of the key objects are identified as:

- A – Storage rack for containers with parts
- B – Operator
- C – Welding table,
- D – Handcart for moving products.

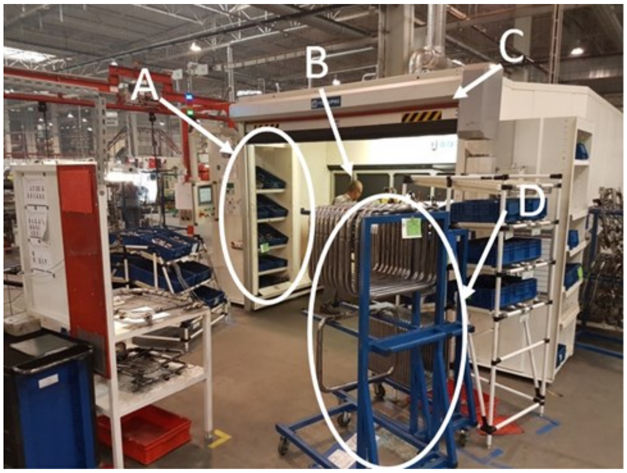

Figure 1: View of a basic automotive parts welding cell (Source: PUT).

In this cell, an operator assembles two parts that are taken from the storage rack. This operation involves the following sequence of activities:

1. Go (travel) to storage rack A where containers with parts are located
2. Check the contents of the desired containers. If there is no part, wait for a part to appear in the container (e.g., delivered by a material handler)
3. Select/pick two parts from their containers
4. Go (travel) to workstation C (welding table)
5. Place the parts on the welding table C
6. Assemble the parts (welding)
7. Select/pick the assembly from workstation C

8. Go (travel) to the location of the container for the assembly, handcart D.
9. Place the assembly in the container
10. Repeat the preceding 9 steps

The activities defined above can be classified as:

- Travel – activities 1, 4, and 8
- Check & Wait – activity 2
- Load – activities 3 and 7
- Unload – activities 5 and 9
- Assemble – activity 6
- Control – activity 10

Lean manufacturing defines eight types of waste :

- Transport: moving people, products, and information
- Inventory: storage of parts, materials, etc. prior to being needed
- Motion: bending, turning, reaching, lifting, etc.
- Waiting: delay for a needed resource, such as equipment, material, information, etc.
- Over production: producing more than is immediately required
- Over processing: using tighter tolerances or a higher grade materials than is necessary
- Defects: rework, scrap, incorrect documentation
- Skills: under utilizing capabilities, delegating tasks with inadequate training

The basis for Lean manufacturing is to focus on the process and on the analysis of added value. Activities are classified as Value-Added activities (VA) and Non-Value-Added activities (NVA). NVA activities are further divided into activities that do not add value, but are necessary to implement the process and those that are unnecessary, actually waste. The latter is referred to as NVAA (Non-Value-Added Attackable) and are real possibilities to eliminate work and shorten the time of execution.

In the example welding operation that is described above, all of the activities involve waste except Activity 6; it is a VA activity. The activities are classified in Table 1. Step 10 is a control activity; it is not considered in the value-added analysis since its time of execution is zero.

Table 1: Classification of process activities according to Lean concepts.

Type of Value	Individual activities	Group of activities
Value Added (VA)	6 (Assembly)	Assembly
Non-Value Added (NVA)	1 (Travel), 3 (Select/pick), 4 (Travel), 5 (Place), 7 (Select/pick), 8 (Travel), 9 (Place)	Traveling, Loading, Unloading,
Non-Value Added Attackable (NVAA)	2 (Check/Wait)	Checking & Waiting

In order to perform value-added analyses using simulation, activities need to be defined in this higher-level engineering-focused language.

4 SOLUTION APPROACH

In order to improve the use of simulation to support Lean analyses, model building of operations should be via a higher-level, more operational-focused, language, rather than a programming-focused language. The language should consist of operational instructions rather than programming instructions. Table 2 contains a list of such operational instructions that corresponds to the process described in Section 3.

Table 2: Operational instructions that describe the example's activities.

Process step	Description of the step (from the example)	Instruction representing the process step
1	Go (travel) to storage rack A where containers with parts are located.	Travel
2	Check the contents of the desired container. If there is no part, wait for the parts to appear in the container (e.g,, delivered by a material handler)	CheckPartInToteWait
3	Select/pick two parts from their containers	LoadFromTote 2
4	Go (travel) to workstation C (welding table)	TravelLoaded
5	Place the parts on the welding table C	Unload 2
6	Assemble the parts (welding)	Assembly 10
7	Select/pick the assembly from workstation C	Load 1
8	Go (travel) to the location of container for the assembly, handcart D	TravelLoaded
9	Place the assembly in the container	UnloadToTote 1
10	Repeat this procedure	Call 1

Of course, the instruction names by themselves are not sufficient because the operator only knows what to do, the operator does not know: where to do the task or where to move to, how many parts to take, how long to carry out the operation, etc. Therefore, information on the number of parts to be picked up and the times for completing the operation are specified through parameters of the instruction. The instruction definition in Table 2 contain the parameters for Steps 3, 5, 6, 7, and 9. For example, in Step 3 the parameter value 2 represents the number of parts to pick from the container in storage rack A; in Step 6 the parameter value 10 represents the operation time 10 seconds; etc.

The instruction parameter is also used to specify where an operation should be performed. Lean manufacturing uses the term Lean Layout (Harris et al. 2003) to define the location of key elements on a production floor. It is a bird's eye view of the entire production area or the portion of the facility that contains all of the desired elements, such as machines and workstations.

From a lean perspective, those elements are referenced on macro and micro scales. A macro-scale layout is a reflection of the arrangement of all elements throughout the production hall and is focused on obtaining continuous flow, creating linear and nested workstations, reducing distances between cooperating stations, etc. A micro-scale layout considers individual workstations or areas and contains all of the elements necessary to perform work.

In order to specify where an operation takes place, key objects are considered in a micro-scale layout. Locations are places where containers with parts and products are located. They are identified by an address or unique name. Each location may contain operator access points and logistician access points. As illustrated in Figure 2, P_xx is the address, or reference name, of a location that is a rectangular area for containers or parts. The location is defined in terms of several parameters: length, width, angles of rotation relative to the X, Y and Z axes, and height above the ground. The location is connected to a point on the ground, which is named N_xx and represents the operator's access point to the location (note it uses the same reference number xx). It is also connected to another point on the ground, which is named I_xx and represents the logistician's access point to the location (similarly, it uses the same reference number xx). P_xx, N_xx and I_xx with the same number xx form a set.

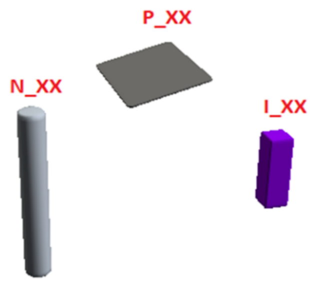

Figure 2: Example location with associated access points.

Referring to the example and the objects identified in Figure 1, consider object A, a shelf on the storage rack that contains the parts to be assembled, and object D, a handcart for storing the finished assembly. Their representation in conjunction with an operator is illustrated in Figure 3. In this case one type of operator performs the assembly and another type delivers and collects containers. The numbers refer to the elements.

1. Operator: performs tasks within the workstation; in terms of the operational language, the operator is identified by the reserved name Op_xx (where xx is the operator's sequence number).
2. Parts: components needed to manufacture a product.
3. Container: means to transport and store parts and products; it is of a certain type, has a defined shape and dimensions, weight, and color.
4. Location: place for storing containers; a rectangular area with specified width and length located at a specified height; in terms of the operational language, it is identified by the reserved name P_xx (where xx is the number of the corresponding location).
5. Operator's access point: point location on the ground where the operator accesses the location; in terms of the operational language, it is identified by the reserved name N_xx (where xx is the number of the corresponding location).
6. Logistician's access point: point location on the ground where the logistician accesses the location; in terms of the operational language, it is identified by the reserved name I_xx (where xx is the number of the corresponding location).

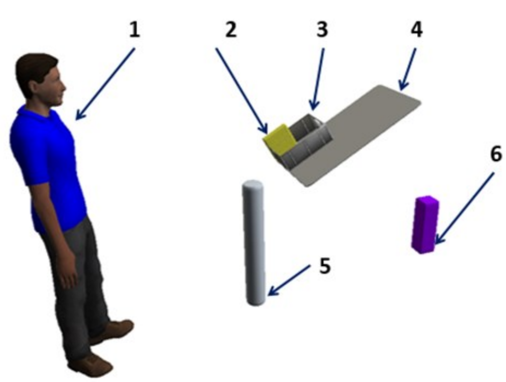

Figure 3: Example location, access points, operator, container, and parts.

The other object in the micro-scale layout from Figure 1 is assembly workstation C (welding). It is defined similarly with a reserved name Ass_xx (where xx is the sequence number of the assembly station) and point on the floor for the operator's access to the assembly station with the reserved name G_xx. In this case there is no logistician access point since only the operator interacts with the assembly station.

All of the objects from Figure 1 are represented in terms of micro-scale layout elements in Figure 4. Op_1 is object B, the operator that performs the ten steps. Ass_01 is the welding station and G_01 is the location where the operator performs the welding. P_01 is the location of the handcart for the finished assembly and N_01 is where the operator interacts with the cart. Similarly, P_02 is the location of the storage rack that contains the parts to be assembled and N_02 is where the operator download two parts.

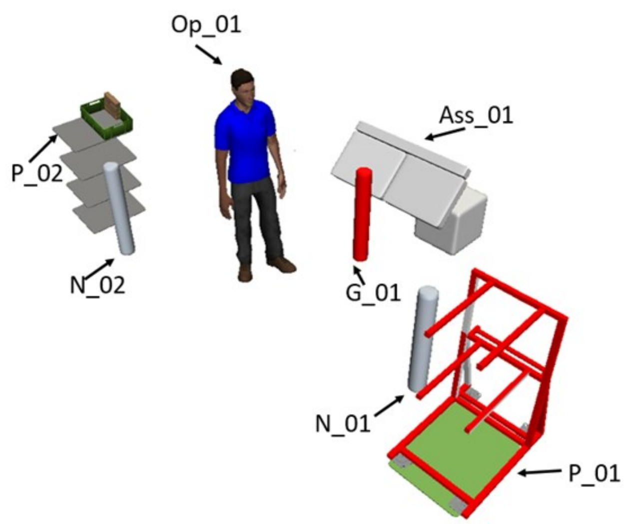

Figure 4: Model representation of the welding cell shown in Figure 1.

Each of the ten steps in Table 2 are now represented by a corresponding instruction, address, and parameter, as shown in Table 3. These describe the operator cycle Op_01.

Table 3: List of instructions for the example.

Step	Address	Instruction	Parameter
1	N_02	Travel	0
2	P_02	CheckPartInToteWait	2
3	P_02	LoadFromTote	2
4	G_01	TravelLoaded	0
5	Ass_01	Unload	2
6	Ass_01	Assembly	10
7	Ass_01	Load	1
8	N_01	TravelLoaded	0
9	P_01	UnloadToTote	1
10		Call	1

5 IMPLEMENTATION AND RESULTS

Based on the approach describe in Section 4, a comprehensive set of 64 instructions is defined, developed, and implemented. This set contains instructions for traveling, loading/unloading, checking/waiting, deciding (check and call), setting attributes, driving and controlling logistics trains, etc. A full list of instruction is found in (Pawlewski 2018).

As described in Section 3, most simulation software in the market provide libraries of general, low-level instructions for defining logic and activities (Dias et al. 2016). We chose *FlexSim* because of its ability to easily create custom objects and logic for modeling complex processes. The logic can be defined by using Process Flow, a graphically-based, logic-building tool that is a part of *FlexSim* (Beaverstock et al. 2017). The implementation of one instruction, LoadFromTote, defined in Process Flow is shown in Figure 5. The figure also shows the low-level programming-like instructions that are used to create the high-level operational instructions.

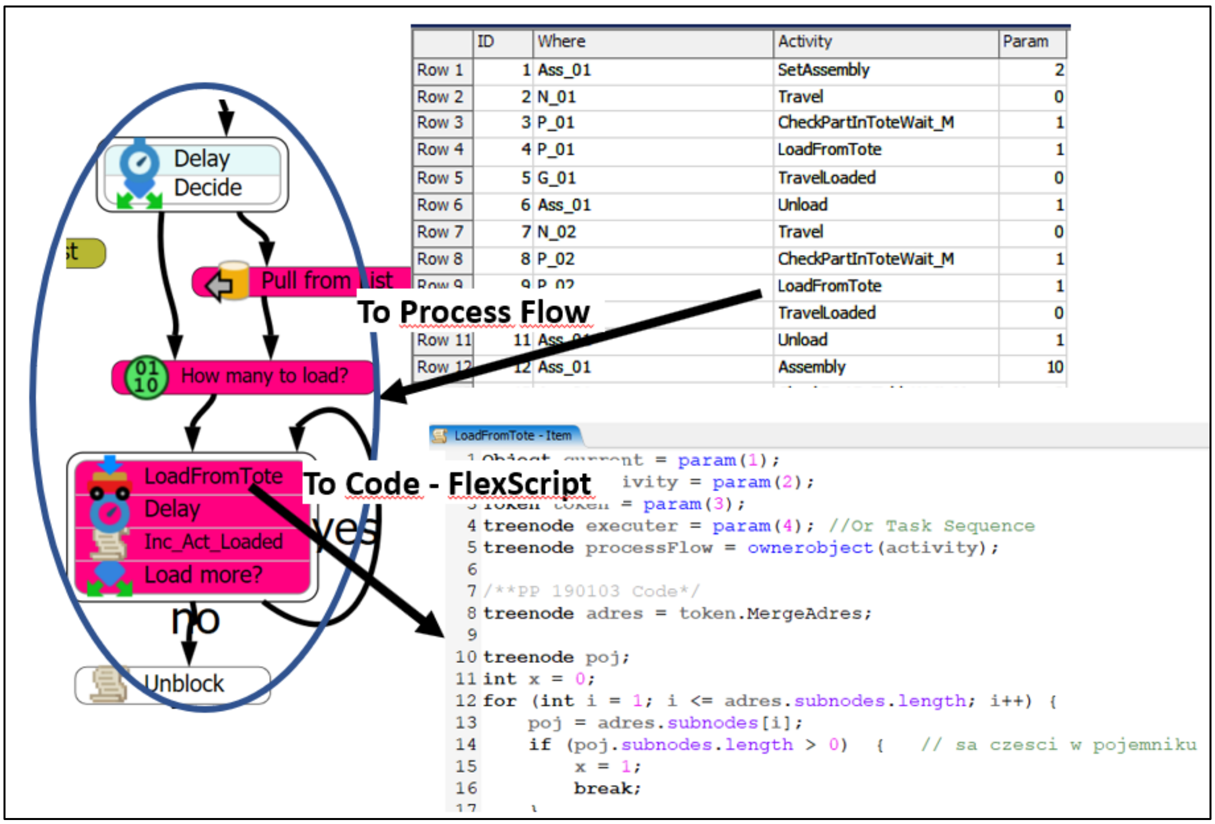

Figure 5: Implementation of an operational instruction, LoadFromTote, via *FlexSim's* logic builder ProcessFlow.

Instruction sets are saved in tables and are assigned to operators. The high-level operational instructions are transformed to low-level instructions for FlexSim to interpret and process, primarily through the use of Process Flow. When a simulation experiment is run, a database is created with records containing the names of instructions, start time of the instruction, completion time, and a value-added attribute. The database is used to create Value-Added charts. The user, e.g. engineer, can redefine value-added attributes that are assigned to instructions and then reload the database and generate charts without having to rerun the simulation experiment; this is an important benefit of the developed approach. The information flow is illustrated in Figure 6.

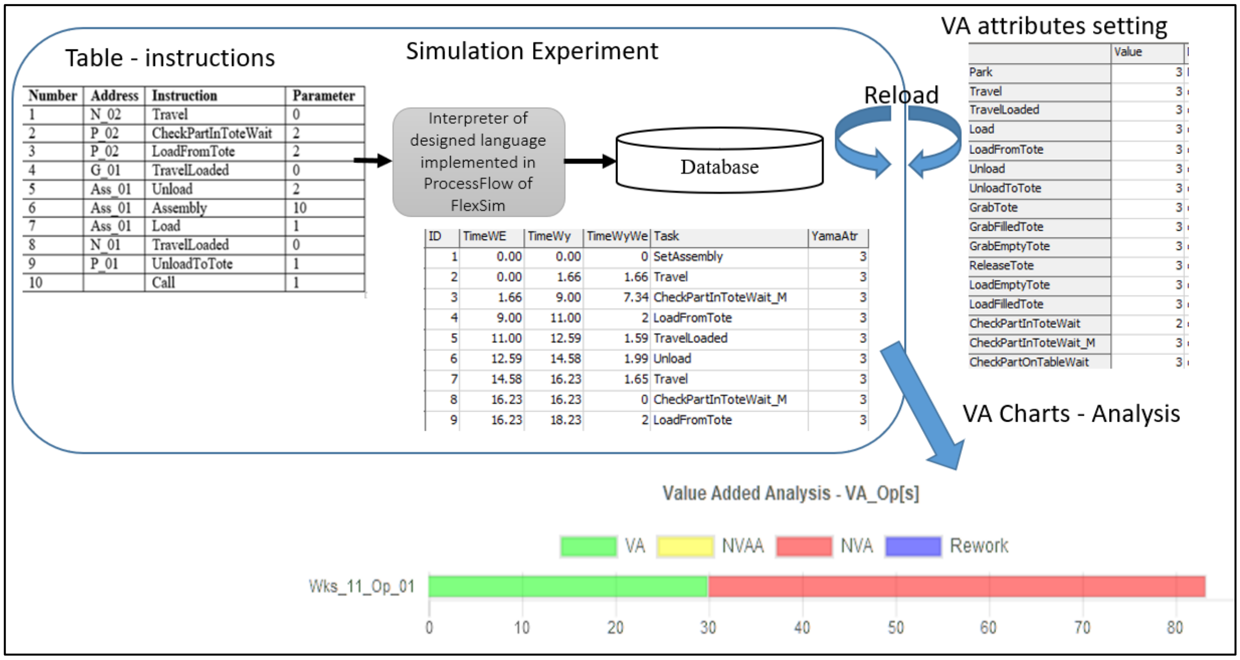

Figure 6: Information flow for generating value-added charts.

To illustrate the application of the high-level, operation-focused set of instructions, a simulation model is built based on the example described in Section 3 and using the set of instructions defined in Table 3. An additional attribute, called Rework, is introduced to better represent a customer's needs. Two experiments are performed:

- A container with parts is available in location P_02 from the start of the experiment so that the operator does not have to wait for parts. Results are shown in Figure 7.
- A container with parts is available in location P_02 nine seconds after the start of the experiments so that an operator must wait, after arriving at position N_02, until a container in location P_02 becomes available. Results are shown in Figure 8; the black arrow indicates waiting time.

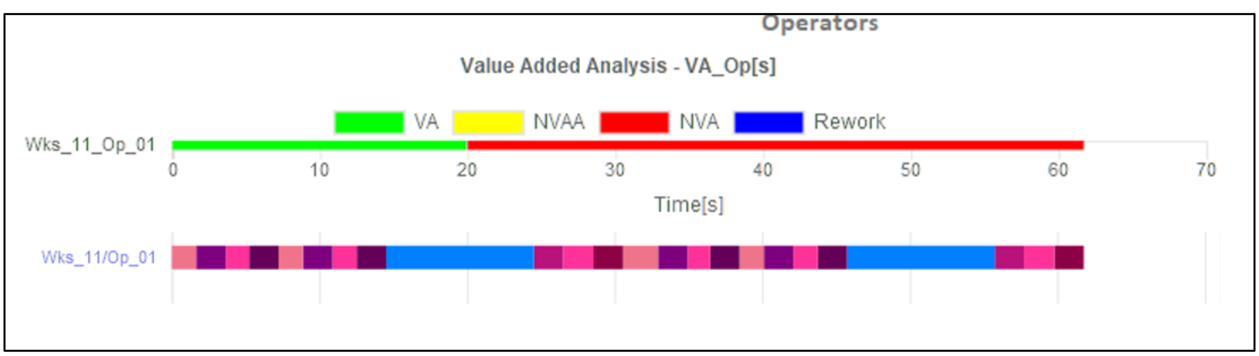

Figure 7: Value-added diagram (top chart) and process diagram (bottom chart) without waiting time.

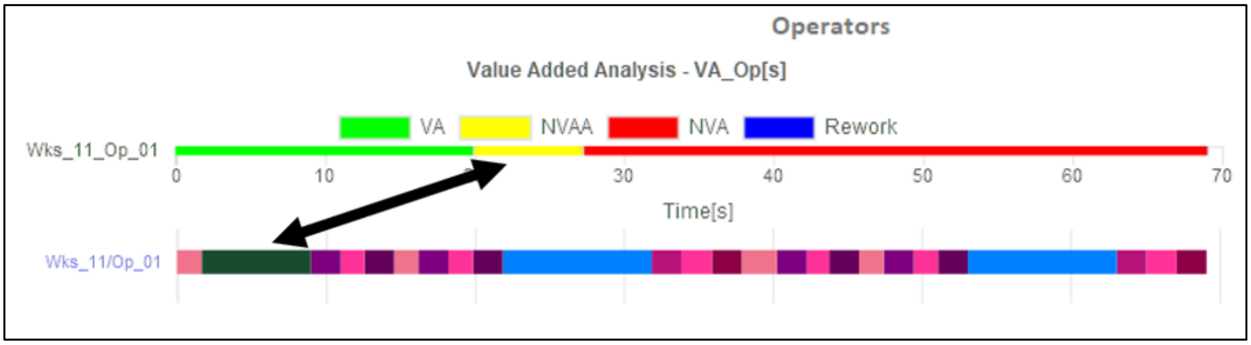

Figure 8: Value-added diagram (top chart) and process diagram (bottom chart) with waiting time, as noted by the black arrow.

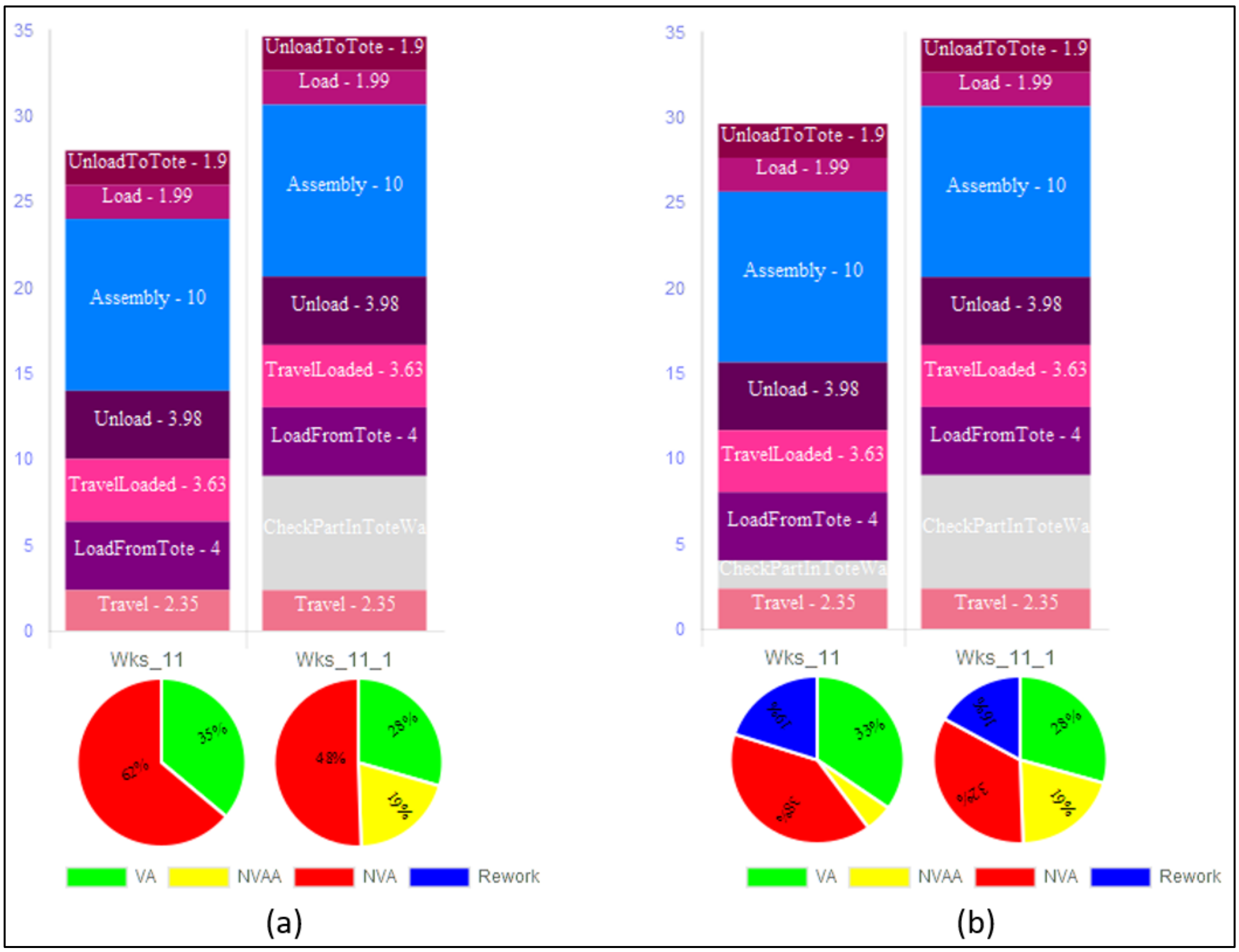

Figure 9: Yamazumi charts before (a) and after (b) the database is reloaded with new attribute settings.

Both sets of Yamazumi plots in Figure 9 compare the cases considered in Figures 7 and 8 (no waiting on the left and waiting on the right). The set of Yamazumi charts on the left in Figure 9 (a) show the base designation of activities in terms of value add and the set on the right (Figure 9 (b)) have the designations for the instructions Travel and TravelLoaded reassigned to attribute type 4 (Rework) to illustrate their effect on percentage of the process cycle time.

6 CONCLUSIONS AND FURTHER WORK

This paper defines an approach for developing simulation models using a set of high-level parameterized operational instructions to describe the behaviors of operators in many types of manufacturing and logistics situations. This is in contrast to the traditional approach that involves representing the behaviors through lower-level, more abstract, logic-based instructions. The operational instructions are closely aligned with basic operations terminology and Lean thinking rather than flowcharting and computer programming. Thus the approach will make the power of simulation analyses readily available to production engineers and allow them to perform value-added analyses using simulation-based experiments.

Now that the methodology is clearly established and tested, it will be integrated with layout and intralogistics redesign using the methodology presented in (Pawlewski 2018).

ACKNOWLEDGMENTS

This work was funded by the 11/142/SBAD/1005 project, "Endogenic Minds of Development of Enterprises," Poznan University of Technology.

REFERENCES

Abdulmalek, F. A., and J. Rajgopal. 2007. "Analyzing the Benefits of Lean Manufacturing and Value Stream Mapping Via Simulation: A Process Sector Case Study". *International Journal of Production Economics* 107 (1):223-236.

Atieh, A. M., H. Kaylani, A. Almuhtady, and O. Al-Tamimi. 2016. "A Value Stream Mapping and Simulation Hybrid Approach: Application to Glass Industry". *International Journal of Advanced Manufacturing Technology* 84 (5-8):1573-1586.

Baybars, I. 1986. "A Survey of Exact Algorithms for the Simple Assembly Line Balancing". *Management Science*, 32, 909-932.

Beaverstock, M., A. Greenwood, and W. Nordgren. 2017. "*Applied Simulation. Modeling and Analysis using Flexsim*", Flexsim Software Products, Inc., Canyon Park Technology Center, Orem, USA.

Dias, L., G. Pereiro, P. Vilk, and J. Oliveira. 2016. "Discrete Simulation Software Ranking - a Top List of the Worldwide Most Popular and Used Tools". *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick. 1060-1071. Piscataway, New Jersey: IEEE.

Dolgui, A. 2007. "Operations Research Techniques For Design and Analysis of Lean manufacturing Systems". *IFAC Proceedings Volumes* 40, Issue 8:11-19

Gurumurthy, A. and R. Kodali. 2011. "Design of Lean Manufacturing Systems Using Value Stream Mapping with Simulation a Case Study". *Journal of Manufacturing Technology Management* 22 (4):444-473.

Harris, R., Ch. Harris, and E. Wilson. 2003. *Making Materials Flow. A Lean Material-handling Guide for Operations, Production Control, and Engineering Professionals.* Cambridge, UK: Lean Enterprise Institute Inc.

Helleno, A. L., C. A. Pimentel, R. Ferro, P. F. Santos, M. C. Oliveira, and A. T. Simon. 2015. "Integrating Value Stream Mapping and Discrete Events Simulation as Decision Making Tools in Operation Management". *The International Journal of Advanced Manufacturing Technology* 80(5):1059-1066.

Jia, Y. 2010. *Simlean: A Reference Framework for Embedding Simulation in Lean Projects*. Ph.D. thesis. Sheffield Hallam University, United Kingdom.

Lian, Y. H. and H. Van Landeghem. 2007. "Analysing the Effects of Lean Manufacturing Using a Value Stream Mapping-Based Simulation Generator". *International Journal of Production Research* 45(13):3037-3058.

Marvel, J. H. and C. R. Standridge. 2009. "Simulation-Enhanced Lean Design Process". *Journal of Industrial Engineering and Management* 2(1):90-113.

McDonald, T., E. M. Van Aken, and A. F. Rentes. 2002. "Utilising Simulation to Enhance Value Stream Mapping: A Manufacturing Case Application". *International Journal of Logistics Research and Applications* 5(2):213-232.

Negahban, A. and J. S. Smith. 2014. "Simulation for Manufacturing System Design and Operation: Literature Review and Analysis". *Journal of Manufacturing Systems* 33(2):241-261.

Pawlewski, P. 2018. "Script Language to Describe Agent's Behaviors". In *Highlights of Practical Applications of Agents, Multi Agent Systems, and Complexity: The PAAMS Collection*, edited by J. Bajo, J. M. Corchado, E. M. Navarro Martinez, E. O. Icedo, P. Mathieu, P. Hoffa-Dabrowska, W. delVal, S. Giropux, a. j. m. Castro, N. Sanchez-Pi, V. Julien, R. A. Silveira, A. Fernandez, R. Unland, R. Fuentoes-Fernandez, 137–148. Cham, Switzerland: Springer.

Pawlewski, P. 2018b. "Methodology for Layout and Intralogistics Redesign Using Simulation". *Proceedings of the 2018 Winter Simulation Conference,* edited by M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3193-3204. Piscataway, New Jersey: IEEE.

Robinson, S., Z. J. Radnor, N. Burgess, and C. Worthington. 2012. "SimLean: Utilising Simulation in the Implementation of Lean in Healthcare." *European Journal of Operational Research* 219(1):188-197.

Rother, M., and J. Shook. 1999. *Learning to See: Value Stream Mapping to Add Value and Eliminate Muda*. Brooklin, MA.: The Lean Enterprise Institute, Inc.

Sabadka, D., V. Molnár, G. Fedorko, and T. Jachowicz. 2017. "Optimization of Production Processes Using the Yamazumi Method". *Advances in Science and Technology Research Journal* 11(4):175–182.

Schmidtke, D., U. Heiser, and O. Hinrichsen. 2014. "A Simulation-Enhanced Value Stream Mapping Approach for Optimisation of Complex Production Environments". *International Journal of Production Research* 52(20):6146-6160.

Scholl A. and R. Klein. 1998. "Balancing Assembly Lines Effectively: A Computational Comparison". *European Journal of Operational Research* 114, Issue 1 :51-60.

Solding, P., and P. Gullander. 2009. "Concepts for Simulation Based Value Stream Mapping". *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 2231-2237. Piscataway, New Jersey: IEEE.

Uriarte, A. G., M. U. Moris, A. H. C. Ng, and J. Oscarson. 2015. "Lean, Simulation And Optimization: A Win-Win Combination", *Proceedings of the 2015 Winter Simulation Conference* edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 2227- 2238. Piscataway, New Jersey: IEEE.

Wilson, L. 2010. *How to Implement Lean Manufacturing*. New York, NY: McGraw-Hill.

Womack, J. P. and D. T. Jones. 2003. "*Lean Thinking: Banish Waste and Create Wealth in Your Corporation*." New York, NY: Free Press, Simon & Schuster, Inc.

AUTHOR BIOGRAPHIES

PAWEL PAWLEWSKI is Professor of the Faculty of Engineering Management at Poznan University of Technology in Poland. He is the Managing Director of SOCILAPP - Simulation and Optimization Center in Logistic and Production Processes. His research interests include discrete-event simulation, agent-based simulation, and their applications: service systems, transportation networks, and manufacturing. He has published over 130 papers. He is member of the board of Atres Intralogistics. His e-mail address is pawel.pawlewski@put.poznan.pl.