Event Triggered Simulation of Push and Pull Processes

Stefan Haag, Lara Zakfeld, Carlo Simon, Christian Reuter

{haag, lara.zakfeld, simon, reuter}@hs-worms.de

Hochschule Worms
University of Applied Sciences

October 2020

Anticipation of Change

How can manufacturers reduce costs and produce more flexible?

Obviously, they could invest in ”better” machines

Rethinking production strategies and processes may potentially be cheaper

But what are the consequences of change?

Favourability of Push or Pull Principles

Using push principles, information flows parallel to workpieces
Thus, production is supply-driven

Using pull principles, information flows anti-parallel to workpieces
Thus, production is demand-driven

Both approaches have their benefits and drawbacks

But how to determine which one is favorable?

What to Model & How to Simulate

- Simulatable Models could help in objectifying decisions on reorganizations or investments
- Complex real-world processes necessitate sophisticated models
- Event triggered models are introduced, in which calculations are only performed when the system state actually changes
- Utilizing the Process-Simulation.Center (P-S.C) - a novel, web-based Petri net modeling and simulation environment - an approach to examine costs and consequences of change is presented
- To this end, a simulation laboratory for teaching processes in logistics is introduced as example to evaluate the change from push to pull principles in a manufacturing setting
- In parallel, a guideline is established as to how to develop simulatable models following an event triggered approach

Push, Pull & Kanban

Zhou and Venkatesh (1999): Modeling, Simulation, and Control of Flexible Manufacturing Systems
Dombrowski and Mielke (2014): Ganzheitliche Produktionssysteme
Gottmann (2019): Produktionscontrolling
Wildemann (2020): Kanban-Produktionssteuerung

Push & Pull with Petri nets

Zhou and Venkatesh (1999): Modeling, Simulation, and Control of Flexible Manufacturing Systems
Recalde et al. (2004): Lectures on Concurrency and Petri Nets

Time & Petri nets

Ramchandani (1974): Analysis of Asynchronous Concurrent Systems by Timed Petri Nets
Merlin (1974): The Time-Petri-Net and the Recoverability of Processes
Sifakis (1977): Use of petri nets for performance evalutation
Hanisch (1992b): Petri-Netze in der Verfahrenstechnik
Hanisch (1992a): Dynamik von Koordinierungssteuerungen
König and Quäck (1988): Petri-Netze in der Steuerungs- und Digitaltechnik
Genrich and Lautenbach (1981): System Modelling with High-Level Petri Nets
Genrich (1991): Predicate/Transition Nets
Jensen (1992): Coloured Petri-Nets
Ghezzi et al. (1991): A unified high level petri net formalism for time-critical systems
Hanisch et al. (1998): Timestamp Nets in Technical Applications
Lautenbach and Simon (1999): Erweiterte Zeitstempelnetze

Alternative Modeling Approaches

Knoeppel (1915): Installing Efficiency Methods
Ohno (1988): Toyota Production System
BPMI (2004): BPMN 1.0 - Business Process Model and Notation
OMG (2011): BPMN 2.0 - Business Process Model and Notation

The *Box Game*

The *Box Game* has been developed at the University of Applied Sciences Worms

Implemented as a teaching laboratory, it aims at imparting knowledge in logistics

Focus lies on the strategic level, not on scheduling or problems concerning mechanical production

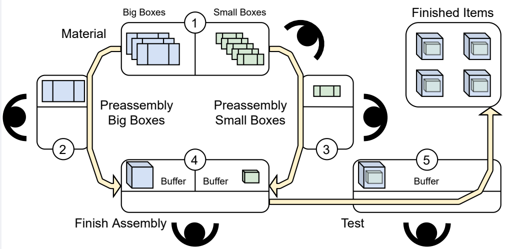
The *Box Game* has almost non-existing requirements:

- Five tables are arranged as working and storage places
- Standard positions like buffers are marked with adhesive tape
- Cardboard boxes are used as workpieces

The game is easily transferable to assembly workstations

The Rules of the Game

- Deliver unfolded boxes from the warehouse (1) to the preassemblies (2) & (3)
- Fold and close big (2) and small (2) boxes
- Pass the boxes to the final assembly (4)
- Open the big box and insert the small one Then, close and seal the big box
- Pass the box to the quality assurance (5) where a shake (as acoustic check) is performed
- Put the finished box on the outgoing storage

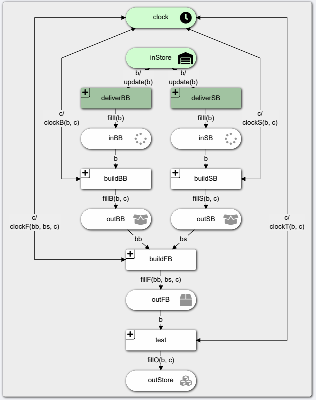


Spatial organization of the *box game*

The *Event Triggers*

- Event triggered models rely on the knowledge of state changes in a system
- Only when such a system state change occurs, a new model state is processed
- This possibly provides execution speed advantages over clock pulse models which compute a new model state each discrete time step
- The presented models simulate the *box game* and implement two different logistics strategies:
  - Push processes guide workpieces through production as fast as possible
  - Pull processes initiate production only when there actually is a demand
- Both models have the same setup - initial stocks and a batch size of one - and also look quite similar

Push Model



- Places (ovals) serve as storage,
- transitions (rectangles) as activities,
- the *clock* "stores" time information,
- and arcs transport objects and, thus, update data records

A system change occurs when a workpiece alters its position, i.e. it leaves or enters a storage

Items flow from the *inStore* through the production to the *outStore*

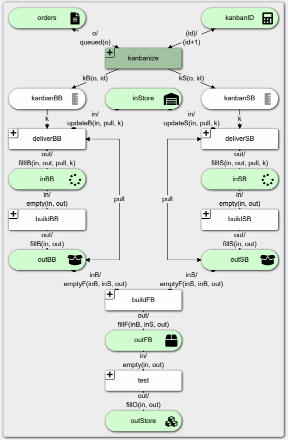
Petri net model as depicted in the P-S.C

Additional elements and conditions implement pull principles

Orders get kanbanized in preparation for pulls

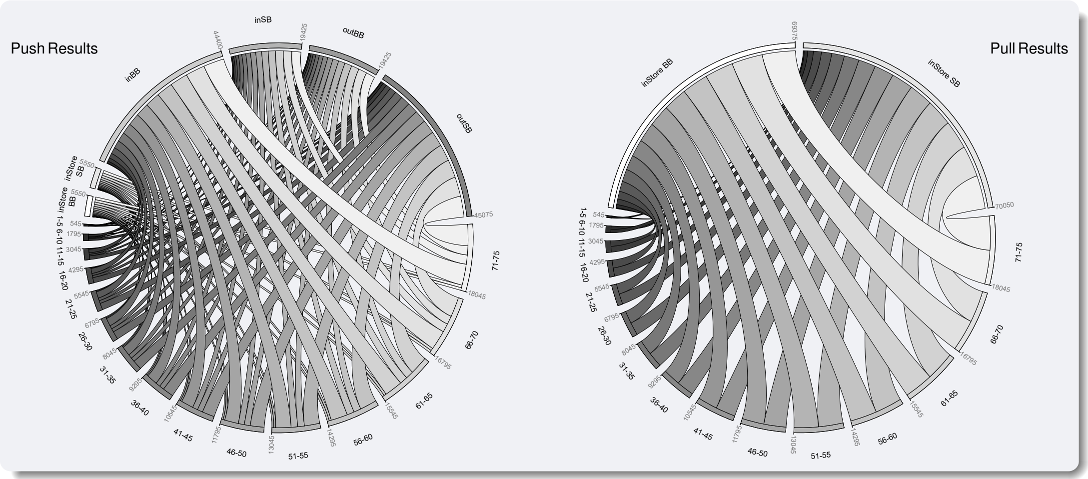
Tokens representing stocks may save the time of the place's last update

type	count	batchSize	updated
FB	1	1	0:03:15



Pull Model

(left) Data record on *outFB* after 22 computation steps
(right) Petri net model as depicted in the P-S.C

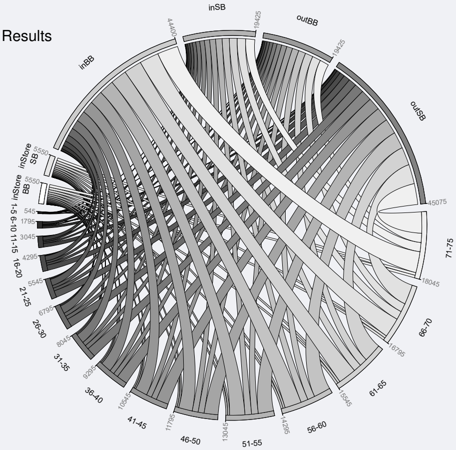


Accumulated waiting times [in seconds] as costs

(Upper half circles) Named storage places · (Lower half circles) clusters of five successive boxes

(Circular areas) Distribution of accumulated waiting times of five-box-clusters to individual storage places

Push Results



Costs on the main storage:

- 5 550 for big boxes
- 5 550 for small boxes

Costs on the *in* buffer storage:

- 44 400 for big boxes
- 19 425 for small boxes

Costs on the *out* buffer storage:

- 19 425 for big boxes
- 45 075 for small boxes

139 425 total

11 100 potentially externalizable

Accumulated waiting times [in seconds] as costs
(Upper half circle) Named storage places · (Lower half circle) clusters of five successive boxes
(Circular area) Distribution of accumulated waiting times of five-box-clusters to individual storage places

Costs on the main storage:

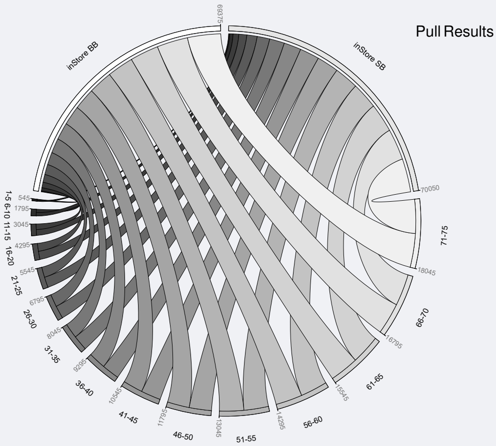
- 69 375 for big boxes
- 70 050 for small boxes

No costs on the *in* buffer storage

No costs on the *out* buffer storage

139 425 total

139 425 potentially externalizable



Accumulated waiting times [in seconds] as costs
(Upper half circle) Named storage places · (Lower half circle) clusters of five successive boxes
(Circular area) Distribution of accumulated waiting times of five-box-clusters to individual storage places

Visualization

- Tokens carry all information about the current state of the system
- Thus, all information about the simulation can be extracted from a reachability set
- The actually reached set of system states is exported (e.g. as .csv-File) and storage costs are extracted and processed from the complete set
- The result is a table with dimensions storage place and costs per item lot
- As visualization a circular ideogram layout was chosen using the tool set Circos (http://www.circos.ca):
  - The upper halves of the circles represent the tables' columns, the lower halves represent the tables' rows and the connections relate to the value of single cells
- Obviously, as data sets become larger, good visualization becomes even more important

Guidelines

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

Takeaways

- Consequences of change can be modeled upfront given the right tools
- To a large extent, these tools are still missing or lacking, thus, their capabilities need to be enhanced, which includes the P-S.C
- Modeling pull principles is hard, even if following the presented guidelines, hence, they should be refined to account for this
- Such a refinement should include handling of mixed push/pull systems

What's to be done next?

- The presented models react on discrete, clearly distinguishable events which may not be present in a continuous and fuzzy reality - thus, different techniques for the conceptual modeling of dynamic systems focusing on this aspect need to be developed
- For large systems, scaling needs to be examined - both in performance and in modelers' usage
- Regarding the usage in the *box game*, modeling concepts like in-house transportation or scheduling may further improve the students' learning experience and their knowledge of logistic connections

BPMI (2004): BPMN 1.0 - Business Process Model and Notation. https://www.omg.org/spec/BPMN/ (last accessed 2020.09.20).
Dombrowski, U. and T. Mielke (2014): *Ganzheitliche Produktionssysteme: Aktueller Stand und zukünftige Entwicklungen*. Springer Vieweg, München.
Genrich, H. J. (1991): Predicate/Transition Nets. In: *High-level Petri Nets: Theory and Applications*, eds. K. Jensen and G. Rosenberg. Springer, pp. 3–43.
Genrich, H. J. and K. Lautenbach (1981): System Modelling with High-Level Petri Nets. *Theoretical Computer Science*, 13.
Ghezzi, C.; D. Mandrioli; S. Morasca; and M. Pezzè (1991): A Unified High-Level Petri Net Formalism for Time-Critical Systems. *IEEE Transactions On Software Engineering*, 17(2):160–172.
Gottmann, J. (2019): *Produktionscontrolling: Werströme und Kosten optimieren*. Springer Gabler, Wiesbaden, 2nd edn.
Hanisch, H.-M. (1992a): Dynamik von Koordinierungssteuerungen in diskontinuierlichen verfahrenstechnischen Systemen. In: *Petrinetze in der Automatisierungstechnik*, ed. E. Schnieder, Oldenbourg Verlag, München, Wien.
Hanisch, H.-M. (1992b): *Petri-Netze in der Verfahrenstechnik*. Oldenbourg, München.
Hanisch, H.-M.; K. Lautenbach; C. Simon; and J. Thieme (1998): Timestamp Nets in Technical Applications. In: *IEEE International Workshop on Discrete Event Systems*. San Diego, CA.
Jensen, K. (1992): *Coloured Petri-Nets*. Springer, Berlin, 1st edn.
Knoeppel, C. E. (1915): *Installing Efficiency Methods*. The Engineering Magazine.
König, R. and L. Quäck (1988): *Petri-Netze in der Steuerungs- und Digitaltechnik*. Oldenbourg Verlag, München, Wien.
Lautenbach, K. and C. Simon (1999): *Erweiterte Zeitstempelnetze*. Fachberichte Informatik 03–99, Universität Koblenz-Landau, Universität Koblenz-Landau.
Merlin, P. (1974): *The Time-Petri-Net and the Recoverability of Processes*. Tech. rep., University California, Irvine.
Ohno, Taiichi (1988): *Toyota Production System*. Taylor & Francis, Milton Park, UK.
OMG (2011): BPMN 2.0 - Business Process Model and Notation. http://www.bpmn.org/ (last accessed 2020.09.20).
Ramchandani, C. (1974): *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*. Technical Report 120, MIT, Project MAC.
Recalde, L.; M. Silva; J. Ezpeleta; and E. Teruel (2004): *Lectures on Concurrency and Petri Nets: Advances in Petri Nets*, Springer, Berlin, chap. Petri nets and manufacturing systems: An examples-driven tour, pp. 742–788.
Sifakis, J. (1977): Use of Petri nets for performance evalutation. In: *Measuring, modelling and evalutating computer systems*, eds. H. Beilner and E. Gelenbe. North Holland Publ. Co., IFIP, pp. 75–93.
Wildemann, H. (2020): *Kanban-Produktionssteuerung*. TCW, München, 28th edn.
Zhou, M. and K. Venkatesh (1999): *Modeling, Simulation, and Control of Flexible Manufacturing Systems - A Petri net Approach*, vol. 6 of *Intelligent Control and Intelligent Automation*. World Scientific, Singapore, New Jersey.