Event Triggered Simulation of Push and Pull Processes

Stefan Haag, Lara Zakfeld, Carlo Simon and Christian Reuter

Hochschule Worms
Erenburgerstr. 19, 67549 Worms, Germany
Email: {haag, lara.zakfeld, simon, reuter}@hs-worms.de

***Abstract*—The development of conceptual models relies on a proper modeling language, the existence of a beneficial tool, and knowledge about modeling techniques. We use Petri nets as our preferred modeling language for decades, because Petri net models are illustrative and can be executed or simulated, respectively. However, since most of the Petri net tools that have been developed in the past are no longer executable, we developed a novel, Web-based Petri net modeling and simulation environment called Process-Simulation.Center (P-S.C). It facilitates the definition of individual data objects as tokens such that data-driven process simulations can be conducted within the tool. In the process, we realized the absence of a general modeling technique for the definition of such models. Generally, two different approaches exist: a clock-pulse simulation where the state of the modeled system is simulated for each point in time and an event triggered simulation that calculates new states only for those moments the system state changes. Using teaching in a logistics laboratory as an example, this paper demonstrates how event triggered simulation models can be developed and how they have to be interpreted. Concretely, the consequences of switching logistics processes from push to pull principles are considered concerning the storage costs.**

***Keywords*–Conceptual models of timed dynamic systems; Simulation; Petri nets; Logistics; Teaching.**

I. Introduction

Reducing costs while at the same time increasing the production's flexibility is a combined goal for manufacturers. Beside an investment in better and faster machines, rethinking production strategies and processes is also feasible and potentially cheaper. Changing the production from push to pull is one option and its advantages have been demonstrated in many production lines. Nonetheless, push strategies are still widely in practical use. What is the reason for this? We assume that producers are uncertain about the consequences of such changes. In this case, conceptual and simulatable models of the current and the intended production lines could objectify decisions on the reorganization of the production.

This paper shows the effect of turning a push into a pull production using the example of a logistics training laboratory at the University of Applied Sciences Worms. For this, two different models of Petri nets with individual tokens have been developed and simulated: a clock-pulse simulation that – like a movie – demonstrates the behavior in every second, and an event triggered simulation that is the objective of this paper. Such a simulation, however, needs a powerful Petri net modeling and simulation tool that can handle individual tokens and that can also be used in a distributed environment. We chose to use the novel, Web-based Process-Simulation.Center (P-S.C) developed on our own, which is free to use for academic users.

Part of the research design for this tool is its use in various environments like production, logistics, trade and teaching in order to develop and understand different Petri net modeling techniques. For the concrete example, the P-S.C helps learners to predict the behavior of the processes to be reorganized.

Following this introduction, in Section II, a short explanation of the design science research method is given. Then, the logistics training laboratory for processes is explained as a sample application in Section III. Section IV provides an overview of push, pull, and kanban as well as possibilities for the modeling and simulation of timed dynamic systems with Petri nets and other modeling approaches. Also, some information about the P-S.C is given. Afterwards, in Section V, the utilized modeling approach and its push and pull based implementations are introduced. Lastly, in Section VI a conclusion both on the simulation results and the modeling approach is presented. The paper closes with an overview of planned and possible future work.

II. Research Method

Research on new and comparison of different modeling techniques for Petri nets with individual tokens relies on the existence of proper modeling and simulation tools. For this, we work on the P-S.C for several years following the guidelines for design science research as per [1]. Application to a simulative learning environment is one more step in this process. A brief overview of conducted research is as follows:

Design as an Artifact: The P-S.C is a Web-based specification and simulation software for processes encompassing both user defined and primitive data types (including date and time types), organizational structures and process maps. Process execution can be controlled by business relevant data that is linked by a data interface. Also, an interface can be used to control sensors and actors connected to a Raspberry Pi if installed on such a device.

Problem Relevance: A simulation permits the extension of real-world experiences in a learning environment. It helps to overcome typical limitations concerning time, resources, space, and people. The simulation environment, though, must be generic enough to assure the intended learning success. From a conceptual modeling perspective, it must be determined if all these aspects can be expressed and simulated due to a formal, semantic base.

Design Evaluation: The P-S.C has already been used by companies in logistics and trade. Students of an integrated logistics degree program developed a simulation model for the reorganization of a returns process [2]. The tool is also used for problem-based and research-oriented learning in bachelor and master degree programs [3].

ISBN: 978-1-61208-831-0

Research Contribution: The P-S.C is a practical application on the theoretical basis of high-level Petri nets combined with views on organizational structures, process maps and data types. The tool offers a novel user experience and provides new insights in Petri net based modeling techniques. This is important since as an abstract concept Petri nets do not force specific modeling approaches such as flow diagrams, value stream diagrams or other pictorial modeling approaches.

Research Rigor: The benefits of a simulation approach in opposite to pure visual methods is evaluated in mentioned bachelor and master courses as well as in cooperation with partner companies of integrated degree programs.

Design as a Search Process: The presented prototype is the latest in a series that starts from the initial implementation of the underlying principles and ends in a productive system. Each implementation step has been evaluated and published (for instance, [4]-[6]).

Communication of Research: The results achieved so far are relevant for both research and practice. They are presented on pertinent conferences but also, more eidetic, for practitioners in advanced training programs.

III. A Simulation Laboratory for Processes in Logistics

The so-called *box game* has been developed at the University of Applied Sciences Worms to teach students in logistics. Despite its obvious simplicity, it effectively demonstrates different kinds of problems and processes that have a high impact for practical applications. We chose it as the modeling and simulation objective in order to enrich the learning experience of the students.

The concrete example is a simple construction process: students assemble big and small boxes, put the smaller into the bigger ones, and eventually check the quality. This production is conducted following varying strategies (push and pull) in order to observe influences on organizational issues, production time, and storage costs of these strategies as well as consequences when they are changed. Learners taking part in the game gain first-hand experience of different work situations and can recognize several types of waste (called *muda*) such as overproduction, waiting, and motion, but also the transformation of waste types. Discussing the shared experiences is a major part of the learning success. A complete simulation run of the box game lasts about two to three hours.

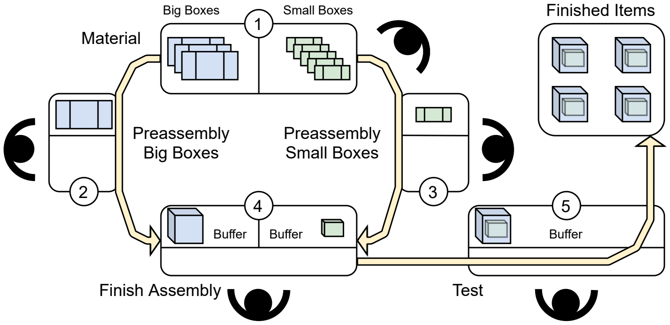

Figure 1. Layout Design of the *Box Game*

Despite the simplicity of the used material and the low level of technical requirements, the *box game* is easily transferable to assembly workstations in general and has a high practical impact. It focuses on the strategic principles and leaves out problems of a mechanical production such as shift patterns, changeover times and multiple machine set-ups.

Figure 1 shows the spatial organization of the *box game* in the learning laboratory: five worktables are arranged in a suitable location and standard positions like interim storages are marked with adhesive tape. As can be seen, the setting can also be built up in locations such as conference rooms, training rooms, or even canteens.

The following activities have to be conducted:

Storage (S): Deliver the boxes' building sets.

Preassembly big boxes (PBB): Fold the box, close the lid, and pass the box to the next processing step.

Preassembly small boxes (PSB): Fold the box, close the lid, and pass the box to the next processing step.

Finish assembly (FA): Open big box, insert small one, label small box with a package note (a simple post-it), close and seal big box lid (with adhesive tape), cut of tape with scissors if necessary, and pass the box to the test step.

Test (QA): Shake the sealed box for an acoustic quality check, apply a red dot to the upper left corner of the box to indicate a pass, and place the box in the storage.

The following players can partake in the game:

5 employees who will occupy the work stations.
3 timekeepers who record the processing times.
1 observer who records the inventory in the system.
1 observer who records productivity levels.
2 substitues who disassemble the boxes.

The initial stock of the *box game* is 75 big and small boxes each. However, the aim is not to produce the entire demand in the shortest time possible with all employees but to respect the customer's demand - one part every 15 seconds - without inventory and with as few employees as possible.

Typically, the simulation is played in four rounds with defined duration, for example 5 or 8 minutes. During the game two types of principles with two batch sizes are played.

Batch size 3 - push principle: The products are passed on in batches of size 3. Each process step works functionally independent from the other and the participants are rewarded based on the number of pieces they work on. Hence, it is the goal at each station to produce as much output as possible.

Batch size 3 - pull principle: Stations produce and pass on products in size 3 batches. Upstream stations have to hold their pieces and stop production until it is demanded internally or externally. Capacity of a station is limited to 3 items and items can only be replaced accordingly.

Batch size 1 - pull principle: The third round is played like the second one, but the batch size is reduced to 1.

Improvement - pull principle: The last round is used to find tweaks autonomously and to implement them as a team.

The advantage of this approach is that the participants gather personal experiences. This can hardly be replaced by a computer simulation. However, augmenting this hands-on experience by such a simulation helps to scale up both complexity and range of the considered process.

IV. RELATED WORK

Since this paper combines conceptual modeling with Petri nets and the simulation of laboratory processes in logistics, for both fields related work is considered.

A. Push, Pull and Kanban

In a push production, every workstation produces when supplied sufficiently regardless of a given demand. The advantages of this are steady production and a high utilization rate. In a pull production, the workstations only produce for a given actual demand, leading to lower stocks and a more flexible production. Which of these paradigms is better than the other depends on the circumstances. Sometimes, mixed solutions are best [7].

Kanban is a method to realize pull principles in logistics and hence to lower stocks by controlling the replenishment of material. If a threshold is recognized, a kanban signal initiates a pull request that includes information on the batch size, leading to stable production sizes. Dependent on the used variant, the replenishment can be controlled by cards, empty containers, via e-kanban or by use of a supermarket system [8] [9].

As the pull requests establish a chain starting at the dispatch warehouse, in kanban systems information flows upstream, while material flows downstream. Different types of kanban may be used to account for the type of material, set up times for production or internal factors [10].

B. Push and Pull Simulation with Petri Nets

Petri nets can be used to study the performance of push and pull approaches. Constraints like manufacturing and setup times, vehicle routing or concurrent processing become operational and, thus, flexible manufacturing systems can be examined [7]. Large and interlocked systems can be modeled by expanding on local components; applying different Petri net specifications suited for respective tasks is beneficiary [11].

Almost all of these approaches lack the handling of real time values and rely on some kind of generic times. We assume that this is because of the absence of proper modeling tools for timed Petri nets [12].

C. Time Concepts in Petri Nets

Since time is an important dimension for the modeling of processes and dynamic systems in general, there exist numerous approaches for handling time aspects in Petri nets. They differ concerning their expressiveness (discrete or continuous time) and which Petri net elements are used to express time constraints (place, transition, arc, or token).

One possible implementation puts one or two time values on transitions, the lower being a delay up to which the transition is not enabled while the higher presents the latest possible moment of firing. This may lead either to a forced firing, the reset of a clock - where the lower value describes a kind of preparation time and the higher one an expiration time from when a new preparation needs to be conducted - or even to a dead net. Variations include time consuming firing [13] as well as firing without time consumption [14].

Another obvious possibility is to assign time values to the places, again representing lower and upper bounds. These bounds represent the availability of tokens, either as delay until a token becomes available [15] or as time windows in which they are available [16] [17].

Yet, another possible implementation is to define the permeability of arcs relative to the moment an adjacent place was marked or an adjacent transition was enabled. The so far cited concepts are equivalent [18]. However, they have the disadvantage that the state of such nets does not only rely on the respective markings, but also on some kind of timer clocks.

D. Higher Petri Nets and Time

Originally, Petri nets have been defined as Place/Transition nets (P/T) with anonymous tokens indicating a system's state [19]. Diverse concepts for representing high level information in Petri nets exist, with the most widely known being Predicate/Transition nets (Pr/T) and Colored Petri Nets (CPN).

Pr/T and CPN omit anonymous tokens for ones carrying information that can be processed and altered by functions encoded on transitions and arcs. They are used to select tokens from the preset and to calculate new values for tokens on the postset of transitions. Places serve as predicates according to which transitions may fire. Thus, it is possible to model data-driven processes or dynamic systems in general. In CPN, elements are additionally provided with a color, hereby inducing a parallelization in representation: when determining the status of a transition, places and tokens are examined by color separately [20]-[22].

Timestamps are means to encode time information in the marking. In Timestamp Nets, tokens designate the moment the corresponding token was placed. Transitions may fire in time windows as given on the transitions' incoming arcs [23] [24]. Extended Timestamp Nets integrate the concepts of Pr/T and Timestamp nets such that tokens carry timestamps and any further information [25].

Some of the approaches may be transformed into each other effortlessly [26] [27]. All of the presented Petri net formalisms use artificial, abstract time units. To model and simulate real-world applications, real time values however should be used instead. To this end, date and time data types seem beneficial to be included as possible information on tokens.

E. Further Modeling Approaches

For reference, there are other modeling methods that were developed to combine time and process structures. One of these methods are Value Stream Diagrams, which are suitable for high-level overviews of processes as they consider whole value streams from customer to supplier [28] [29].

Also, Business Process Model and Notation (BPMN) is extensively used for modeling business processes due to the relative ease of both creating and understanding models. Using BPMN, it is possible to create both high-level models of companies and low-level models of single processes in a graphical approach similar to flowcharts [30].

F. Process-Simulation.Center

If conceptual models are to be developed for process simulation or even execution, tools are needed in addition to a formal mathematical base. The P-S.C is a novel, Web-based modeling and simulation environment supporting the development of P/T and Pr/T nets.

In P-S.C, it is possible to assign data types to places and use these places in analogy to tables in databases. Special types for date and time are important substructures for the simulation of processes in production and logistics and enhance the mentioned approaches to timed Petri nets.

In opposite to relational algebra and, hence, SQL, where operations like select or projection are applied to all affected tuples at once and the result of a relational operation is always a set, in P-S.C the processing of tuples on places is serialized. The reason for this design choice is that, in business and production processes, work items are also treated one after another. The concrete sequence is decided upon locally by the transitions of the net and its marking.

Moreover, the P-S.C can be used to connect the process view on a system with other views. For instance, usage of process maps may combine different processes with each other and express the strategic value of a specific process as a primary process, a support process or a process on the management level. Also, the organizational structure of an institution can be combined with the Petri net view on the processes by assigning its nodes to the swim lanes of the corresponding responsible organizational units. Organizational charts themselves complete the functions of the P-S.C.

It is worth mentioning that the P-S.C draws nodes in such a way that their labels can be presented within, which facilitates reading and understanding. To further strengthen visual clues of their functionality, nodes can be provided with symbols.

Contrary to most other conceptual modeling tools, especially those developed for Petri nets, a specification language rather than a graphical editor has been designed for the P-S.C with which all types of models are scripted. Due to strong algorithms for automatic layout, modelers can concentrate on structural aspects of the domain to be expressed.

The dearth of current Petri net tools, the quaint user experience of most of the still working ones and the unique approach of using textual programming instead of drag-and-drop modeling in combination with the added functionality are the main reasons for the implementation of the P-S.C.

Further information about the tool has already been referred to in Section II.

V. Event Triggered Simulation Models

One target of this paper is to develop an event triggered simulation model to determine the total costs of all involved stocks and to explain the applied modeling technique. Hence, the presented Petri net models mainly consist of places for these stocks, connected by transitions and arcs that represent the functionality. Since an ideal production flow is one with minimal stocks and short throughput times, the individual boxes are passed from one production step directly to the following, establishing a batch size of 1. Other batch sizes are implemented accordingly [9].

Transitions carry selection criteria about which tokens are to be processed next if there are several available (normally the one with the lowest id according to the FIFO principle is chosen). These criteria may be displayed on the transitions by clicking the plus symbol.

Further computation instructions are deposited on the arcs; they are required for the simulation's output as they provide information about how much time is needed for each production step in the complete process. Again, arc inscriptions can be shown or hidden as required.

A. The Push Model

Figure 2 shows the push version of the *box game*. Single boxes get delivered from the main storage *inStore* to the upstream buffers *inBB* and *inSB* of the assembly stations.

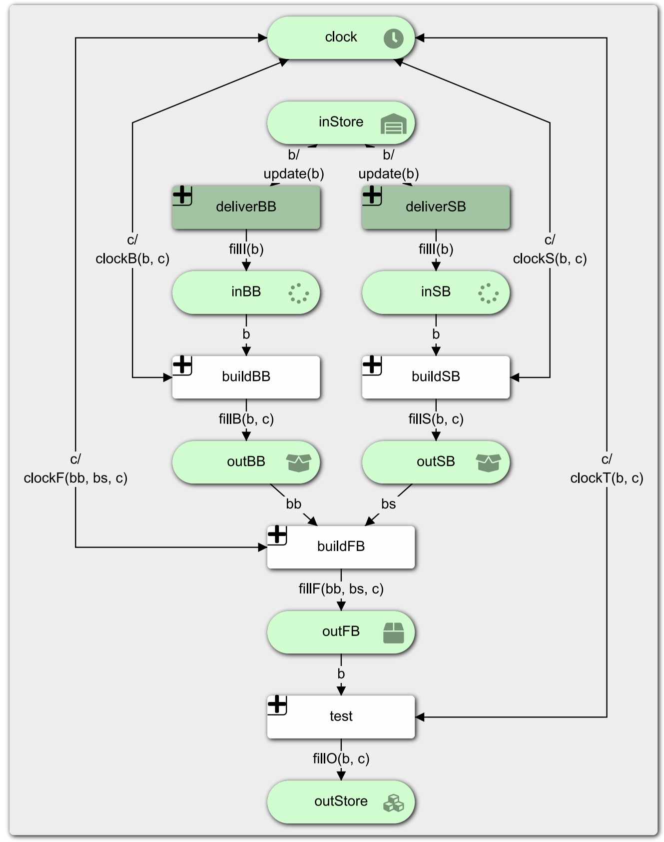

Figure 2. Event Triggered Push Model

Modeling and simulation of the workplaces themselves is abridged by use of the build transitions *buildBB*, *buildSB*, *buildFB* and later on the transition *test*. Successively, the boxes are stored in the outgoing buffers of the workplaces *outBB* and *outSB*, final assembly *outFB*, and the exit storage *outStore*.

The place *clock* establishes a possibility to track elapsed times. As the simulations runs, the time data on when items enter or leave a storage position - a part of the respective marking of the Petri net as provided by the *clock* - is logged and the whole data set is supplied in form of comma-separated values. The data set embraces the entire reachability set of all times and amounts which allows for a deep analysis of the net's behavior, especially of waiting times and storage costs.

B. The Pull Model

The pull version of the *box game* as depicted in Figure 3 is partly more complicated, as supplementary elements are needed to implement pull requests. On the other side, as time logging can be implemented by tracing tokens forward and backward, the *clock* and some arcs may be dropped.

In Section IV-A, it has been mentioned that information flows from the dispatch warehouse upstream. Thus, the kanban chain should start at *outStore* instead of where it is modeled. However, in the presented model theoretically necessary arcs between the storage places establishing single pulls may partly be omitted. Skipping these connections is feasible as all pulls beside the modeled one arise from working steps that require less time than *buildFB*, so their storage places are empty by specification at the relevant times. Modeling the full kanban chain becomes necessary when considering larger batch sizes.

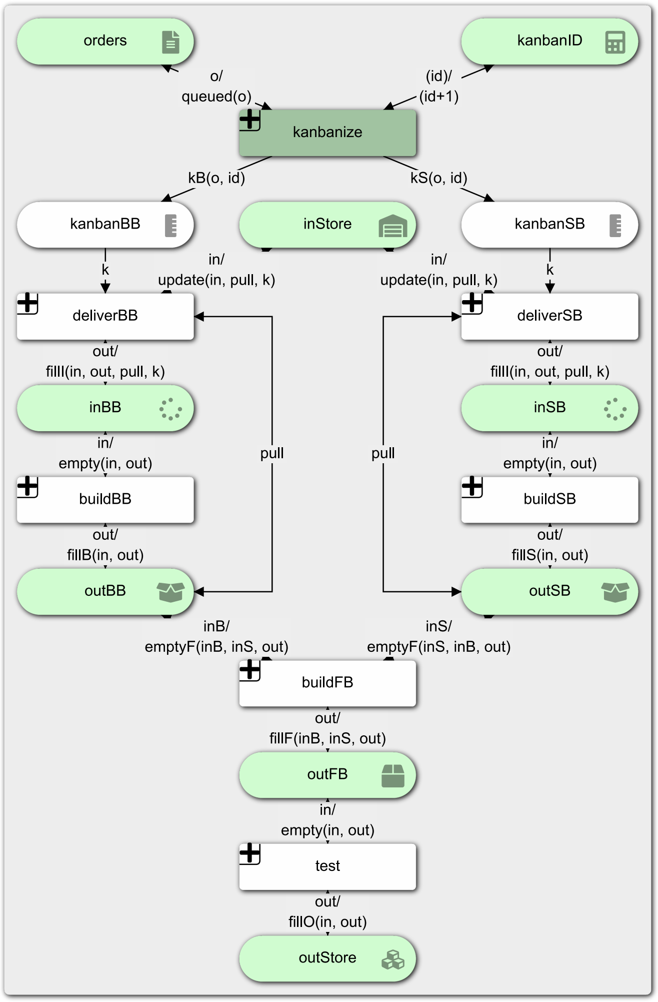

Figure 3. Event Triggered Pull Model

As a first step, tokens on *orders* are transposed into kanban tokens that adhere to the given batch size. These tokens are put on *kanbanBB* and *kanbanSB* in order to control the deliveries from the main storage *inStore*.

In connection with information flowing through additional *pull* arcs, the kanban principle is implemented. When there are no boxes on the preassembly's outgoing storage places *outBB* and *outSB* (which also serve as input buffer for *buildFB*), a pull request for raw material and, thus, production of the specified batch is initiated. The model accounts for both necessary delivery times from main storage to buffers and the longer processing time of the big boxes, leading to later dispatch times of the small boxes. This is done to minimize the waiting time of intermediates in the buffers.

The remainder of the model works like the push version.

VI. Conclusion and Future Work

As expected, the simulation of the two models show no difference in total processing or idle times for single workplaces. Differences on storage places become evident, though.

As the distribution of waiting times behind *buildFB* is the same for both models, only the first six storage places need to be analyzed. The upper part of Figure 4 shows the allocation of the boxes' waiting times to single stocks in the push model while the lower part shows the same for the pull model.

Stocks on *inStore* are split by type. The boxes themselves are aggregated into Five-Box-Clusters for clarity. The trend to successively longer waiting times is clearly visible.

The push model's *inStore* gets cleared as fast as possible. This, however, leads to large buffers just before the first concurrent production steps. As the preassembly requires less time than the final assembly, the two incoming buffers for *buildFB* are also highly occupied.

Although the accumulated waiting times are equivalent, the pull model unites them on the main storage *inStore* only.

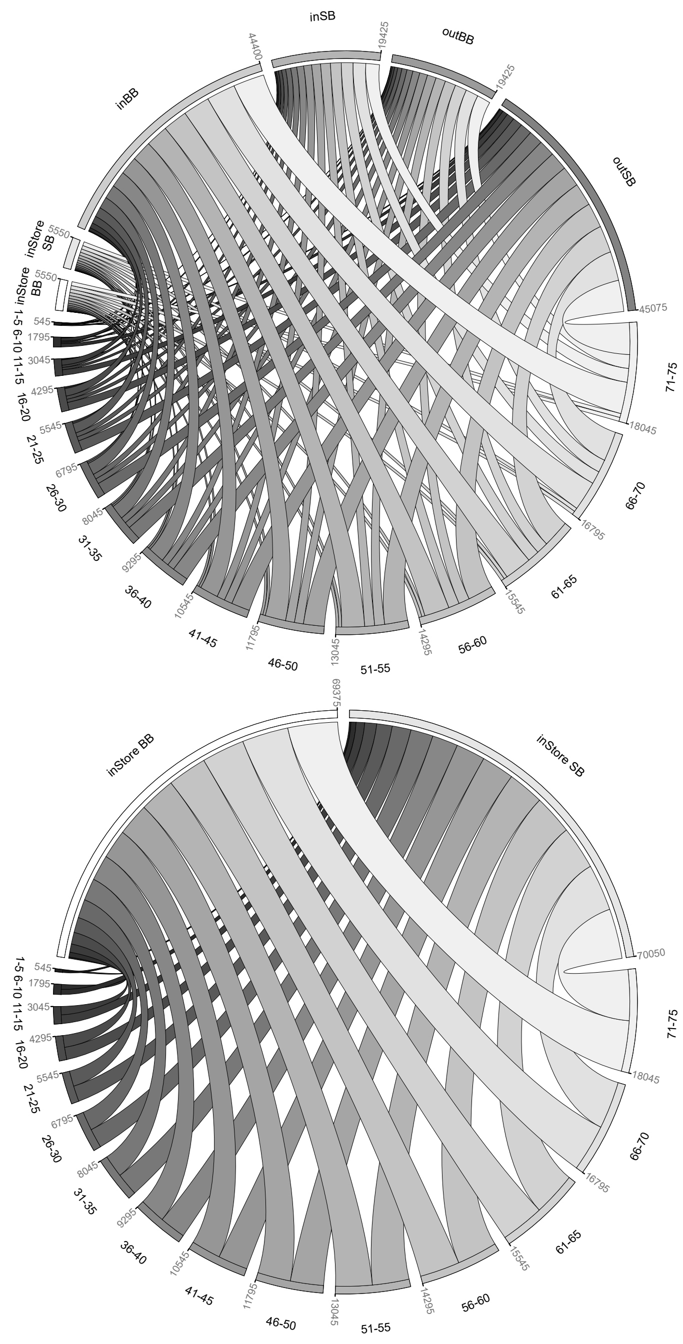

Figure 4. Accumulated Waiting Times [in Seconds]:
(Upper Circle) Push Model · (Lower Circle) Pull Model:
(Upper Half Circles) Named Storage Places · (Lower Half Circles) Clusters of Five Successive Boxes · (Circular Area) Distribution of Accumulated Waiting Times of Five-Box-Clusters to Individual Storage Places

 ISBN: 978-1-61208-831-0

As anticipated, the pull principle leads to zero interim buffer inventory throughout the stocks under consideration. For manufacturing companies, this opens up the possibility to externalize costs. One such alternative is to use consignment stores: suppliers maintain warehouses inside the customer's facility and, thus, must keep the material in their balance sheet. Another common possibility is just-in-time delivery. With high process stability and fast response times, even the use of a kanban cycle with the supplier is conceivable [10]. These options may lead to partial or total eradication of stocks.

An example for what is not accounted for in these models is the cost of transportation, as smaller batch sizes usually correspondent to higher transportation costs, leading to familiar knowledge: decreasing one muda typically increases others. Hence, batch size 1, which is optimal for interim storage costs, may not necessarily be the globally optimal solution.

Event triggered simulations show performance advantages over clock pulse simulations. In this example, the push model needs 79 calculation steps while the pull model requires 228, due to the additional kanban flows. However, both models outperform their respective clock pulse counterpart that both run for 1902 computation steps each - one step for every second elapsed. Hence, the presented approach scales better, which will be beneficial for an industrial application.

During development of the shown models, the following modeling techniques turned out to be helpful:

1. Define data types for the different stocks and other data objects, and initialize the corresponding places in accordance with the starting condition.
2. Augment the model by transitions for beginning and ending specific tasks like delivering raw materials, building or testing a box.
3. Identify the next item to be taken and the moment this will occur. This also allows for implementation of different prioritization strategies.
4. Start with modeling the simpler push principle and augment this model by pull principles.
5. Look for a proper visualization of the simulation results.

Event triggered and clock pulse simulation both react to discrete, clearly distinguishable events and calculate predictable new states. The world, however, is continuous and decisions are often fuzzy. Future work will, therefore, focus on further, different techniques for developing conceptual models for dynamic systems, taking this aspect into account. This work will also rely on the existence of a modeling and simulation environment and hence work on the P-S.C will be continued.

References

[1] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” MIS Quarterly, vol. 28, no. 1, 2004, pp. 75–106.

[2] C. Simon and S. Haag, “Simulatable Reference Models To Transform Enterprises For The Digital Age – A Case Study,” in ECMS 2020: 34rd International Conference on Modelling and Simulation, M. Steglich, C. Müller, G. Neumann, and M. Walther, Eds., 2020, pp. 294–300.

[3] ——, “Digitale Zwillinge modellieren und verstehen: Eine Fallstudie zum problembasierten und forschenden Lernen,” in MoHoL 2020 (Workshop): Modellierung in der Hochschullehre, J.-R. Rehse, M. Striewe, and M. Ullrich, Eds., Wien, Austria, 2020, pp. 101–112.

[4] S. Haag and C. Simon, “Simulation of Horizontal and Vertical Integration in Digital Twins,” in ECMS 2019: 33rd International Conference on Modelling and Simulation, M. Iacono, F. Palmieri, M. Gribaudo, and M. Ficco, Eds., 2019, pp. 284–289.

[5] C. Simon and S. Haag, “Simulation vertikaler Integration: Vom Top-Floor zum Shop-Floor und zurück,” in Tagungsband AKWI, T. Barton, F. Herrmann, V. G. Meister, C. Müller, C. Seel, and U. Steffens, Eds., 2018, pp. 104–113.

[6] C. Simon, “Eine Petri-Netz-Programmiersprache und Anwendungen in der Produktion,” in Tagungsband AKWI, T. Barton, F. Herrmann, V. G. Meister, C. Müller, and C. Seel, Eds., 2017, pp. 61–70.

[7] M. Zhou and K. Venkatesh, Modeling, Simulation, and Control of Flexible Manufacturing Systems - A Petri net Approach, ser. Intelligent Control and Intelligent Automation. Singapore, New Jersey: World Scientific, 1999, vol. 6.

[8] U. Dombrowski and T. Mielke, Ganzheitliche Produktionssysteme: Aktueller Stand und zukünftige Entwicklungen. München: Springer Vieweg, 2014.

[9] J. Gottmann, Produktionscontrolling: Werströme und Kosten optimieren, 2nd ed. Wiesbaden: Springer Gabler, 2019.

[10] H. Wildemann, Kanban-Produktionssteuerung, 28th ed. München: TCW, 2020.

[11] L. Recalde, M. Silva, J. Ezpeleta, and E. Teruel, Lectures on Concurrency and Petri Nets: Advances in Petri Nets. Berlin: Springer, 2004, ch. Petri nets and manufacturing systems: An examples-driven tour, pp. 742–788.

[12] “Petri Nets Tools Database Quick Overview,” https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html (last accessed 2020.09.01).

[13] C. Ramchandani, “Analysis of Asynchronous Concurrent Systems by Timed Petri Nets,” MIT, Project MAC, Technical Report 120, 1974.

[14] P. Merlin, “The Time-Petri-Net and the Recoverability of Processes,” University California, Irvine, Tech. Rep., 1974.

[15] J. Sifakis, “Use of petri nets for performance evalutation,” in Measuring, modelling and evalutating computer systems, ser. IFIP, H. Beilner and E. Gelenbe, Eds., North Holland Publ. Co., 1977, pp. 75–93.

[16] H.-M. Hanisch, Petri-Netze in der Verfahrenstechnik. München: Oldenbourg, 1992.

[17] ——, “Dynamik von Koordinierungssteuerungen in diskontinuierlichen verfahrenstechnischen Systemen,” in Petrinetze in der Automatisierungstechnik, E. Schnieder, Ed. München, Wien: Oldenbourg Verlag, 1992.

[18] R. König and L. Quäck, Petri-Netze in der Steuerungs- und Digitaltechnik. München, Wien: Oldenbourg Verlag, 1988.

[19] C. A. Petri, “Kommunikation mit Automaten,” Schriften des Institutes für instrumentelle Mathematik, Bonn, Tech. Rep., 1962.

[20] H. J. Genrich and K. Lautenbach, “System Modelling with High-Level Petri Nets,” Theoretical Computer Science, vol. 13, 1981.

[21] H. J. Genrich, “Predicate/Transition Nets,” in High-level Petri Nets: Theory and Applications, K. Jensen and G. Rosenberg, Eds. Springer, 1991, pp. 3–43.

[22] K. Jensen, Coloured Petri-Nets, 1st ed. Berlin: Springer, 1992.

[23] C. Ghezzi, D. Mandrioli, S. Morasca, and M. Pezzè, “A unified high-level petri net formalism for time-critical systems,” IEEE Transactions On Software Engineering, vol. 17, no. 2, 1991, pp. 160–172.

[24] H.-M. Hanisch, K. Lautenbach, C. Simon, and J. Thieme, “Timestamp Nets in Technical Applications,” in IEEE International Workshop on Discrete Event Systems, San Diego, CA, 1998.

[25] K. Lautenbach and C. Simon, “Erweiterte Zeitstempelnetze,” Universität Koblenz-Landau, Universität Koblenz-Landau, Fachberichte Informatik 03–99, 1999.

[26] K. Jensen, “High-Level Petri Nets,” Informatik-Fachberichte, vol. 66, 1983, pp. 166–180.

[27] L. Popova-Zeugmann, Time and Petri Nets. Berlin: Springer, 2013.

[28] C. E. Knoeppel, Installing Efficiency Methods. The Engineering Magazine, 1915.

[29] T. Ohno, Toyota Production System. Milton Park, UK: Taylor & Francis, 1988.

[30] OMG, “BPMN 2.0 - Business Process Model and Notation,” http://www.bpmn.org/ (last accessed 2020.09.01), 2011.