

Simulation taktgesteuerter Modelle von Push- und Pull-Prozessen in der Logistik

Carlo Simon und Stefan Haag und Lara Zakfeld

Fachbereich Informatik

Hochschule Worms

Erenburgerstr. 19, 67549 Worms

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

Zusammenfassung—Der Wechsel von einer Push- zu einer Pull-Strategie stellt einen erheblichen Eingriff in die betriebliche Logistik dar und hat Auswirkungen auf Beschaffungsstrategien, die Ausgestaltung von Abläufen sowie die bewerteten, internen Lagerbestände. Die Entwicklung von Simulationsmodellen, die die Konsequenzen einer solchen Umstellung vorwegnehmen, ist Gegenstand dieses Beitrags. Ein solches Simulationsmodell muss das Systemverhalten über den Zeitlauf aber in Zeiträufeln abbilden - die Systemzustände sind also nach Zeittakten zu bestimmen. Gleichzeitig soll es tatsächliche Produktionsdaten berücksichtigen und muss somit Entscheidungsregeln abbilden können.

Mit Petri-Netzen kann man diese Aspekte realisieren, doch benötigt man dann ein Modellierungs- und Simulationstool wie das Process-Simulation.Center (P-S.C), welches höhere Petri-Netze mit individuellen Marken verarbeiten kann. Da vergleichbare Software bislang nicht verfügbar war, fehlte das „Werkzeug“, um Vorgehensweisen und Techniken der Modellerstellung zu trainieren und zu erproben. Die Entwicklung eines Simulationsmodells für einen Wechsel von einer Push- zu einer Pull-Strategie ist also mit der Verwendung eines Tools verwoben.

Um die Aufgabe konkret zu machen, wurde ein Logistik-Lernlabor abgebildet. Das entwickelte Modell kann nun genutzt werden, um die persönlichen Beobachtungen von Studierenden im Labor zeitlich, mengenmäßig und wertmäßig zu skalieren.

Im Folgenden wird zunächst der Laboraufbau als Beispiel-szenario für simulationsgestützte Managemententscheidungen beschrieben. Nachdem dann die neuartigen Funktionen des P-S.C vorgestellt werden, wird hierauf aufbauend das Vorgehen beschrieben, um Modelle für Push- und Pull-Strategien inkl. in den Petri-Netzen selbst integrierter Dashboards zu erklären. Abschließend wird die taktgesteuerte Simulation einer weiteren Modellierungstechnik, der ereignisgesteuerten Simulation, gegenübergestellt.

Keywords—Konzeptuelle Modelle zeitdynamischer Systeme, Simulation, Petri-Netze, Logistik, Lehre

I. SIMULATIONSGESTÜTZTE MANAGEMENTENTSCHEIDUNGEN

Enterprise-Resource-Planning-Systeme (ERP) sammeln und integrieren betriebliche Daten in einer Datenbank [1]. Diese Informationen können in allen Bereichen eines Unternehmens wie Vertrieb, Einkauf und Finanzen, aber auch in der Produktion genutzt werden. Somit werden sie in vielen Unternehmen zur Planung und Steuerung von Fertigungsprozessen eingesetzt. Sie bewerten und terminieren aufkommende Aufträge, legen Produktionspläne fest und erfassen den Arbeitsfortschritt durch Rückmeldungen. Damit unterstützen bzw. automatisieren ERP-Systeme operative Entscheidungen.

Ihre Grenzen haben ERP-Systeme jedoch bei der Bewertung möglicher Veränderungen von Produktionsstrategien. So geht ein Wechsel von Push- zu Pull-Prozessen in Logistik und Produktion zumeist einher mit einer deutlichen Senkung der Lagerkosten bei gleichzeitiger Flexibilitätssteigerung. Da ERP-Systeme aber üblicherweise nur die aktuell eingesetzte Strategie implementieren, sind die Effekte eines möglichen Strategiewechsels mit diesen Systemen nicht abbildbar. So lässt sich beobachten, dass noch immer viele Unternehmen nach dem Push-Prinzip produzieren, obwohl bei ihrem Produktionssortiment ein Wechsel opportun wäre. Ein Grund mag sein, dass alternative, IT-gestützte Methoden der Produktionsplanung und -steuerung nur unzureichend bekannt sind.

Um letzteres zu vermeiden, sollte die Vielfalt an Strategien zur Produktionssteuerung fest in Vorlesungen der Logistik verankert sein und aktiv mit IT-Unterstützung vermittelt werden. Auch Lernlabore wie das sogenannte *Box Game*, das folgend vorgestellt wird, machen die Auswirkungen strategischer Veränderungen für die Studierenden erlebbar. Fehlende Ressourcen in der Lehre münden leider oft in einer verkürzten, eher theoretischen Darstellung. Dieser Trend wurde im letzten Jahr durch eine präsenzlose Lehre während der Corona-Krise noch weiter verschärft. Simulationen könnten genutzt werden, um den Wegfall dieser realen Erfahrungen zumindest teilweise zu kompensieren, oder auch, um Realerfahrungen hinsichtlich ihrer Komplexität zu erweitern.

Das *Box Game* wurde als Anwendungsbeispiel im Studiengang International Logistics Management der Hochschule Worms entwickelt. Trotz des simplen Aufbaus können sehr unterschiedliche Arten von Prozessen beobachtet werden, die auch in der Praxis eine hohe Relevanz aufweisen. Es eignet sich daher hervorragend, um verschiedene Möglichkeiten der konzeptionellen Modellierung und Simulation zu erproben.

Das konkrete Beispiel besteht aus einem einfachen Konstruktionsprozess, bei dem die Studierenden kleine und große Schachteln falten, die kleineren in die größeren Schachteln stecken und am Ende die Qualität überprüfen. Mit diesem Prozess werden, wie noch detaillierter erklärt werden wird, die Eigenschaften von Push- und Pull-Systemen veranschaulicht.

Idealerweise spielen die Teilnehmenden des Kurses das Spiel selbst, erleben so den Arbeitsfluss aus erster Hand mit und entwickeln im Laufe des Spiels ein Gefühl für die Verbesserungsmöglichkeiten.

Schnell erkennen sie dann verschiedene Arten von Verschwendung (sogenannte *Muda* gemäß [2]) wie Überproduktion, Warten und Bewegung. Auch ist der Austausch der erlebten Erfahrungen mit anderen Studierenden ein wesentlicher Teil des Lernerfolgs. Allerdings dauert ein kompletter Durchlauf des *Box Game* zwei bis drei Stunden.

Trotz seiner vermeintlichen Schlichtheit ist das *Box Game* leicht auf industrielle Montagearbeitsplätze übertragbar und hat einen hohen Praxisbezug. Eine maschinelle Fertigung, bei der Zeitpläne, Schichtmuster, Rüstzeiten oder Mehrfachaufstellungen von Maschinen von besonderer Bedeutung sind, würde zwar die Zahl der Optimierungsparameter erhöhen, die Organisationsprinzipien aber nicht substantiell verändern.

Abbildung 1 zeigt die räumliche Organisation des *Box Game* im Lernlabor: Fünf Tische sind als Arbeitsstationen angeordnet. Weitere wichtige Positionen wie Zwischenlager werden mit Klebeband markiert. Aufgrund seiner Einfachheit lässt sich das Lernlabor daher auch auf Schulungs- und Konferenzräume übertragen.

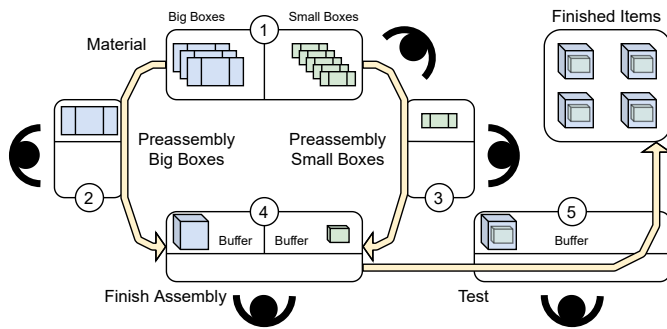


Abbildung 1. Räumliches Layout des *Box Game* [3]

An und zwischen den Arbeitsstationen werden die folgenden Aktivitäten verrichtet:

1. **Lager:** Boxen bereitstellen.
2. **Vormontage große Boxen:** Box auffalten, Deckel schließen und Box weiterleiten.
3. **Vormontage kleine Boxen:** Box auffalten, Deckel schließen und Box weiterleiten.
4. **Endmontage:** Große Box öffnen, kleine Box einlegen und mit einem Post-it als „Packzettel“ labeln, große Box schließen, mit Klebeband versiegeln und weiterleiten.
5. **Qualitätssicherung:** Box als akustischen Qualitätscheck schütteln, roten Klebepunkt an der oberen linken Ecke des Deckels anbringen, Box im Ausgangslager ablegen.

Die folgenden Personen können am Spiel teilnehmen:

- 5 Personen besetzen die Arbeitsstationen,
- 3 Personen erfassen die Bearbeitungszeiten,
- 1 Person erfasst die Bestände,
- 1 Person nimmt die Produktivitätsdaten auf und
- 2 mögliche weitere Personen demontieren die Boxen.

Der anfängliche Lagerbestand beträgt 75 große und 75 kleine Schachteln. Es ist jedoch ausdrücklich nicht das Ziel, möglichst viel in einer möglichst kurzen Zeit zu produzieren.

Stattdessen sollen die Boxen in einem vorgegebenen Kundentank von 15 Sekunden bei geringem Lagerbestand und mit möglichst wenigen Mitarbeitenden hergestellt werden.

Abbildung 2 zeigt das *Box Game* als Wertstromdiagramm inkl. der Bearbeitungszeiten je Prozessschritt. Eine zeitliche und mengenmäßige Simulation dieses Szenarios ist innerhalb des Wertstromdiagramms aufgrund einer fehlenden mathematischen Semantik leider nicht möglich.

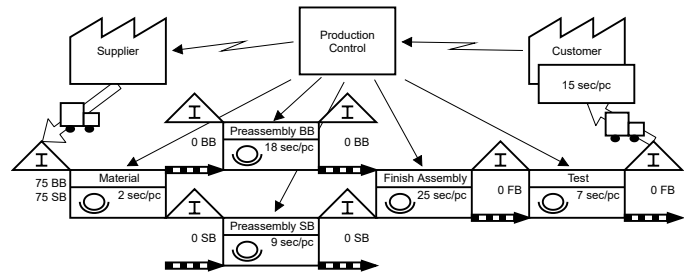


Abbildung 2. Wertstromdiagramm des *Box Game* [4]

Um die Herausforderungen erlebbar zu machen, vor denen ein Management bzgl. einer möglichen strategischen Neuausrichtung steht, sollte das *Box Game* in vier Runden von jeweils 5 oder 8 Minuten Dauer gespielt werden:

Losgröße 3 - Push-Prinzip: Die Produkte werden in der Losgröße 3 weitergereicht. Jeder Prozessschritt arbeitet funktional unabhängig von den anderen und die Teilnehmer werden nach der Stückzahl bezahlt, die sie bearbeiten. Daher ist es das Ziel, an jeder Station so viel Output wie möglich zu produzieren.

Losgröße 3 - Pull-Prinzip: Stationen produzieren und reichen Produkte in der Losgröße 3 weiter. Vorgelagerte Stationen müssen ihre Produkte aber zurückhalten und die Produktion stoppen, bis Produkte von einem nachgelagerten, internen oder externen Kunden angefordert werden. Die Kapazität einer Station und ihres Puffers ist auf drei Stück begrenzt und Teile können nur bis zu dieser Grenze ersetzt werden.

Losgröße 1 - Pull-Prinzip: Die dritte Runde wird analog der zweiten gespielt, allerdings werden Losgröße und Kapazitäten auf eins reduziert.

Verbesserung - Pull-Prinzip: In der letzten Runde geht es darum, selbstständig Verbesserungen zu finden und diese im Team umzusetzen.

Den Vorteil der persönlichen Erfahrung, etwa den Stress, den ein überlaufendes Lager verursacht bis hin zum Gefühl des Kontrollverlustes, kann eine Computersimulation nicht ersetzen. In Zeiten von Corona ermöglicht sie aber zumindest eine teilweise Kompensation. Darüber hinaus erlaubt sie, Komplexität und Reichweite des betrachteten Prozesses zu vergrößern. In den folgenden Abschnitten soll daher untersucht werden, wie statt des gespielten Prozesses eine IT-gestützte virtuelle Simulation gefunden werden kann.

II. GRUNDLEGENDE ÜBERLEGUNGEN ZU EINER IT-GESTÜTZTEN SIMULATION

Aus der jahrzehntelangen Erfahrung der Autoren mit Petri-Netzen sowie dem Umstand, dass sie über ein für akademische Zwecke derzeit frei nutzbares Werkzeug zur Modellierung und Simulation auch höherer Petri-Netze verfügen, resultiert die Entscheidung, das beschriebene Szenario als Petri-Netz umzusetzen. Diese Aufgabe erwies sich jedoch als anspruchsvoller als zu Beginn vermutet:

1. Modelle und Simulationen von Pull-Prozessen müssen zwischen verschiedenen Kundenaufträgen unterscheiden. Dies lässt sich in höheren Petri-Netzen mit individuellen Marken ausdrücken. Aber obwohl solche Petri-Netz-Klassen seit vielen Jahren bekannt sind (z. B. [5], [6]), gibt es keine Modellierungsmuster, die man zur Modellerstellung heranziehen kann.

Umso wichtiger ist es, Modelle bei ihrer Erstellung schrittweise testen zu können und somit en passant neue Modellierungstechniken zu entwickeln. Die gesammelten Erfahrungen sind daher ebenso ein Artefakt der Arbeit wie die Modelle selbst.

2. Ohne ein geeignetes Werkzeug zur Modellierung und Simulation höherer Netze können diese Erfahrungen aber nicht gesammelt werden, was eine Hürde für viele Modelliererinnen und Modellierer darstellt. Fast alle Petri-Netz-Werkzeuge, die in [7] aufgelistet sind, sind entweder veraltet, unterstützen keine Zeitaspekte oder Petri-Netze mit individuellen Token, und keines von ihnen verfügt über eine moderne Benutzeroberfläche. Daher sind sie unbrauchbar für die beschriebene Aufgabenstellung.

Das Process-Simulation.Center (P-S.C) wurde entwickelt, um diese Restriktionen zu überwinden. Dabei handelt es sich um eine neuartige, webbasierte Petri-Netz-Modellierungs- und Simulationsumgebung, mit der das beschriebene Logistikkolabor gut modelliert werden konnte.

Um auszuschließen, dass nur die persönlichen Präferenzen der Autoren die gewählte Modellierungssprache rechtfertigen, untersucht das folgende Kapitel III verwandte Ansätze für Modellierung und Simulation.

III. VERWANDTE ARBEITEN

Ursprünglich sind Petri-Netze als Stellen/Transitions-Netz (S/T) mit anonymen, nicht unterscheidbaren Token definiert, die den Zustand eines Systems angeben [8]. Dies schränkt die Menge der beschreibbaren Zustände jedoch erheblich ein, weswegen diverse Konzepte zur Anreicherung mit Informationen definiert wurden:

Prädikat/Transitions-Netze (Pr/T) interpretieren Stellen als Prädikate, die für die auf ihnen liegenden informations-tragenden Marken gelten. Transitionen verarbeiten und ändern diese Prädikate. Eine Transition kann dann für Marken schalten, die die Transitionsbedingung erfüllen und erzeugt neue Marken im Sinne ihrer ausgehenden Kanten. So können Interaktionen zwischen Marken modelliert werden. [5]

Colored Petri Nets (CPN) unterscheiden Token anhand von „Farben“, die dann aber letztlich auch im Sinne der Pr/T-Netze als Datensätze interpretiert werden. Auch hier erfolgt die Aktivierung der Transitionen getrennt, also für die individuellen Token. [6], [9]

Sollen technische Systeme oder Geschäftsprozesse modelliert werden, so spielt Zeit eine wichtige Rolle. Ohne Anspruch auf Vollständigkeit sollen die folgenden Ansätze genannt werden, die Petri-Netze mit Zeitkonzepten kombinieren: Time Petri Nets (TPN) [10] und Timed Petri Nets (TdPN) [11] assoziieren Zeit mit Transitionen, Time Place & Transition Nets (TPTN) [12] verbinden Zeit mit Stellen und deren Markierung, und Arc Timed Petri Nets (ATPN) [13], [14] verknüpfen Zeit mit den Kanten von Petri-Netzen.

Die genannten Formalismen haben ein gemeinsames Konzept, welches sie grundsätzlich von den ursprünglichen Petri-Netzen unterscheidet: Ihr Zustand beruht nicht nur auf Informationen, die durch ihre jeweiligen Markierungen gegeben sind, sondern auch auf einer Uhr. Zeitstempelnetze kodieren die Zeitinformation hingegen ebenfalls in der Markierung und sind somit eine Spezialisierung der Pr/T-Netze:

Zeitstempelnetze verwenden Token mit Zeitstempeln, die deren Entstehungszeitpunkt angeben. Transitionen können in Zeitfenstern schalten, wenn ihre eingehenden Kanten mit Blick auf diese Zeitstempel durchlässig sind. [15]

Erweiterte Zeitstempelnetze kombinieren die Konzepte der Zeitstempelnetze und Pr/T-Netze vollständig. [16]

Die Ansätze lassen sich ineinander überführen [17], [18], da sie abstrakte Zeiteinheiten verwenden. Für reale Anwendungen werden aber reale Zeitwerte benötigt, also Datums- und Zeitinformationen. Diese sind, wie im nächsten Abschnitt gezeigt, im P-S.C realisiert. Somit lassen sich Zeitinformationen mit allen anderen für Prozesse relevanten Daten verknüpfen.

Wertstromdiagramme (WSD) erlauben die Darstellung von Informations- und Materialflüssen, um Wertströme zu bewerten und hinsichtlich der Wartezeiten zu optimieren. Das Konzept beruht auf dem Produktionssystem von Toyota, wobei die Entwicklung der Methode bis in das Jahr 1914 zurückführt [2], [19]. Auch in Abbildung 2 wurde auf ein Wertstromdiagramm zur Visualisierung des Gesamtprozesses zurückgegriffen. Die Grenzen des Ansatzes mit Blick auf die Simulation wurde bereits in Abschnitt I diskutiert.

Business Process Model and Notation (BPMN) ist eine Sprache zur Modellierung von Geschäftsprozessen, die sich derzeit großer Beliebtheit erfreut. Aufgrund einfacher Basiskonzepte und einer anschaulichen Darstellung von Verantwortlichkeiten entwickelt BPMN Konzepte von Ereignisgesteuerten Prozessketten (EPK) und Flussdiagrammen weiter [20], [21]. Für BPMN gibt es, ähnlich wie auch bei EPK, Ansätze, um den Durchlauf einzelner Prozessinstanzen zu validieren [22]. Nicht berücksichtigt werden können hierbei aber die Prozessdaten oder auch wechselseitige Beziehungen zwischen verschiedenen Prozessinstanzen [23].

IV. NEUE MODELLIERUNGS- UND SIMULATIONSKONZEPTE IM P-S.C

Bei der Entwicklung konzeptioneller Modelle für die Prozesssimulation oder -ausführung werden neben der formalen mathematischen Basis auch Werkzeuge benötigt. Das Process-Simulation.Center (P-S.C) ist eine webbasierte Modellierungs- und Simulationsumgebung, die die Entwicklung von S/T- und Pr/T-Netzen unterstützt [24]. Hierzu ist es möglich, Stellen Datentypen zuzuordnen, um diese wie Tabellen in Datenbanken zu verwenden. Auch besteht die Möglichkeit über die Tupel einer Stelle Aggregate zu bilden. Spezielle Typen für Zeit und Datum sind wichtige Unterstrukturen für die Simulation von Prozessen in Produktion und Logistik und erweitern die genannten Ansätze für zeitgesteuerte Petri-Netze. Weiterführende Konzepte des P-S.C, wie die Abbildung der Aufbauorganisation mit Hilfe von Organigrammen, das Verbinden von Aufbau- und Ablauforganisation in Swimlanes sowie die Verknüpfung von Modellen mittels Prozesslandkarten, sind zwar für die praktische Anwendung der im Folgenden gezeigten Modelle wichtig, nicht aber für deren Verständnis. Auf eine genauere Darstellung wird hier daher verzichtet. Zu erwähnen ist aber, dass das P-S.C Knoten so zeichnet, dass sowohl deren Beschriftung als auch die Anzahl der Token innerhalb des Knotens dargestellt werden können, wodurch sich die Lesbarkeit der Modelle deutlich erhöht.

Im Gegensatz zu Relationaler Algebra und SQL, deren Operatoren auf Mengen angewendet werden und auch Mengen als Ergebnis liefern, verarbeitet das P-S.C die Tupel sequentiell, da auch Geschäfts- und Produktionsprozesse auftragsbezogen nacheinander abgearbeitet werden. Eine Entscheidung über die konkrete Reihenfolge wird lokal durch die Transitionen des Netzes getroffen.

Das P-S.C wird seit mehreren Jahren gemäß der Richtlinien des Design Science Research nach [25] entwickelt und wurde auch schon vor dem hier vorgestellten Szenario als Tool für problembasiertes und forschendes Lehren und Lernen in Bachelor- und Masterstudiengängen eingesetzt [26]. Studierende eines dualen Logistik-Studiengangs entwickelten unter Verwendung des P-S.C ein Simulationsmodell, um den Retourenprozess in ihrem Unternehmen zu reorganisieren [27].

V. TAKTGESTEUERTE SIMULATIONSMODELLE

Ziel der Simulation des *Box Game* ist es, die Auslastung der Lager im zeitlichen Verlauf beobachten zu können. Allerdings verwendet das P-S.C keine zeitlich getakteten Transitionen, da deren Uhren einen neuen Zustandsraum definieren würden und sich dieser nicht ohne weiteres mit den anderen Prozessdaten kombinieren ließe. Daher musste zunächst eine „Uhr“ bestehend aus einer zeittypisierten Stelle *clock* und einer adjazenten Transition *pulse* realisiert werden, wie in Abbildung 3 (oben links) gezeigt. Mit jedem Schalten der Transition wird das Zeit-Token auf der Stelle um eine Sekunde erhöht und *pulse* ist aufs Neue aktiviert.

Als ein Bedienkonzept können im P-S.C alle in einem Augenblick konfliktfrei aktivierten Transitionen gleichzeitig schalten, wodurch das Schalten aller anderen Transitionen im

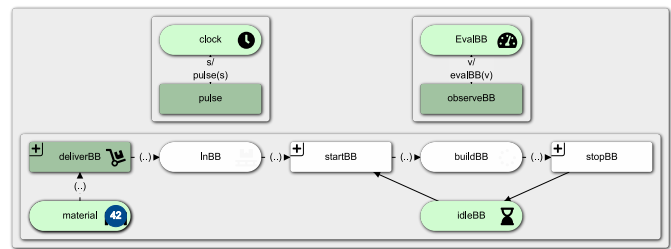


Abbildung 3. (Oben links) Einfache Petri-Netz-Uhr · (Oben rechts) Beobachternetz zur Evaluation des Vorlagers in (unten) · (unten) Montage großer Boxen als Beispiel eines einzelnen Arbeitsplatzes [4]

Netz mit dem Paar aus *clock* und *pulse* synchronisiert wird. Die aktuelle Uhrzeit auf *clock* kann dann mittels einer Aggregatfunktion, wie man sie aus Datenbanken kennt, ausgelesen werden. So lassen sich Arbeits- und Liegezeiten ausdrücken.

Das Grundmodell einer Arbeitsstation - in Abbildung 3 (unten) beispielhaft das Auffalten großer Boxen - besteht aus einer Stelle *inBB* für die Materialbereitstellung und einer Stelle *buildBB* für den Arbeitsplatz. Die Transition *deliverBB* führt der Arbeitsstation aus dem Eingangslager *material* Material und bei Bedarf Informationen über Losgrößen zu. Dies erfolgt nach dem FiFo-Prinzip, ebenso wie bei der Transition *startBB*, durch die das Material dem Arbeitsplatz zugewiesen wird. Hierzu sind die Arbeitsmaterialien nummeriert und zudem wird unter Rückgriff auf *clock* für das Material der Zeitpunkt gespeichert, wann dieses auf eine Stelle (Pufferlager oder Arbeitsstation) gelegt wird.

Wiederum unter Verwendung der *clock* wartet nun die Transition *stopBB* darauf, dass die für die Arbeitsstation bestimmte Bearbeitungszeit abgelaufen ist. Die hierzu formulierte Bedingung kann durch Aufklappen der Transition im Modell angezeigt werden, worauf hier aus Gründen der Übersichtlichkeit verzichtet wurde. Bei den im Modell hinterlegten Bearbeitungszeiten handelt es sich um Erfahrungswerte aus *Box Game*-Durchläufen mit Studierenden. Im Allgemeinen können mit dem Tool aber auch Einflussfaktoren wie verwendete Arbeitsmaterialien oder weitere Parameter der Arbeitsstation berücksichtigt werden.

Die Transition *stopBB* ist, wie in den Abbildungen 4 und 5 gezeigt, mit einem Zwischenspeicher verbunden, der als Puffer für die nachfolgende Arbeitsstation dient. Die Stelle *idleBB* dient als Semaphor für die exklusive Nutzung des Arbeitsplatzes für eine Box und als Zähler für die Stillstandszeiten des Arbeitsplatzes.

Schließlich wurden für das Modell Beobachternetze realisiert. Abbildung 3 (oben rechts) zeigt dies beispielhaft für die Produktion der großen Boxen. Hierüber lassen sich die mit dem Lager verbundenen Kosten ermitteln, indem auf der Stelle *inBB* mit jedem Schalten der Transition *observeBB* die in einem Schalttakt auf der Lager-Stelle *inBB* liegende Anzahl der Materialien erhöht wird. Auch wenn im vorgestellten Modell die Kosten für die unterschiedlichen Lager gleich bewertet werden, könnten die Materialkosten auch nach Verarbeitungsgrad differenziert und gestaffelt sein.

Durch das Zusammenspiel dieser Modelle lassen sich schwankende Lagerbestände, Engpässe und Lagerkosten über den Zeitverlauf hinweg beobachten. Symbole sollen hierbei das visuelle Verständnis des Gesamtmodells erhöhen.

Im Modell aus Abbildung 4 wurden diese Komponenten für alle Arbeitsstationen für eine Simulation der Push-Produktion zusammengestellt. Dies ergibt sich generisch aus den erklärten Konzepten. Hierbei wurden folgende Abkürzungen verwendet: *BB* steht für *Big Box*, *SB* für *Small Box*, *F* für *Finish*, das Zusammensetzen der Boxen, und *T* für *Test*, die abschließende Qualitätskontrolle.

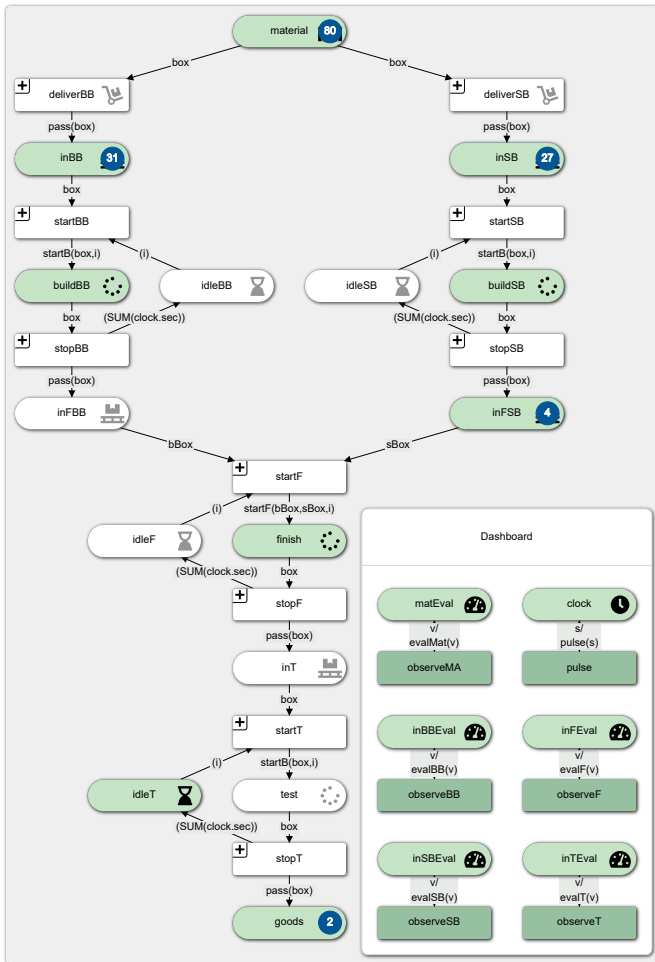


Abbildung 4. Taktgesteuertes Push-Modell [4]

Die Pull-Version des *Box Game*, wie sie in Abbildung 5 dargestellt ist, ist etwas komplexer, da zusätzliche Elemente benötigt werden, um das Pull-Prinzip zu implementieren. Dies geschieht durch zusätzliche Pull-Transitionen, auf denen der Inhalt der vorgelagerten Speicherplätze für alle drei Arbeitsplätze beobachtet wird. Sobald diese leer sind, wird eine Pull-Anforderung ausgegeben, die zu einer Lieferung führt.

Der Vergleich der Simulationsläufe zeigt erwartungsgemäß keine Unterschiede bzgl. der Gesamtbearbeitungszeit und der Leerlaufzeiten der verschiedenen Arbeitsplätze. Jedoch werden Unterschiede bei der Auslastung der Lagerplätze deutlich.

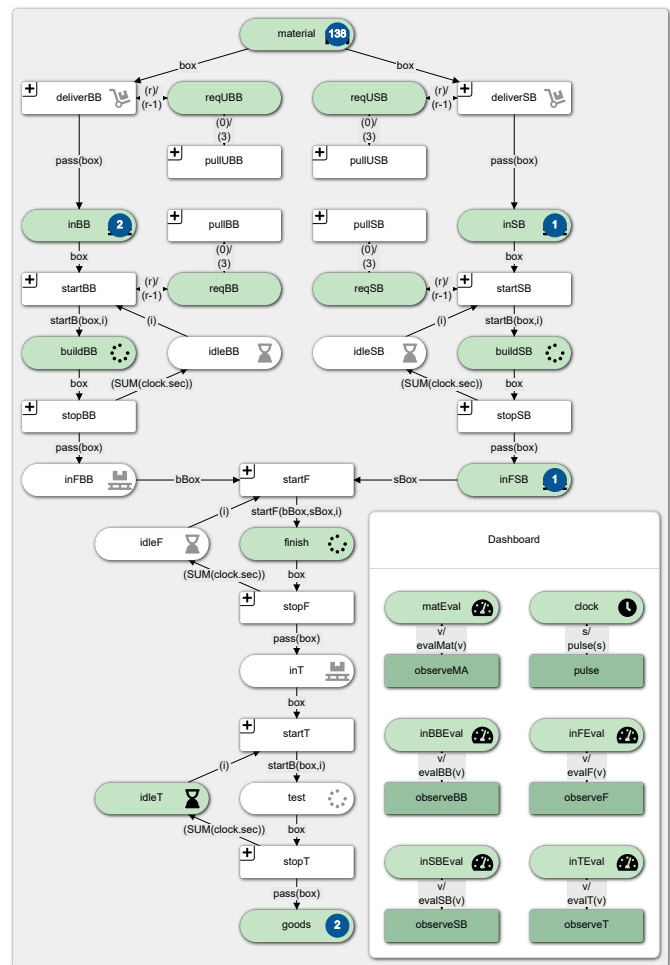


Abbildung 5. Taktgesteuertes Pull-Modell [4]

Das erste Modell nach dem Push-Prinzip zeigt deutlich den Nachteil dieses Ansatzes: große Zwischenlager und damit hohe Bestandskosten. Abbildung 6 (oben) zeigt den Bestand im Materiallager, den Pufferlagern und dem Fertigwarenlager während der Push-Simulation.

Abbildung 6 (unten) zeigt die Ergebnisse des Pull-Simulationslaufs mit sonst unveränderten Vorbedingungen. Die Zwischenlager sind deutlich weniger ausgelastet, da nur die Artikel in die Fertigungslinie eingebracht werden, die von nachgelagerten Stationen angefordert werden. Dies ist nicht weiter verwunderlich, da es ja gerade dem Ziel einer Just-in-Time-Produktion entspricht.

Für die beiden Diagramme in Abbildung 7 sind die unterschiedlichen Skalierungen der Y-Achsen zu beachten: Abbildung 7 (oben) zeigt die Kosten der Zwischenlager im Push-Modell, während Abbildung 7 (unten) das Gleiche für das Pull-Modell darstellt. Ebenfalls sind die über alle Zwischenlager kumulierten Kosten angegeben. Der Durchsatz ist in beiden Modellen identisch, was zu den gleichen Kosten innerhalb des Fertigwarenlagers führt und somit nicht weiter ausgewertet werden muss.

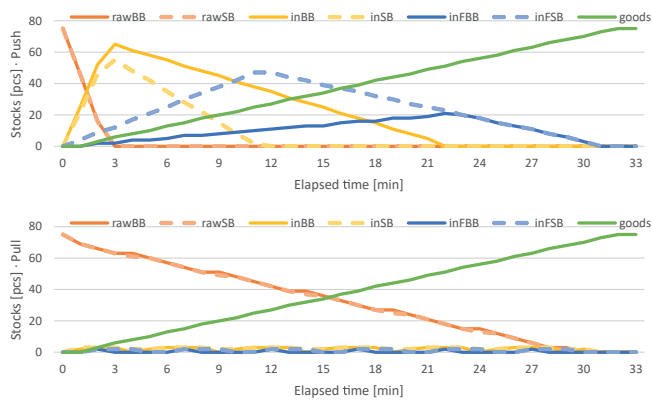


Abbildung 6. Lagerbestände für Push (oben) und Pull (unten) [4]

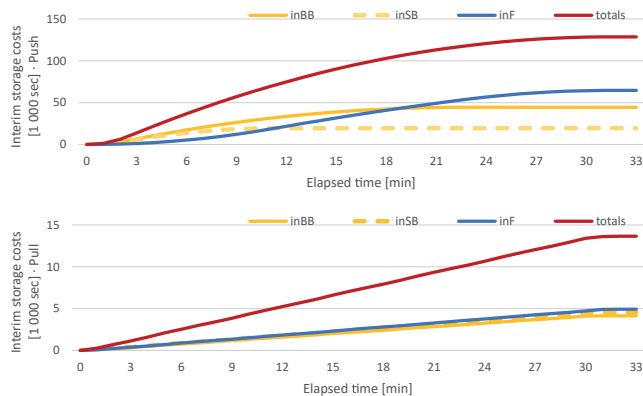


Abbildung 7. Lagerkosten je Zwischenlager und kumuliert für Push (oben) und Pull (unten) [4]

Die Differenz bezüglich des Materialeingangslager zeigt die möglichen Einsparungen durch Just-In-Time-Bestellungen bei den Lieferanten. Zudem kann beobachtet werden, wie durch Einführung des Pull-Prinzips die bewerteten Bestandskosten auf den Zwischenlagern sinken.

Selbstverständlich könnten diese Modelle noch um weitere Aspekte wie etwa die Transportkosten erweitert werden, wobei kleinere Losgrößen in der Regel mit höheren Transportkosten einhergehen. Anders als das hier gezeigte Modell suggeriert, würde dann deutlich, dass die für die Zwischenlagerkosten optimale Losgröße 1 nicht unbedingt gleichzeitig auch die global optimale Lösung darstellt.

VI. EIN ALTERNATIVER MODELLIERUNGSANSATZ

Durch die beschriebene Modellierung und Simulation können die Fluktuationen der Bestände während der Produktion im Zeitverlauf beobachtet und so die Vor- und Nachteile von Push und Pull ideal visualisiert werden. Für die Studierenden im Logistiklabor stellt die Simulation daher eine anschauliche Erweiterung ihrer persönlichen Erfahrungen dar.

Allerdings benötigt eine vollständige Simulation 1902 Schaltakte. Dies entspricht ebenso vielen Sekunden (oder fast 32 Minuten) im realen Spiel. Deren Berechnung dauert auf einem iMac mit 4 GHz Quad-Core Intel Core i7 Prozessor und 16 GB RAM in Chrome etwa 8234 Millisekunden und die Berechnungsdauer einer größeren Zeitspanne, etwa eines Arbeitstages, würde linear hierzu ansteigen.

Tatsächlich lässt sich die Simulationszeit drastisch reduzieren, wenn das Ergebnis, also die Bewertung der Lagerbestände, wichtiger ist als ihre Visualisierung im Zeitverlauf. In diesem Fall würde es ausreichen, neue Zustände nur in den Momenten der Zustandsänderungen zu berechnen, also wenn eines der Materialien an eine Arbeitsstation oder ein Lager weitergereicht wird. Die „virtuellen Sekunden“, in denen beim *Box Game* nichts passiert, könnten übersprungen werden.

Damit kann im vorliegenden Beispiel die Simulation von 1902 auf nur 79 Schritte für Push und 228 Schritte für Pull reduziert werden. Auf dem genannten Rechner dauern die Simulationen dann nur noch etwa 315 bzw. 923 Millisekunden. Diesen zweiten Ansatz kann man daher als ereignisgesteuerte Simulation bezeichnen.

Im Gegenzug muss dann die Visualisierung der Simulationsergebnisse in einem separaten Arbeitsschritt erfolgen. Außerdem ist die Entwicklung eines ereignisgesteuerten Modells im Vergleich zu einem taktgesteuerten Modell anspruchsvoller, weil nicht nur das Verhalten für einen spezifischen Augenblick abzubilden ist, sondern der nächste relevante Zeitpunkt präjudiziert werden muss. Für die zukünftige Modellentwicklung resultiert hieraus die folgende Anregung:

Wähle eine taktgesteuerte Simulation, wenn entweder eine getaktete Visualisierung der Systemzustände benötigt wird, oder wenn der Rechner schnell genug für die wenigen Simulationen ist, die für das modellierte System durchgeführt werden müssen.

Wähle eine ereignisgesteuerte Simulation, wenn eine hohe Simulationsgeschwindigkeit aufgrund der Komplexität des modellierten Systems erforderlich ist, wenn schnelle Antworten in der Produktion benötigt werden, oder wenn eine große Anzahl von Variationen des Produktionsplans oder der Eingangsdaten verglichen werden müssen. Im Allgemeinen gilt: Je häufiger ein bestimmtes Modell simuliert werden muss, desto mehr lohnt es sich, ein ereignisgesteuertes Modell anstelle eines taktgesteuerten Modells zu entwickeln.

VII. FAZIT UND ZUKÜNFTIGE ARBEITEN

Die Modellierungsaufgaben, die für diese Arbeit zu lösen waren, führten zu einigen neuen Erkenntnissen bei der Entwicklung konzeptioneller Modelle für zeitdiskrete Systeme. Für die Autoren haben sich Best Practices herauskristallisiert, die aus den folgenden Schritten bestehen:

1. Definiere Datentypen für die Bestände sowie andere Datenobjekte wie Kanbans oder Ressourcen und initialisiere die Stellen entsprechend der Startbedingungen.

2. Erweitere das Modell um Transitionen für den Beginn und das Ende bestimmter Aufgaben, wie z.B. die Lieferung von Rohstoffen, den Bau oder das Testen einer Box.
3. Identifiziere das nächste zu bearbeitende Werkstück und den Zeitpunkt, an dem dies geschehen wird. Dies ermöglicht auch die Umsetzung verschiedener Priorisierungsstrategien.
4. Beginne mit der Modellierung des einfacheren Push-Prinzips und erweitere dieses Modell um Pull-Prinzipien.

Außerdem erscheint die Entwicklung eines taktgesteuerten Modells als Vorstufe für die Entwicklung eines ereignisgesteuerten Modells sinnvoll zu sein, falls ein solches aus den oben genannten Gründen benötigt wird:

1. Entwickle zunächst das taktgesteuerte Modell.
2. Beobachte die Ereignisse, die zu Zustandsänderungen führen, mit Hilfe des taktgesteuerten Modells und leite aus dieser Beobachtung das ereignisgesteuerte Modell ab.
3. Suche nach einer geeigneten Visualisierung der Simulationsergebnisse.

Vorausgesetzt, man verfügt über ein geeignetes Werkzeug wie das P-S-C, sind die Möglichkeiten dessen, was Modelliererinnen und Modellierer abbilden und simulieren können, nur durch ihre Vorstellungskraft begrenzt. Eine Erweiterung um Restriktionen, durch Ressourcen oder alternative Produktionsstrategien können effektiv und effizient ins Modell integriert werden. Nun sollen noch die Möglichkeiten zur Visualisierung der Simulationsergebnisse im P-S-C erweitert werden - diese wurden in diesem Beitrag noch mit Hilfe externer Tools erstellt. Eine entsprechende Forschungsagenda ist bereits formuliert. [28]

Die akademische Community ist herzlich eingeladen, das P-S-C und die hier vorgestellten Modelle auszuprobieren. Treten Sie gerne mit den Autoren in Kontakt!

LITERATUR

- [1] C. Caserio and S. Trucco, *Enterprise Resource Planning and Business Intelligence Systems for Information Quality*. Cham, Switzerland: Springer, 2018.
- [2] T. Ohno, *Toyota Production System*. Milton Park, UK: Taylor & Francis, 1988.
- [3] S. Haag, L. Zakfeld, C. Simon, and C. Reuter, "Event Triggered Simulation of Push and Pull Processes," in *SIMUL 2020: The Twelfth International Conference on Advances in System Simulation*, L. Parra, Ed., Porto (Portugal), 2020, pp. 68–73.
- [4] C. Simon, S. Haag, and L. Zakfeld, "Clock Pulse Modeling and Simulation of Push and Pull Processes in Logistics," in *SIMMaApp: Special Track at SIMUL 2020: The Twelfth International Conference on Advances in System Simulation*, F. Herrmann, Ed., Porto (Portugal), 2020, pp. 31–36.
- [5] H. J. Genrich and K. Lautenbach, "System Modelling with High-Level Petri Nets," *Theoretical Computer Science*, vol. 13, 1981.
- [6] K. Jensen, *Coloured Petri-Nets*, 1st ed. Berlin: Springer, 1992.
- [7] Petri Nets World, "Petri Nets Tools Database Quick Overview," <https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html> (last accessed 20.09.2020), 2020.
- [8] W. Reisig, *Understanding Petri Nets*. Berlin: Springer, 2013.
- [9] M. Montali and A. Rivkin, "From DB-nets to Coloured Petri Nets with Priorities (Extended Version)," *CoRR*, vol. abs/1904.00058, 2019. [Online]. Available: <http://arxiv.org/abs/1904.00058>
- [10] P. Merlin, "The Time-Petri-Net and the Recoverability of Processes," University California, Irvine, Tech. Rep., 1974.
- [11] C. Ramchandani, "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets," MIT, Project MAC, Technical Report 120, 1974.
- [12] J. Sifakis, "Use of petri nets for performance evaluation," in *Measuring, modelling and evaluating computer systems*, ser. IFIP, H. Beilner and E. Gelenbe, Eds., North Holland Publ. Co., 1977, pp. 75–93.
- [13] R. König and L. Quäck, *Petri-Netze in der Steuerungs- und Digitaltechnik*. München, Wien: Oldenbourg Verlag, 1988.
- [14] H.-M. Hanisch, *Petri-Netze in der Verfahrenstechnik*. München: Oldenbourg, 1992.
- [15] 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.
- [16] C. Simon, "Developing Software Controllers with Petri Nets and a Logic of Actions," in *IEEE International Conference on Robotics and Automation, ICRA 2001*, Seoul, Korea, 2001.
- [17] K. Jensen, "High-Level Petri Nets," *Informatik-Fachberichte*, vol. 66, pp. 166–180, 1983.
- [18] L. Popova-Zeugmann, *Time and Petri Nets*. Berlin: Springer, 2013.
- [19] C. E. Knoeppel, *Installing Efficiency Methods*. The Engineering Magazine, 1915.
- [20] BPMI, "BPMN 1.0 - Business Process Model and Notation," <https://www.omg.org/spec/BPMN/> (last accessed 20.09.2020), 2004.
- [21] OMG, "BPMN 2.0 - Business Process Model and Notation," <http://www.bpmn.org/> (last accessed 20.09.2020), 2011.
- [22] O. Kloos, *Generierung von Simulationsmodellen auf der Grundlage von Prozessmodellen*. Ilmenau: Universitätsverlag Ilmenau, 2014.
- [23] W. M. P. van der Aalst, "Geschäftsprozessmodellierung: Die Killer-Applikation für Petrinetze," in *Informatik Spektrum*, vol. 37, no. 3, 2014, pp. 191–198.
- [24] C. Simon, "Web-Based Simulation Of Production Schedules With High-Level Petri Nets," in *32rd International ECMS Conference on Modelling and Simulation (ECMS 2018)*, L. Nolle, A. Burger, C. Tholen, J. Werner, and J. Wellhausen, Eds. Wilhelmshaven, Germany: SCS Europe, 2018, pp. 275–281.
- [25] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, mar 2004. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2017212.2017217>
- [26] C. Simon and S. Haag, "A Case-Study to Teach Process-Aware Information Systems," *EMISA Forum: Proceedings of the SIG Enterprise Modelling and Information Systems Architectures of the German Informatics Society*, vol. 40, pp. 9–10, 2020.
- [27] —, "Simulatable Reference Models To Transform Enterprises For The Digital Age - A Case Study," in *ECMS 2020: 34th International ECMS Conference on Modelling and Simulation*, M. Steglich, C. Müller, G. Neumann, and M. Walther, Eds., 2020, pp. 294 – 300.
- [28] C. Simon, S. Haag, and L. Zakfeld, "Research-Agenda for Process Simulation Dashboards," in *ECMS 2021 : 35th International ECMS Conference on Modelling and Simulation*, 2021, pp. 243–249.