

FACHWISSENSCHAFTLICHE PROJEKTARBEIT

Modellierung und Simulation selbststeuernder miniaturisierter Logistiksysteme

Bearbeitet von: Lena Büchner

Studiengang: Logistik
Matrikel-Nr.: 166659

Ausgegeben am: 09.05.2017
Eingereicht am: 08.08.2017

Prüfer: Univ.-Prof. Dr.-Ing. Markus Rabe
Betreuer: Dipl.-Geoinf. Maik Deininger

Inhaltsverzeichnis

Modellierung und Simulation selbststeuernder miniaturisierter Logistiksysteme

Inhaltsverzeichnis	I
Abbildungsverzeichnis	II
Tabellenverzeichnis	III
1 Einleitung	1
2 Selbstgesteuertes produktionslogistisches System	3
2.1 Begriffsdefinition produktionslogistisches System	3
2.1 Selbststeuerung	5
2.1.1 Voraussetzungen für eine Selbststeuerung	6
2.1.2 RFID-Technologie als Möglichkeit zur Umsetzung einer Selbststeuerung	7
3 Modellierung und Simulation	11
3.1 Modellierung	11
3.2 Simulation.....	14
4 Modellierung und Simulation von Selbststeuerungen in Automod	17
4.1 Beschreibung des Tools Automod.....	17
4.2 Vorgehensweise zur Modellierung eines selbststeuernden Systems in Automod	18
4.3 Besonderheiten bei der Modellierung von Systemen in Automod	19
4.4 Beispielhafte Umsetzung.....	20
4.4.1 Systembeispiel 1: Automobilfertigung	20
4.4.2 Systembeispiel 2: Kommissionierung	25
5 Zusammenfassung und Ausblick	29
Literaturverzeichnis	31
Anhang	33
Anhang A: Modell Systembeispiel 1	33
Anhang B: Modell Systembeispiel 1	34
Anhang C: Source File Systembeispiel 1	35
Anhang D: Source File Systembeispiel 2.....	37

Abbildungsverzeichnis

Abbildung 2-1: Produktion als Kombinationsprozess	3
Abbildung 2-2: Grundlagen der Systemtheorie	5
Abbildung 2-3: Positive Emergenz in Produktionslogistischen Systemen	7
Abbildung 2-4: Komponenten eines RFID-Systems	8
Abbildung 3-1: Hauptmerkmale von Modellen.....	11
Abbildung 3-2: Beziehungen zwischen Original, Mensch und Modell	13
Abbildung 3-3: Beziehungsdreieck.....	14
Abbildung 3-4: Vom realen Modell zum Simulationsmodell und wieder zurück.....	15
Abbildung 4-1: Fahrzeugvarianten	20
Abbildung 4-2: Produktionsprozess Automobilfertigung.....	22

Tabellenverzeichnis

Tabelle 1: Auftragsvariation	25
Tabelle 2: Queues und Resources	22
Tabelle 3: Prozesse	24
Tabelle 4: Loads und Attribute	24
Tabelle 5: Notwendige Bausteine.....	27

1 Einleitung

Die fortschreitende Globalisierung führt zu einer kontinuierlichen Steigerung der Marktdynamik. Diese Entwicklung geht mit steigender Individualisierung von Produkten, verstärkter Kundenorientierung und kürzeren Produktlebenszyklen einher. Die Trends führen zu einem Wachstum der Komplexität logistischer Systeme, welche immer höheren Ansprüchen an Flexibilität, Robustheit und Anpassungsfähigkeit gerecht werden müssen. Herkömmliche Logistiksysteme können diese Anforderungen zunehmend weniger erfüllen (Freitag et al, 2004; de Beer, 2008, S. 1-2).

Eine Möglichkeit, der steigenden Komplexität zu begegnen, ist die Umstellung von zentralen Steuerungen auf dezentrale Selbststeuerungen, in denen Produkte eigenständig ihren Weg durch das System ermitteln (de Beer, 2008, S. 2; Scholz-Reiter et al., 2005). Durch Selbststeuerungen ist es möglich die Komplexität logistischer Systeme zu senken und sie gleichzeitig schneller, flexibler und robuster zu gestalten. Somit kann deren Gesamtverhalten im Hinblick auf logistische Zielsetzungen verbessert werden (de Beer, 2008, S. 35-36).

In der industriellen Praxis bewährt sich die auf Simulation basierende Gestaltung, Planung, Implementierung und Optimierung komplexer Systeme zunehmend. Daher liegt es nahe, Materialflusssimulationen auch für selbstgesteuerte Systeme anzuwenden (Eley, 2012, S. 5).

Das Hauptziel der vorliegenden Projektarbeit ist die Entwicklung und Umsetzung eines Konzepts, anhand dessen sich Selbststeuerungen produktionslogistischer Systeme modellieren und simulieren lassen. Für die Umsetzung sollen zwei unterschiedliche Systembeispiele erarbeitet werden, um zu plausibilisieren, ob das zu erarbeitende Konzept universell für verschiedene produktionslogistische Systeme anwendbar ist.

Nebenziel dieser Arbeit ist es, literaturbasiert zu erläutern und abzugrenzen, worum es sich bei selbststeuernden produktionslogistischen Systemen handelt, wie solche definiert sind und was deren Merkmale und Besonderheiten sind. Dadurch wird das Anwendungsfeld des Konzeptes aufgezeigt. Darüber hinaus gilt es als Basis für die Konzeptentwicklung zu analysieren, welche Merkmale Modellierung und Simulation aufweisen und welche Besonderheiten bei deren Einsatz zu berücksichtigen sind.

Für die vorliegende Projektarbeit ist die Verwendung von Automod als Simulationstool vorgesehen. Deshalb ist im Zuge dieser Arbeit darzulegen, wie Automod aufgebaut ist und welche Besonderheiten bei der Modellierung und Simulation selbstgesteuerter produktionslogistischer Systeme in diesem Tool zu berücksichtigen sind.

Den Einstieg in diese Arbeit bilden umfassende Literaturrecherchen bezüglich der Definition und Abgrenzung produktionslogistischer Systeme und über Möglichkeiten und Voraussetzungen von Selbststeuerungen. Weitere Recherchen dienen der Analyse von Merkmalen und Besonderheiten bei der Modellierung und Simulation derartiger Systeme. Der RFID-Technologie kommt für Selbststeuerungen eine besondere Bedeutung zu, weshalb diese separat erörtert wird. Es wird aufgezeigt was deren grundsätzliche Vor- und Nachteile sind, und dargestellt, welchen Beitrag diese zur Realisierung effektiver selbststeuernder Systeme leisten können. Aus den Erkenntnissen der Literaturrecherche geht hervor, welche Systeme für die Anwendung des Konzepts geeignet sind. Darüber hinaus dienen sie als Grundlage für die dessen Erarbeitung.

Als weitere Basis für das Konzept folgt eine detaillierte Vorstellung des Simulationstools Automod. Diese schließt ein, wie sich das Programm aus verschiedenen Arbeitsbereichen aufbaut und wie dessen Bausteine definiert, abgegrenzt und zu verwenden sind. Im Hinblick auf die Konzeptumsetzung wird aufgezeigt, welche Besonderheiten bei der Modellierung und Simulation von selbststeuernden produktionslogistischen Systemen in Automod zu berücksichtigen sind.

Auf den zuvor beschriebenen Grundlagen erfolgt die Entwicklung eines Konzepts, welches basierend auf Automod eine Vorgehensweise aufzeigt, die es erlaubt selbststeuernde Systeme zu modellieren und zu simulieren. Um die Eignung und Anwendbarkeit des entwickelten Konzepts darzulegen, werden anhand dieses Konzepts zwei fiktive produktionslogistische Systeme modelliert und simuliert.

2 Selbstgesteuertes produktionslogistisches System

Im ersten Teil dieses Kapitels wird der Begriff „produktionslogistisches System“ erklärt und eingeordnet. Im zweiten Teil wird das Prinzip der Selbststeuerung erläutert. Das Kapitel dient dazu, den aktuellen Forschungsstand darzustellen und bildet die fachliche Basis für das im weiteren Verlauf entwickelte Konzept.

2.1 Begriffsdefinition produktionslogistisches System

Die Produktionslogistik setzt sich aus den Begriffen „Produktion“ und „Logistik“ zusammen, auf die zu Beginn des Kapitels eingegangen wird. Darauf basierend folgt die Einordnung der Produktionslogistik, welche am Ende des Kapitels systemtheoretisch betrachtet wird.

Produktion

Produktion bedeutet Kombination von Produktionsfaktoren. Produktionsfaktoren bezeichnen alle Güter, die zur Herstellung eines gewünschten Produktes notwendig sind. Dies können beispielsweise menschliche Arbeitsleistung, Arbeits- und Betriebsmittel oder Werkstoffe sein (Arnold et al., 2008, S. 295). Durch die Kombination von verschiedenen Produktionsfaktoren werden beispielsweise Rohstoffe verbraucht und Güter oder Dienstleistungen hergestellt (Abbildung 2-1) (Bloech et al., 2014, S. 3.).

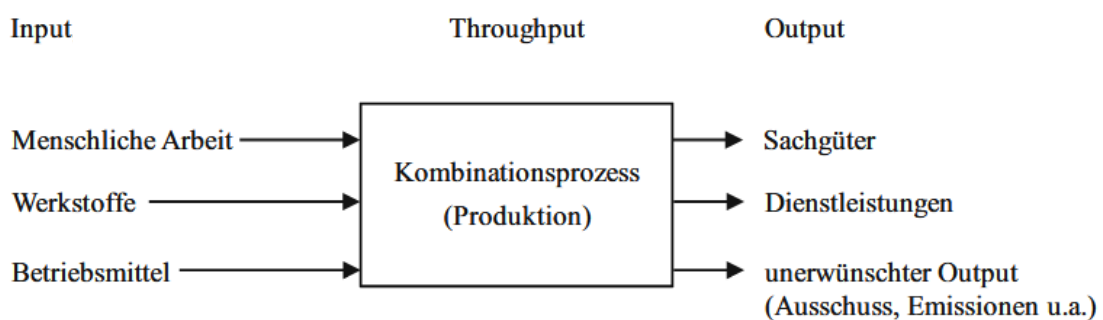


Abbildung 2-1: Produktion als Kombinationsprozess (Bloech et al., 2014, S. 3)

Logistik

Als Kernfunktionen der Logistik gelten Transport, Umschlag und Lagerung von Gütern. Mit diesen Kernfunktionen gehen die Nebenfunktionen Beschaffung, Produktion und Verteilung einher. Diese Funktionen und durch sie ausgelöste Prozesse der Logistik sind dabei auf den Marktbedarf ausgerichtet.

Logistik zeichnet sich dadurch aus, dass sie nicht nur isolierte Einzelprozesse,

sondern alle Prozesse eines Systems gleichzeitig betrachtet (Arnold et al., 2008, S. 295).

Des Weiteren hat die Logistik eine unternehmensübergreifende Perspektive. Das bedeutet, dass sie die gesamte Wertschöpfungskette betrachtet und sich nicht nur auf eine Unternehmung fokussiert, sondern auch Vorlieferanten und Kunden berücksichtigt (Wannenwetsch, 2014, S. 19-20).

Produktionslogistik

Die Produktionslogistik ist ein Subsystem der Logistik und beschreibt alle logistischen Funktionen, die den für die Produktion von Gütern notwendigen Informations-, Material- und Wertfluss gewährleisten. Dieses schließt Planungs-, Steuerungs- und Kontrollfunktionen mit ein (de Beer, 2008, S. 13-15; Tentrop, 2011, S. 37). Die Produktionslogistik ist hinsichtlich des Materialflusses zwischen Beschaffungslogistik und Distributionslogistik einzuordnen (Tentrop, 2011).

Ziel der Produktionslogistik ist es, Kundenwünsche wirtschaftlich und termingetreu zu erfüllen und dabei Produktionsfaktoren optimal einzusetzen. Sie orientiert sich dabei am Markt und am Produkt (Tentrop, 2011, S. 38; Arnold et al., 2008, S. 296). Um dieses Ziel bestmöglich zu erreichen, handelt die Produktionslogistik nach den von Pawellek (2012, S. 467) genannten Teilzielen: „Orientierung der Logistikleistung am Service, Reduzierung der Durchlaufzeit, kundenauftragsgesteuerte Produktion, Reduzierung der Komplexität in der Produktion, Organisation und Auftragsabwicklung.“

Systemtheoretische Darstellung

In der theoretischen Darstellung eines Systems sind dessen Elemente über Wirkungsverknüpfungen miteinander verbunden und können sich dadurch gegenseitig beeinflussen. Elemente und Verbindungen bilden die Systemstruktur und beschreiben den statischen Teil eines Systems. Das Systemverhalten, der dynamische Teil, wird durch Prozesse beschrieben. Ein System ist zwar von der Umwelt abgegrenzt, gleichzeitig aber durch Ein- und Auswirkungen mit ihr verbunden (de Beer, 2008, S. 13-15; Tentrop, 2011, S. 20-21). Verdeutlicht wird dies in Abbildung 2-2.

Sowohl die Produktion als auch die Logistik können systemtheoretisch beschrieben werden, woraus de Beer (2008, S.23) schlussfolgert, dass die Produktionslogistik auf gleiche Weise als System darstellbar ist.

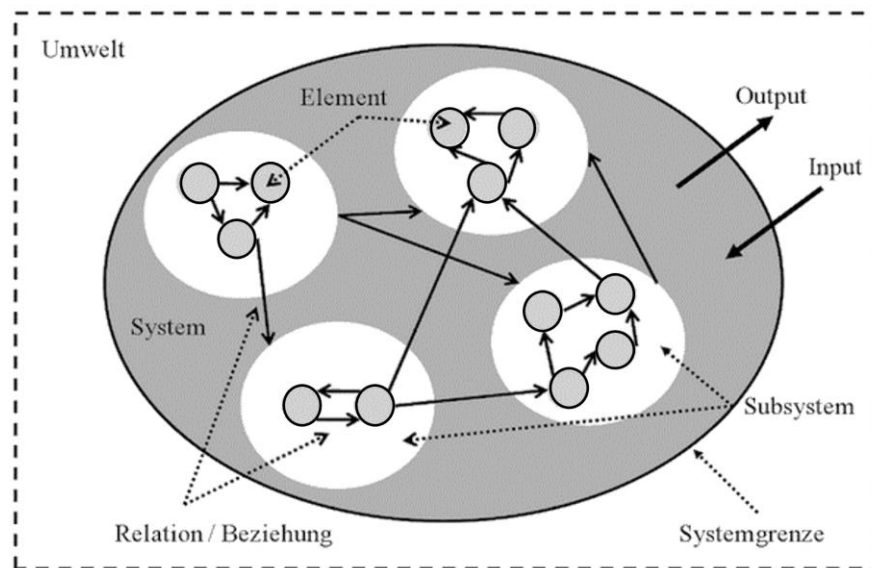


Abbildung 2-2: Grundlagen der Systemtheorie (Tentrop, 2011, S. 21)

Komplexität produktionslogistischer Systeme

Reale produktionslogistische Systeme zeichnen sich durch eine hohe Komplexität aus. Diese besteht dabei aus einer statischen und einer dynamischen Komponente. Für die statische Komplexität sind Anzahl und Vielfältigkeit der Elemente und Beziehungen ausschlaggebend. Die dynamische Komplexität beschreibt die Veränderungen im produktionslogistischen System. Eine zusätzliche Verstärkung der dynamischen Komplexität tritt durch hohe Veränderungsgeschwindigkeiten auf (Kolditz, 2009, S. 9-10). Ein weiterer Aspekt, der die Komplexität steigert, ist die Anzahl unternehmensübergreifender Kooperationen, wodurch die Bereitstellung notwendiger Informationen meist über eine zentrale Verwaltung mit entsprechend vielen Schnittstellen erfolgt. (Freitag et al., 2004). Damit Unternehmen in ihrer geschäftlichen Tätigkeit bestehen können, müssen Möglichkeiten gefunden werden, mit einer solchen Komplexität umzugehen. Eine vielversprechende Option ist der Wechsel von einer zentralen zu einer dezentralen Steuerung. Eine Möglichkeit, diesen Wechsel umzusetzen, stellt die Selbststeuerung der Produktion dar, welche im nachfolgenden Kapitel erläutert wird (Scholz-Reiter et al., 2005).

2.1 Selbststeuerung

Bei der Selbststeuerung handelt es sich um eine dezentrale Steuerung. Diese zeichnet sich dadurch aus, dass logistische Objekte sich selbst via dezentrale Entscheidungsfindung entlang des Materialflusses steuern.

Logistische Objekte bezeichnen dabei einerseits materielle Elemente wie Maschinen und Güter, andererseits immaterielle Elemente, wie Aufträge und

Dienstleistungen. Der Sonderforschungsbereich 637 der Universität Bremen entwickelte eine Definition für Selbststeuerungen:

„Selbststeuerung beschreibt Prozesse dezentraler Entscheidungsfindung in heterarchischen Strukturen. Sie setzt voraus, dass interagierende Elemente in nicht-deterministischen Systemen die Fähigkeit und Möglichkeit zum autonomen Treffen von Entscheidungen besitzen. Ziel des Einsatzes von Selbststeuerung ist eine höhere Robustheit und positive Emergenz des Gesamtsystems durch eine verteilte, flexible Bewältigung von Dynamik und Komplexität.“ (Scholz-Reiter et al., 2006)

Ergänzend dazu wurde eine ingenieurwissenschaftliche Definition entwickelt:

„Selbststeuerung logistischer Prozesse ist gegeben, wenn das logistische Objekt Informationsverarbeitung, Entscheidungsfindung und -ausführung selbst leistet.“ (Scholz-Reiter et al., 2006)

Der Unterschied zur Fremdsteuerung besteht in der dezentralen Entscheidungsfindung. Bei einer Fremdsteuerung werden alle Informationen und Daten zentral gesammelt und Entscheidungen werden anschließend zentral veranlasst.

Die Selbststeuerung bietet die Möglichkeit, Systeme robuster und flexibler zu gestalten und somit das Gesamtverhalten des Systems im Hinblick auf logistische Zielerreichung zu verbessern und effizienter zu gestalten (de Beer, 2008, S. 35-36).

2.1.1 Voraussetzungen für eine Selbststeuerung

Wie in der Definition bereits beschrieben, treffen selbstgesteuerte Objekte eigene Entscheidungen. Um diese Entscheidungen treffen zu können, benötigen die Objekte Informationen und Daten von und über weitere Objekte in ihrem Umfeld, wie zum Beispiel deren Standorte und Bearbeitungszeiten. Dies setzt voraus, dass die Objekte miteinander kommunizieren können. Kommunikation ist demnach eine notwendige Voraussetzung für Selbststeuerungen (Elbert, 2006).

Außerdem benötigen die autonomen Objekte eine eigene Intelligenz, um notwendige Daten aufnehmen, beurteilen und Entscheidungen treffen zu können. Durch moderne Informations- und Kommunikationstechnologien, wie Sensorik, Ortung, datentechnische Vernetzung und autonome Fähigkeit zur Datenverarbeitung, lässt sich logistischen Objekten dezentrale Intelligenz zuordnen (Scholz-Reiter et al., 2005).

Eine wichtige Voraussetzung um die Kommunikation der Objekte untereinander zu ermöglichen, sind standardisierte Schnittstellen. Diese bilden ebenfalls eine Voraussetzung für eine hohe Flexibilität des Systems (Voigt et al., 2006).

Damit sich Effizienz, Flexibilität und Robustheit eines Systems verbessern, ist positive Emergenz notwendig. Emergenz bedeutet, dass ein gewünschtes Verhalten auf globaler Ebene auftritt, welches durch einfaches Verhalten und Zusammenwirken von Elementen auf lokaler Ebene hervorgerufen wird. Emergente Systemeigenschaften ergeben sich aus einer Kombination der Eigenschaften von einzelnen Elementen. Emergenz wird als positiv bezeichnet, wenn die emergenten Systemeigenschaften zu einer besseren logistischen Zielerreichung führen (de Beer, 2008, S. 37-38). Abbildung 2-3 verdeutlicht dies im produktionslogistischen Kontext.

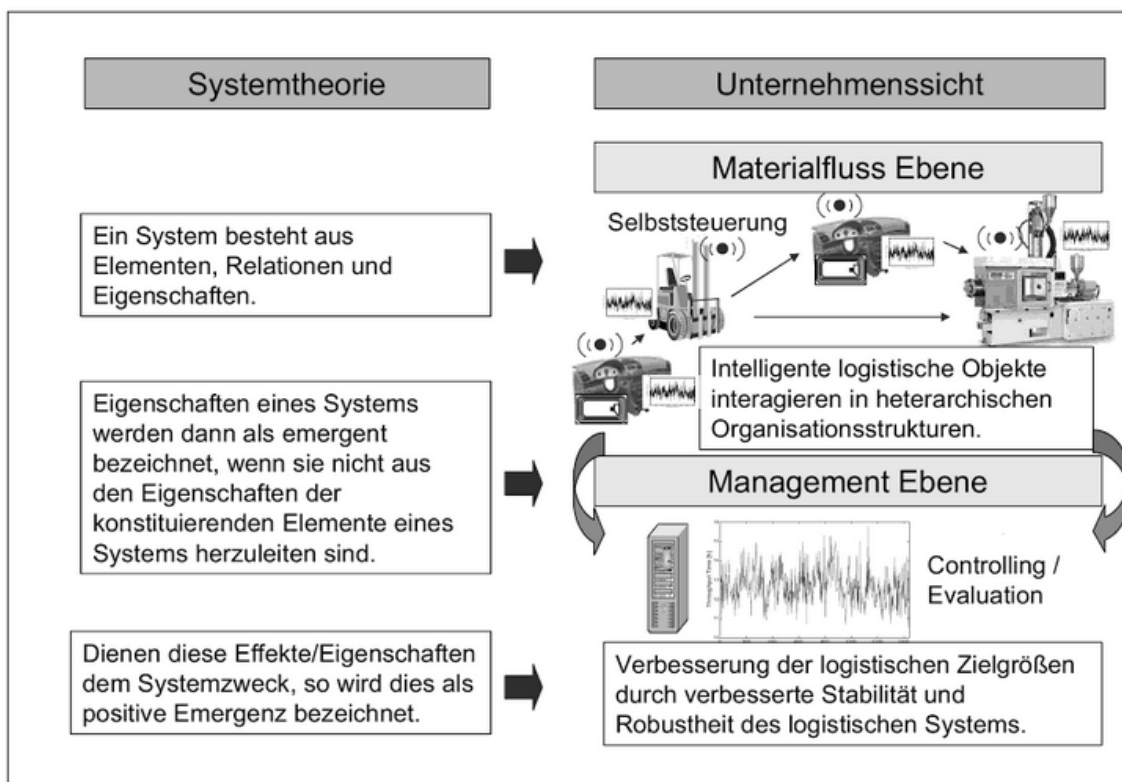


Abbildung 2-3: Positive Emergenz in Produktionslogistischen Systemen
(de Beer, 2008, S. 38)

2.1.2 RFID-Technologie als Möglichkeit zur Umsetzung einer Selbststeuerung

Eine Selbststeuerung lässt sich durch Verwendung der RFID-Technologie (Radio-Frequency-Identification) in einem produktionslogistischen System umsetzen. Nachfolgend wird zunächst die Technologie erklärt und anschließend deren Eignung für die Umsetzung einer Selbststeuerung untersucht.

Eine erste Möglichkeit, dezentrale Steuerungsarchitekturen und einen Materialfluss begleitenden Informationsfluss zu verwirklichen, bot die RFID-Technologie (Günther et al., 2006). Eine weitere Möglichkeit stellt der Einsatz von Multi-

Agenten-Systemen dar. Dabei interagieren mehrere Softwareagenten miteinander und vollenden autonom die ihnen gestellten Aufgaben, indem sie Informationsflüsse dezentral steuern. So entscheiden sie zum Beispiel über den Transport eines Guts und geben entsprechende Befehle an das Fördermittel weiter (Schuh, 2007, S. 761-766).

Die RFID-Technologie ist den automatischen Identifikationstechnologien zuzuordnen. Ein RFID-System besteht aus drei Komponenten: Einem Rechner, einem Lesegerät und einem Transponder, dem „Tag“. Der Rechner sendet Vorgaben an das Lesegerät, wie zum Beispiel: „Identifikationsnummer auslesen“. Die Vorgaben werden vom Lesegerät in ein Signal umgewandelt, welches im Anschluss über ein vom Lesegerät erzeugtes Wechselfeld an den Transponder übermittelt wird. Zum Senden und Aufnehmen der Signale dienen Kopplungselemente, wie Spulen oder Antennen (Abbildung 2-4).

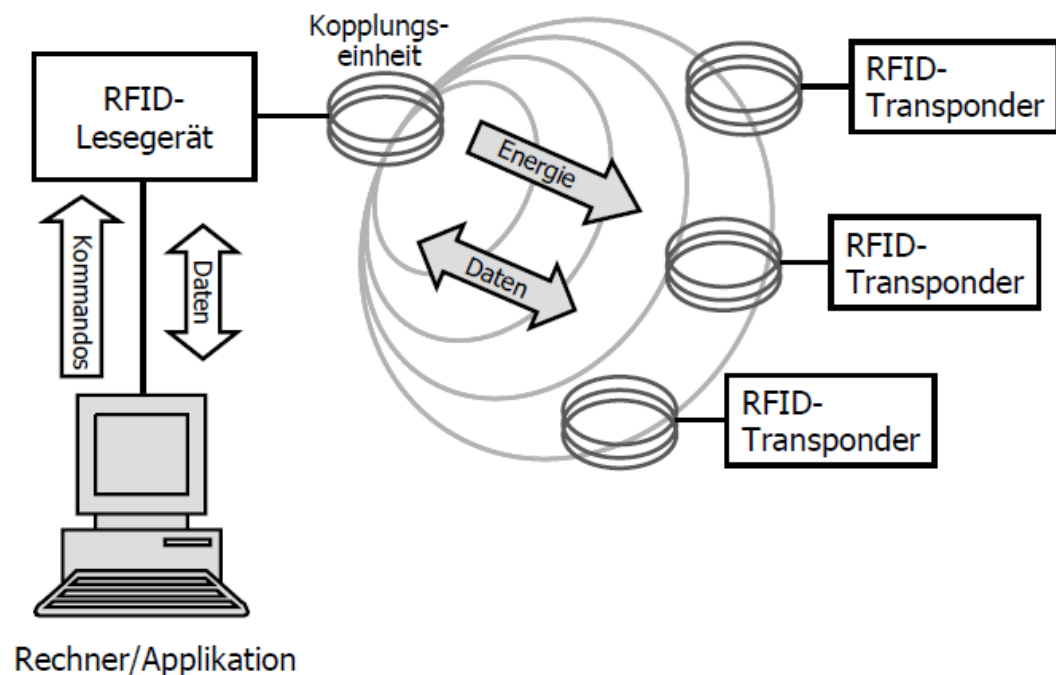


Abbildung 2-4: Komponenten eines RFID-Systems (Lampe et al., 2005, S.71)

Ein Transponder besitzt außerdem einen Mikrochip, welcher als Datenspeicher dient. Die auf dem Chip des Transponders gespeicherten Daten können von dem Lesegerät ausgelesen und je nach Bauart des Tags auch verändert werden. Der Transponder muss sich in dem erzeugten Wechselfeld befinden, um vom Lesegerät angesprochen werden zu können (Lampe et al., 2005).

Transponder unterscheiden sich in ihrer Übertragungsfrequenz und dadurch auch in der Reichweite. Außerdem gibt es Unterschiede in der Energieversorgung. Ein RFID-Tag besitzt entweder eine eigene Energieversorgung, oder er wird durch Induktion, ausgelöst vom Lesegerät, mit Energie versorgt. Ein

weiterer Unterschied zwischen den Transpondern ist das Kopplungselement. Ein RFID-System mit induktiver Kopplung nutzt Spulen, ein RFID-System mit elektromagnetischer Kopplung verwendet Antennen (Lampe et al., 2005).

Vorteile der RFID Technologie

Die RFID-Technologie bietet viele Vorteile gegenüber anderen Identifikationstechnologien, wie beispielsweise dem Barcode. Es ist möglich mehrere RFID-Transponder gleichzeitig auszulesen, was als Vielfachzugriffsverfahren bezeichnet wird. Des Weiteren ist zum Auslesen der Tags kein Sichtkontakt mit dem Lesegerät notwendig. Das ermöglicht, den Transponder beispielsweise in Kunststoff einzubringen, wodurch er vor Schmutz und anderen äußeren Einflüssen geschützt ist. Je nach Bauform können die Tags mit Reichweiten von mehreren Metern ausgelesen werden (Lampe et al., 2005). Durch RFIDs kann außerdem das Kommunikationsaufkommen der Elemente gesenkt werden, da Entscheidungen durch die dezentrale Datenhaltung lokal getroffen werden (Günther et al., 2006). Darüber hinaus ist die Lesegenauigkeit sehr hoch und die Fehleranfälligkeit gering (Franke und Dangelmaier, 2006, S. 72).

Nachteile der RFID-Technologie

Der größte Nachteil der Informationstechnologie ist die fehlende Standardisierung. Durch die Unterschiede in Reichweite, Energieversorgung und Kopplungselement gibt es verschiedene Transponderbauformen, die alle ihr entsprechendes Lesegerät erfordern. Dies ist besonders in Bezug auf die Flexibilität des Gesamtsystems nicht von Vorteil. Ein weiterer Nachteil ist, dass die Verbindung zwischen Lesegerät und Tag durch Metall oder Funk gestört werden kann. Außerdem ist ein schlechter Datenschutz und dadurch eine geringe Sicherheit zu nennen, da die Daten auf dem Transponder durch ein passende Lesegerät, zugänglich sind (Franke und Dangelmaier, 2006, S. 72).

Selbststeuerung mittels RFID

Eine Selbststeuerung stellt folgende Anforderungen an die dezentralen Informationsträger (Voigt et al., 2006):

- Hohe Zuverlässigkeit und Robustheit
- Flexibilität
- Schnelligkeit
- Geringe Kosten

Die RFID-Technologie erfüllt diese vier genannten Anforderungen. RFID-Tags arbeiten laut Finkenzeller (2015, S. 688-689) auch unter schwierigen Umgebungsbedingungen zuverlässig: starker Schmutz, Gegenwart von Flüssigkeiten

oder hohe Temperaturen beeinträchtigen die Funktionsweise nicht. Weiterhin sind RFID-Systeme aufgrund ihrer lokalen Datenverwaltung flexibel. Auch die geforderte Schnelligkeit wird gewährleistet.

Die Erfüllung der vier Anforderungen reicht jedoch nicht aus, um eine Selbststeuerung mit RFID-Technologie umzusetzen. Damit es sich um eine Selbststeuerung handelt sind beschreibbare Tags zu verwenden. Laut Elbert (2006) handelt es sich bei nicht beschreibbaren Tags lediglich um eine erweiterte Fremdsteuerung, da bei dieser Form alle Daten zentral verwaltet werden. Beschreibbare Tags hingegen bieten die Möglichkeit der dezentralen Datenhaltung. Diese ist die Basis für dezentrale Koordination und Entscheidungsfindung, welche für eine Selbststeuerung Voraussetzung ist (Elbert, 2006). Auch Finkenzeller (2015, S. 688-689) nennt die Beschreibbarkeit der Tags als Voraussetzung für eine Selbststeuerung mit RFID-Technologie.

3 Modellierung und Simulation

In diesem Kapitel wird die Modellierung und Simulation von Systemen erläutert. Wie in Kapitel 2 beschrieben, sind produktionslogistische Systeme meist hochkomplex und lassen sich daher nicht vollständig erfassen und steuern. Um diese Komplexität beherrschen zu können, werden durch Vereinfachungen Modelle erstellt, die das reale System abstrahieren und auf die notwendigen Elemente und Verbindungen reduziert darstellen.

Den Prozess der Modellerstellung bezeichnet man als Modellierung. Mit entsprechenden Modellen können schließlich Abläufe innerhalb der Systeme simuliert, analysiert, gestaltet und optimiert werden (Krallmann et al., 2013, S. 52; Eley, 2012, S. 3).

3.1 Modellierung

Das abzubildende reale System wird auf die für die jeweilige Fragestellung relevanten Eigenschaften reduziert, welche im Modell dargestellt werden. Krallmann (2013, S. 59) definiert ein Modell wie folgt: „Ein Modell ist ein System, welches durch eine zweckorientierte, abstrakte Abbildung eines anderen Systems entstanden ist.“ Durch diese Abstraktion reduziert sich die Komplexität des abzubildenden Systems und erleichtert damit dessen Analyse (de Beer, 2008, S.54; Böse, 2012, S.34).

Nach Töllner (2009) ist ein Modell durch drei Hauptmerkmale geprägt, welche nachfolgend erläutert werden (Abbildung 3-1).

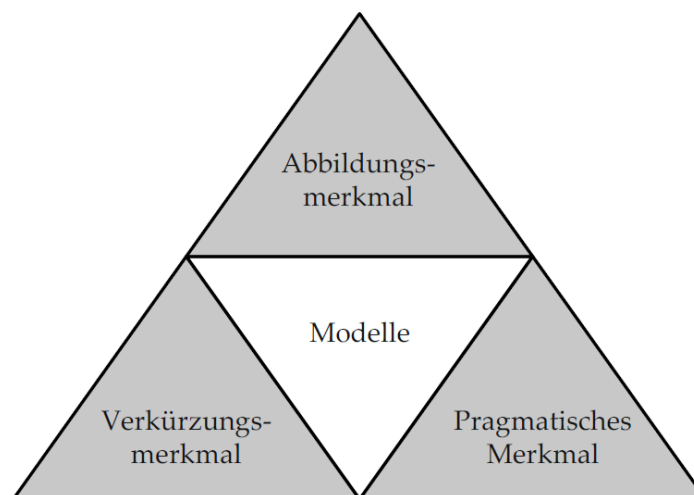


Abbildung 3-1: Hauptmerkmale von Modellen (Töllner et al., 2009, S.9)

In Bezug auf die Komplexitätsreduktion ist das sogenannte Verkürzungsmerkmal von großer Bedeutung. Demnach sollen Modelle nur die relevanten Eigenschaften und Attribute eines realen Systems abbilden.

Das zweite Merkmal ist das Abbildungsmerkmal, welches besagt, dass eine Abbildungs- oder Ähnlichkeitsbeziehung zum realen System, welches selbst auch nur ein Modell sein kann, vorliegt.

Als drittes Merkmal besagt das pragmatische Merkmal, dass ein Modell und sein reales System für einen Zweck und über einen bestimmten Zeitraum miteinander verbunden sind.

Modelle sind vielfältig einsetzbar. Beispiele dafür finden sich in bei Töllner (2009):

- Beschreibung
- Erklärung und Demonstration
- Prognose
- Mathematische Berechnung
- Experimentelle Untersuchung
- Planung und Gestaltung
- Verbesserung

Modelltypen

Produktionslogistische Systeme lassen sich nach de Beer (2008, S. 59-71) je nach Art der Zustandsübergänge, in kontinuierliche, ereignisdiskrete und hybride Modelltypen differenzieren.

Kontinuierliche produktionslogistische Systeme weisen stetige Zustandsveränderungen im Zeitverlauf auf und werden oft benutzt, um dynamische deterministische Zusammenhänge darzustellen.

Ereignisdiskrete Modelle beschreiben diskrete Zustandsübergänge. Dabei ändert sich der Wert der Variablen sprunghaft, sofern ein auslösendes Ereignis auftritt. Die Ereignisse treten zu bestimmten Zeitpunkten ein, welche sich durch stochastische Werte ergeben.

Als dritter Modelltyp kombinieren hybride Modelle die Eigenschaften der beiden zuvor erläuterten Modelltypen. Für das Verhalten der kontinuierlichen Teilmodelle werden allerdings keine stetigen, sondern unstetige Funktionen benutzt. Die kontinuierlichen und ereignisdiskreten Teilmodelle wirken in einem hybriden Modell zusammen und erlauben somit die Untersuchungen von Systemen mit nichtlinearem und komplexem Verhalten.

Vorgehen bei der Modellierung

Bei einer Modellierung entspricht das methodische Vorgehen einem „iterativen, heuristischen und rückgekoppelten Prozess“ (Krallmann et al., 2013, S. 58-59). Aus den Beziehungen zwischen dem realen System (Objekt), dem Modell des Objekts und dem Menschen ergibt sich eine allgemeine Vorgehensweise (Abbildung 3-2).

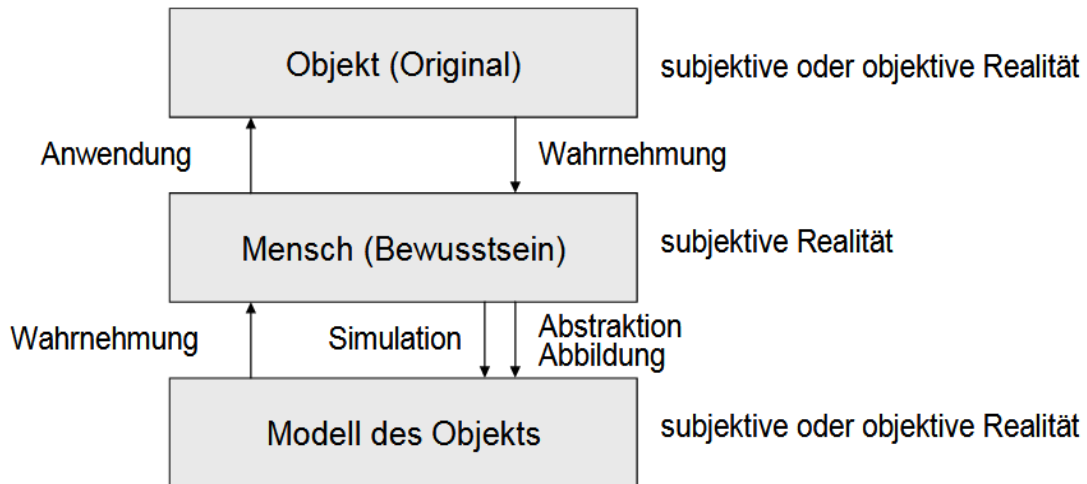


Abbildung 3-2: Beziehungen zwischen Original, Mensch und Modell
(Krallmann et al., 2013, S. 59)

Dabei werden die relevanten Eigenschaften und Ausprägungen des realen Systems durch den Menschen in einem Modell nachgebildet. Bei der Modellierung hat der Mensch sich nach folgenden Zielkriterien, die sich aus dem pragmatischen Merkmal ergeben, zu richten (Krallmann et al., 2013, S. 59):

- Relevanz der abgebildeten Originaleigenschaften (Problemkonformität)
- Korrektheit der Abbildung
- Modelltransparenz

Anschließend ändert der Mensch gezielt Kennzahlen und Ausprägungen im Modell, um das Verhalten des Originalsystems besser zu verstehen und Prognosen über zukünftige, durch Parameteränderungen auftretende Verhaltensänderungen zu geben. Diese Phase wird als Modellexperiment bezeichnet. Aus den gewonnenen Erkenntnissen können Rückschlüsse auf das Original gezogen werden, welche im letzten Schritt auf das reale System übertragen werden (Krallmann et al., 2013, S. 59).

Ziele der Modellierung

In dem Beziehungsdreieck nach Töllner (2009) nutzt eine Person, das Modellsubjekt, ein Modell, um ein Ziel in Bezug auf das reale System zu erreichen. Dadurch ergibt sich ebenfalls ein Zweck des Modells. Um diesen Zweck gezielt zu erfüllen, sollte ein Modell nur für einen oder einige wenige Problemstellungen gestaltet und genutzt werden. Je mehr Zwecke ein Modell erfüllen soll, desto komplexer und anfälliger für Fehler wird es, dabei ist es der Sinn eines Modells, diese Komplexität zu reduzieren. Dieser Widerspruch zeigt, dass der Zweck

eines Modells klar definiert sein muss, um die entsprechende Zielsetzung bestmöglich zu erreichen.

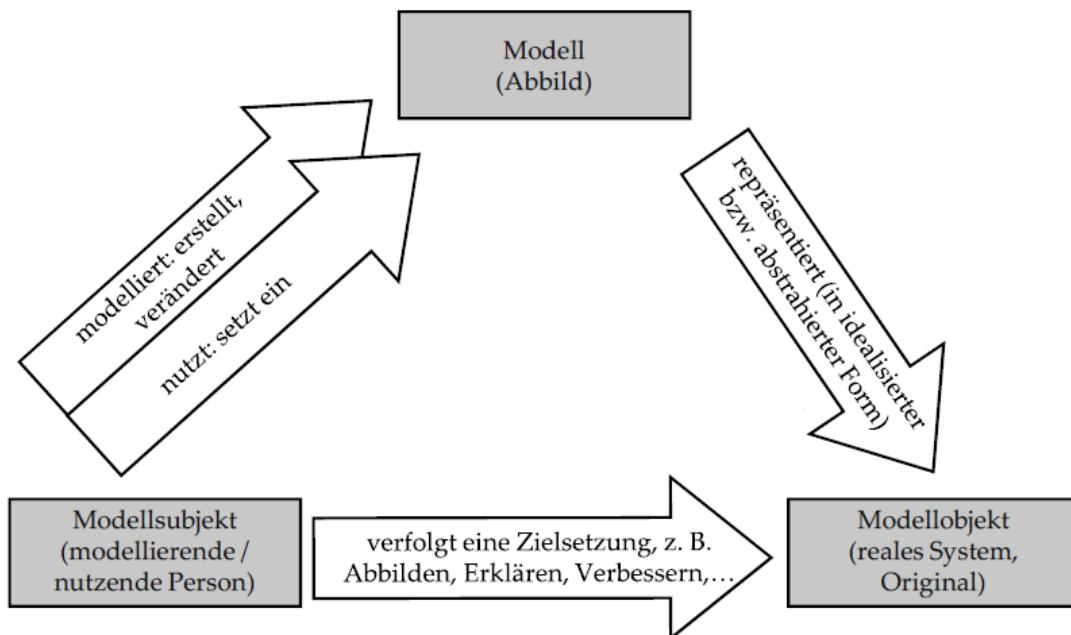


Abbildung 3-3: Beziehungsdreieck (Töllner et al., 2009, S. 10)

Grundsätze ordnungsgemäßer Modellierung

Becker (1995) entwickelte die Grundsätze ordnungsgemäßer Modellierung, welche nachfolgend kurz erläutert werden. Ein Modell muss nach den Grundsätzen syntaktisch und semantisch mit dem realen System übereinstimmen. Außerdem sind jene Elemente abzubilden, durch deren Weglassen der Gesamtnutzen sinken würde. Die Elemente und ihre Beziehungen sind dabei klar strukturiert und übersichtlich darzustellen. Des Weiteren sollen Modelle unabhängig von der Erstellungsmethode vergleichbar sein und es müssen getrennte Sichten der Modellierung integriert werden (Becker et al., 1995).

3.2 Simulation

Simulation ist nach Rose und März (2011) die „Durchführung von Experimenten am Modell“. Durch Simulation lässt sich die Leistung eines Systems bewerten und Schlussfolgerungen ziehen, von denen ausgegangen wird, dass sie auf das reale System übertragbar sind. Das Modell, welches simuliert wird, wird als Simulationsmodell bezeichnet (Eley, 2012, S. 4).

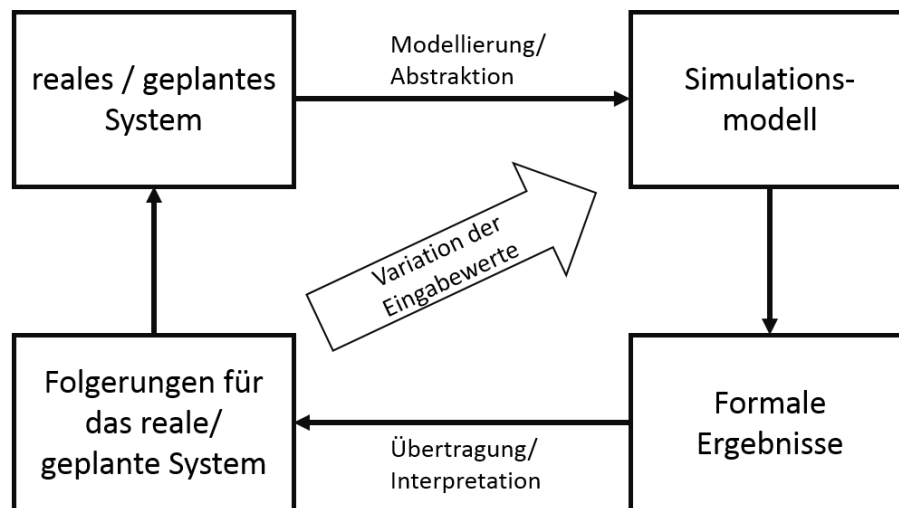


Abbildung 3-4: Vom realen Modell zum Simulationsmodell und wieder zurück nach Eley (2012, S.5)

Zum Aufbau eines Simulationsmodells muss zunächst betrachtet werden, welche der nachfolgend dargestellten Grundeigenschaften für den zu untersuchenden Fall zutreffend sind (Rose und März, 2011):

- Statisch vs. Dynamisch: Ein Modell kann entweder einen Zeitpunkt betrachten (statisch), oder es bildet das zeitliche Verhalten des realen Systems ab (dynamisch).
- Deterministisch vs. Stochastisch: Die Werte und Größen im Modell werden entweder von zufälligen Ereignissen beeinflusst (stochastisch) oder sind vom Zufall unabhängig (deterministisch).
- Kontinuierlich vs. Diskret: Zustände des Systems ändern sich entweder zu bestimmten Zeitpunkten (diskret) oder kontinuierlich.

Ein- und Ausgabewerte

Jede Simulation benötigt Eingabewerte, um Berechnungen durchzuführen und somit verschiedene Szenarien simulieren zu können. Je nach Eingabewerten werden entsprechende Ausgabewerte erzeugt und zurückgegeben.

Eingabewerte können feste, nicht veränderbare Werte (Parameter), stochastische Größen, die durch stochastische Verteilungen beschrieben werden, oder Entscheidungsvariablen, die veränderbar sind, sein.

Über die Ausgabewerte lässt sich die Leistungsfähigkeit eines Systems bestimmen. Ausgabewerte sind entweder Ergebnisgrößen, Zielkriterien oder Kennzahlen (Eley, 2012, S. 4).

Vorteile der Simulation in der Logistik

Simulationen sind vielfältig einsetzbar. Sie kommen sowohl in der Produktions- als auch in der Umschlags- und Transportlogistik zum Einsatz. Vor allem komplexe Systeme mit einem hohen Detaillierungsgrad sind mit Hilfe von Simulation analysier- und auswertbar (Eley, 2012, S. 6; Rose und März, 2011). Neben der guten Eignung zur Annäherung und Abbildung komplexer Systeme bietet Simulation weitere Vorteile (Eley, 2012, S. 6; Banks, 2005, S. 5):

- Jede Überlegung oder Verbesserungsidee kann getestet werden, ohne dabei die laufenden produktiven Systeme oder Prozesse zu unterbrechen oder zu beeinträchtigen.
- In der Simulation ist die Berücksichtigung stochastischer Einflüsse möglich, wodurch sich Extrembedingungen erforschen lassen.
- Auswirkungen von Entscheidungen oder geplanten Änderungen können modelliert und vorhergesagt werden.
- Das Aufdecken, Analysieren und Lösen von Schwachstellen eines Systems ist möglich.

Nachteile der Simulation

Trotz der vielen Vorteile hat Simulation auch Nachteile, die im Folgenden genannt werden (Buchholz, 2016, S.122-123; Banks, 2005, S. 5-6):

- Der Aufwand der Modellerstellung ist sehr hoch und zeitaufwendig, wodurch eine Simulation meist sehr kostenintensiv ist.
- Zur Erstellung eines Modells, sind viele Daten notwendig. Die Erhebung der entsprechenden Daten ist ebenfalls mit einem hohen Aufwand verbunden.
- Die meisten Modelle arbeiten mit stochastischen Größen, wodurch eine statistische Auswertung der Simulationsergebnisse notwendig ist, um zuverlässige Ergebnisse zu erhalten. Hinzu kommt, dass die Interpretation bei Ergebnissen, die von stochastischen Eingangsgrößen abhängen, nicht immer eindeutig ist.
- Um ein Simulationsmodell zu erstellen, ist besonderes Training sowie Erfahrung notwendig.

4 Modellierung und Simulation von Selbststeuerungen in Automod

In diesem Kapitel wird als zunächst das Tool Automod vorgestellt. Darauf aufbauend wird das Konzept für eine Vorgehensweise entwickelt, die es erlaubt, eine auf RFID-Technologie basierende Selbststeuerung in Automod zu modellieren und zu simulieren. Anschließend werden die Besonderheiten der Modellierung in Automod für die nachfolgende Umsetzung des Vorgehens aufgezeigt. Im letzten Abschnitt wird die Umsetzung des Vorgehens anhand der zwei Beispiele erläutert.

4.1 Beschreibung des Tools Automod

Automod ist ein Programm zur ereignisdiskreten Simulation von Materialfluss- und Produktionsprozessen. Entwickelt wurde die Software von der Firma Applied Materials in den USA.

Automod besteht aus drei Umgebungen: Edit Environment, Simulation Environment und Simulation Core. In der Edit Environment werden Modelle erstellt und deren Parameter definiert. In der Umgebung Simulation Environment erfolgt die Ausführung der erstellten Modelle, und es können Simulationsergebnisse, zum Beispiel in Form von Graphen, zur Auswertung des Modells angezeigt werden. Der Simulation Core dient der Berechnung der Modelle und arbeitet im Hintergrund.

In Automod stehen folgende Bausteine für die Modellierung zur Verfügung:

- Entitäten: Entitäten stellen die festen oder beweglichen Elemente des Systems dar. Feste Elemente sind beispielsweise Maschinen oder Arbeitsstationen (Resources) oder Lagerplätze (Queues). Bewegliche Elemente bezeichnen das Gut, welches durch den Materialfluss gesteuert wird (Load)
- Variablen und Attribute: Variablen und Attribute sind Informationsträger, wobei eine Variable für das gesamte Modell gültig ist und ein Attribut nur für das jeweilige Load.
- Prozess: Prozesse bezeichnen ein logisches Konstrukt, in welchem Prozeduren und Anweisungen, denen das Load folgt, beschrieben sind. Die entsprechenden Befehle werden in der Automod Syntax in eine Text-Datei namens *Source File* geschrieben.

Im sogenannten Prozesssystem werden die zur Modellierung eines Systems erforderlichen Bausteine erstellt und parametrisiert. Zum Aufbau eines Systemmodells steht in Automod genau ein Prozesssystem zur Verfügung, welches

mehrere Prozesse beinhalten kann. Bei der Benennung der Bausteine ist es notwendig, den bausteinspezifischen Anfangsbuchstaben vor den eigentlichen Namen zu setzen. Ältere Automod Versionen erkennen sonst nicht um welchen Baustein es sich bei den zugehörigen Anweisungen in der Source File handelt. Über das Prozesssystem hinaus können beliebig viele Movementsysteme erstellt werden. Movementsysteme beschreiben die notwendigen Komponenten verschiedener Transportsysteme, auf welchen das Load transportiert werden kann, beispielsweise ein Förderband oder einen Gabelstapler.

4.2 Vorgehensweise zur Modellierung eines selbststeuernden Systems in Automod

Die Modellierung eines selbststeuernden Systems in Automod erfolgt in fünf aufeinander folgenden Schritten, welche nachfolgend einzeln erklärt werden.

Im ersten Schritt werden die Elemente des realen Prozesses abstrahiert und die zur Modellierung notwendigen Bausteine ausgewählt, angelegt und parametrisiert. Um den Prozess für die spätere Simulation übersichtlich darzustellen, werden die parametrisierten Bausteine dem Materialfluss folgend in der Edit Environment angeordnet.

Im zweiten Schritt werden die Transportsysteme des realen Systems abstrahiert und in Automod als Movementsysteme modelliert. Zugehörige Fahrwege sowie Fahrzeuge werden mit entsprechenden Parametern festgelegt.

Im dritten Schritt folgt die Auswahl an Prozessen, welche im Prozesssystem von Automod angelegt werden müssen. Es empfiehlt sich für jede Prozessstufe im realen System einen separaten Prozess im Simulationsprogramm zu erstellen, um diese logisch zu trennen. Durch die Trennung werden die Anweisungsschritte in der Source File, welche im fünften Schritt erstellt wird, eindeutig voneinander abgegrenzt. Dadurch sind Korrekturen sowie spätere Änderungen des Systems leichter durchführbar.

Nach dem Erstellen aller Prozesse sind im vierten Schritt die notwendigen Loads und zugehörige Attribute zu parametrisieren. Die Attribute stellen einen gutspezifischen RFID-Tag dar und bilden dadurch die spätere Grundlage zur Umsetzung der autonomen Entscheidungen, welche die selbstgesteuerten Maschinen und Fördersysteme anhand der Informationen auf dem RFID-Tag treffen. Anzahl und Art der notwendigen Attribute ergeben sich aus den Informationen, die auf dem RFID-Tag gespeichert werden sollen. Jede Information, die von einer Maschine im realen System abgefragt wird erhält ein eigenes Attribut.

Im fünften Schritt werden die Handlungen der jeweiligen Prozesse durch Anweisungen in der Source File beschrieben. In diesem Schritt wird der Entscheidungsvorgang der Maschinen und Transportsysteme durch bedingte Abfragen und

nachfolgende Handlungsanweisungen in der Source File modelliert. Nach Fertigstellung der Source File ist das so in Automod erstellte Modell bereit zur Simulation.

4.3 Besonderheiten bei der Modellierung von Systemen in Automod

Automod zeichnet sich dadurch aus, dass die Anweisungen in den einzelnen Prozessen immer an das Load gerichtet sind. In den zwei nachfolgend beschriebenen Systembeispielen gibt es jeweils nur einen Load-Typ. Die einzelnen Loads dieses Typs besitzen die gleichen in Automod definierten Attribute und unterscheiden sich nur in deren Ausprägungen. In diesem Fall treten bei der Modellierung keine Probleme auf. Soll jedoch ein System mit mehreren verschiedenen Load-Typen modelliert werden, sind schnell die Grenzen von Automod erreicht. So ist es beispielsweise nicht möglich, zwei verschiedenen Load-Typen unterschiedliche Attribute zu zuweisen, denn jedes Load besitzt automatisch alle in Automod definierten Attribute.

Des Weiteren durchläuft jedes Load bei der Ausführung eines Prozesses alle darin beschriebenen Schritte. Sollen zwei Load-Typen unterschiedlichen Anweisungen folgen, müssen dafür zwei getrennte Prozesse erstellt werden. Wobei ein Prozess dann von Load-Typ eins durchlaufen wird und der andere Prozess von Load-Typ zwei.

Eine weitere Besonderheit der Modellierung mit Automod ist die Darstellung von Arbeitsstationen oder Maschinen, welche immer durch gleichzeitige Verwendung der Bausteine *Queue* und *Resource* erfolgen muss. Das Load kann programmtechnisch nicht in eine Resource bewegt werden, weshalb zu jeder Resource ein Queue zugeordnet werden muss, in welcher sich das Load während der Bearbeitung aufhalten kann. Um die Zusammengehörigkeit der Bausteine in Automod zu verdeutlichen, sind die Namen bis auf den Anfangsbuchstaben gleich zu wählen.

Die Modellierung der Entscheidungsprozesse erfolgt durch bedingte Anweisungen in der Source File. Dabei ist zu beachten, dass Automod nur eine Bedingung abfragen kann. Trifft diese Bedingung zu, wird ein nachfolgender Anweisungsblock ausgeführt. Für den Fall, dass die Bedingung nicht zutrifft kann kein alternativer Anweisungsblock geschrieben werden. Verzweigungen mit mehreren möglichen Bedingungen werden vom Programm nicht erkannt und als Fehler deklariert. Um trotzdem mehr als eine Bedingung prüfen oder eine wenn-dann-mit zugehöriger wenn-nicht-dann-Abfrage modellieren zu können, müssen entsprechend viele bedingte Anweisungen aufeinander folgen und einzeln geprüft werden.

4.4 Systembeispiele

Nachfolgend werden die Prozesse der in diesem Abschnitt entworfenen Beispielsysteme erläutert und anhand der in Kapitel 4.2 entwickelten Vorgehensweise in Automod modelliert und simuliert. Bei der Umsetzung der Systembeispiele werden die Besonderheiten der Modellierung mit Automod berücksichtigt. Zweck der Beispielmodelle ist, zu zeigen, dass das erarbeitete Konzept funktioniert und für verschiedene Systeme anwendbar ist.

Das erste Beispielsystem beschreibt einen selbstgesteuerten, mehrstufigen Produktionsprozess, der an die Automobilfertigung angelehnt ist. Das zweite Systembeispiel stellt einen selbstgesteuerten Kommissioniervorgang dar. Die Selbststeuerung wird sowohl beim ersten als auch beim zweiten Beispiel auf Basis von RFID-Technologie umgesetzt.

Es handelt sich bei beiden Modellen um dynamische und stochastische Simulationen, da das Zeitverhalten der Systeme abgebildet und von zufälligen Größen beeinflusst wird. Die Zustände der Systeme ändern sich nicht kontinuierlich, sondern nur zu bestimmten Zeitpunkten, deshalb handelt es sich zudem um ein diskretes Simulationsmodell.

4.4.1 Systembeispiel 1: Automobilfertigung

Prozessbeschreibung

Die in diesem System gefertigten Autos unterscheiden sich in zwei Merkmalen: Farbe (Blau oder Rot) und Karosserie (Limousine oder Cabrio). Dadurch ergeben sich vier verschiedene Varianten (Abbildung 4-1).

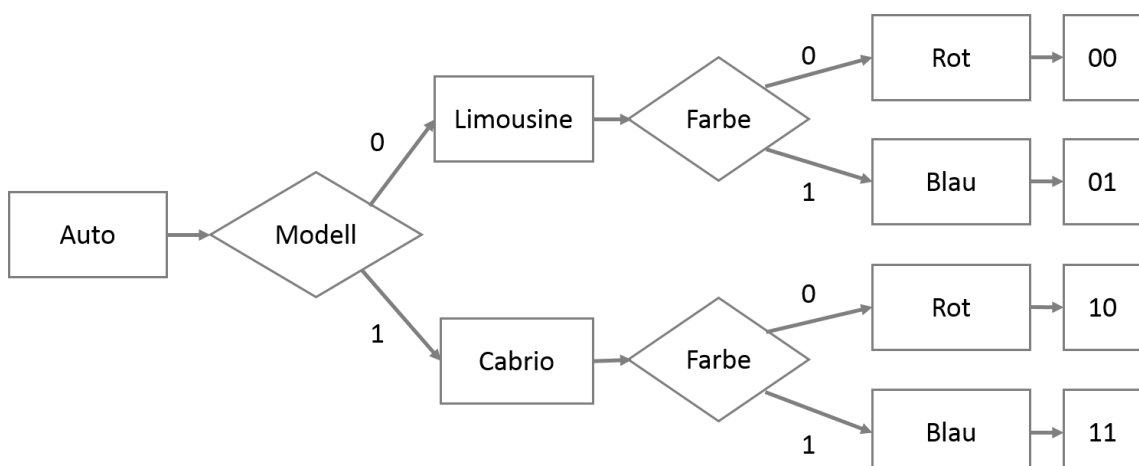


Abbildung 4-1: Fahrzeugvarianten

Welche Varianten produziert werden hängt von den entsprechenden Kundenaufträgen ab. Die vom Kunden geforderten Ausprägungen der Merkmale werden

auf einem RFID-Tag, der am Auto befestigt ist, gespeichert. Jeder Informationsträger speichert außerdem den aktuellen Bearbeitungsstatus des Fahrzeugs. Zu Beginn der Fertigungssequenz wird der Status „Bereit zur Lackierung“ gespeichert.

Im ersten Schritt der Fertigung wird das Auto lackiert. Um eine Lackierung gemäß der Kundenwünsche sicherzustellen, wird der RFID-Tag des jeweiligen Fahrzeugs gescannt und die angeforderte Farbe ausgelesen. Mit Beginn der Lackierung setzt die Maschine den Status des Informationsträgers auf „In Lackierung“ und je nach Farbe folgt anschließend eine entweder 10 oder 11 Sekunden dauernde Lackierung. Durch den Scann-Prozess findet eine Kommunikation zwischen Maschine und RFID-Tag statt, woraufhin die Maschine selbstständig entscheidet, in welcher Farbe lackiert wird. Außerdem ändert die Maschine das Attribut und somit die dezentral gespeicherten Daten. Nach der Lackierung wird das Attribut „Status“ auf „Bereit zur Montage“ gesetzt.

Im nächsten Schritt erfolgt die karosserieformunabhängige Montage des Autos. Zu Beginn der Montage wird der Status des jeweiligen Fahrzeugs geprüft, lautet der Status „Bereit zur Montage“, kann das Auto montiert werden. Der Prozess besteht aus 20 einzelnen Schritten, welche getaktet sind. Durch die Taktung ist die Bearbeitungszeit für jede Variation in jedem Schritt gleich. Der gesamte Montageprozess dauert für beide Karosserieformen jeweils 600 Sekunden. Am Ende der Montage wird das Attribut „Status“ auf „montiert“ gesetzt.

Das Transportsystem scannt den RFID-Tag und prüft bei der fertigen Karosserie, ob es sich um eine Limousine oder ein Cabrio handelt und fördert diese abhängig davon zum nächsten Prozess.

Die Limousinen werden im nächsten Prozessschritt von einer Kontrollmaschine für 25 Sekunden auf Qualität und Fehler geprüft. Die Cabrios erhalten in deren nächsten Prozessschritt ein Cabrio-Verdeck, welches an einer entsprechenden Montagestation in einem 20 Sekunden Takt montiert wird. Die automatische Qualitätskontrolle eines Cabrios durch eine Maschine schließt sich direkt an die Verdeck-Montage an und dauert 25 Sekunden.

Nach der Qualitätsprüfung von Limousine oder Cabrio ändert die jeweilige Maschine den Status in „Verkaufsbereit“, und die Autos werden mit dem fahrerlosen Transportsystem zum Warenausgang gefahren. Das Transportsystem transportiert nur Autos mit dem Attribut „Verkaufsbereit“ zum Warenausgang.

Das Auto wird durch das Transportsystem dezentral gesteuert und durch den Fertigungsprozess geführt. Anhand der Daten auf dem RFID-Tag treffen Maschinen und Transportsysteme autonom Entscheidungen, wie mit dem jeweiligen Auto zu verfahren ist. Veranschaulicht wird der zuvor beschriebene Prozess in Abbildung 4-2.

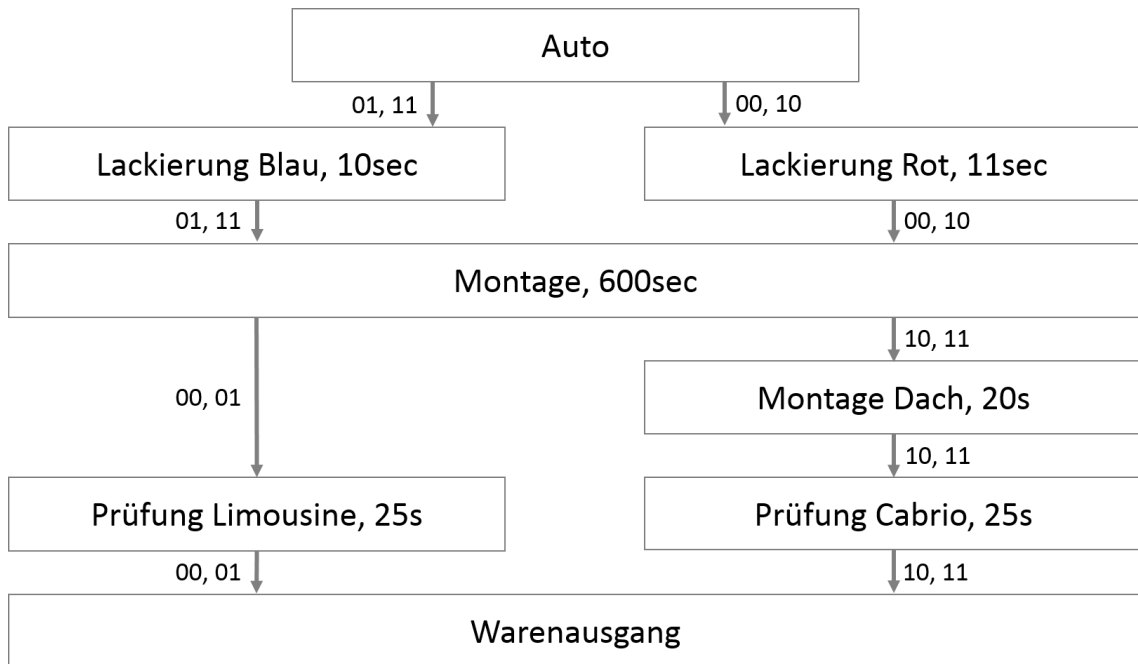


Abbildung 4-2: Produktionsprozess Automobilfertigung

Modellierung und Simulation

Wareneingang, Warenausgang, Zwischenlager zwischen Lackierung und Montage sowie ein weiteres Lager zwischen Montage und Fördersystem werden als Queue dargestellt. Ein Queue ist ein Baustein, der eine gewisse Anzahl an Loads aufnehmen kann. Die maximale Kapazität an Loads kann bei der Erstellung des Bausteins im Feld „Kapazität“ definiert werden. Nach Festlegen von Name und Kapazität wird der Baustein in der Edit Environment platziert.

Lackierung, Montage, Montage des Cabrio Dachs sowie Prüfstationen werden als eine Kombination aus den Bausteinen Queue und Resource dargestellt. Eine Resource hat, wie ein Queue auch, eine Kapazität, die bei der Erstellung des Bausteins festgelegt wird. Der zugehörige Queue muss die gleiche Kapazität aufweisen. Um die Zusammengehörigkeit von Queue und Resource darzustellen, müssen die Namen der Bausteine bis auf den Baustein spezifischen Anfangsbuchstaben gleich sein. In der nachfolgenden Tabelle sind die für das Modell notwendigen Queues und Resources dargestellt.

Tabelle 1: Queues und Resources

Bezeichnung	Aufgabe
Q_WE	Eingangspuffer
Q_WA	Warenausgangspuffer
Q_LackMontage	Puffer zwischen Lackierung und Montage

Bezeichnung	Aufgabe
Q_FTS	Puffer zwischen Montage und Transportsystem
Q_Lack	Lackierung
R_Lack	
Q_Montage	Montage
R_Montage	
Q_CabrioDach	Montage Cabriodach
R_CabrioDach	
Q_CabrioPruef	Prüfstation Cabrio
R_CabrioPruef	
Q_LimousinePruef	Prüfstation Limousine
R_LimousinePruef	

Anschließend wurde das fahrerlose Transportsystem angelegt, welches in Automod als separates Movementsystem zu modellieren ist. Bei einem fahrerlosen Transportsystem handelt es sich um ein weggebundenes, unstetiges Transportmittel, welches in Automod als Movementsystem Path Mover anzulegen ist. Als erstes sind bei der Modellierung eines Path Movers die Wege festzulegen, auf denen sich die Fahrzeuge später bewegen dürfen. Ein solcher Weg ist programmbedingt nur in eine Richtung befahrbar und muss als Kreislauf angeordnet werden, um den Fahrzeugen einen Rückweg zu ermöglichen.

Als nächstes werden auf den Weg sogenannte *Control Points* gesetzt und nah an den verarbeitenden Maschinen und Puffern platziert. An diesen Punkten kann ein Fahrzeug später Loads aufnehmen oder abgeben und auch parken. Nach dem Anlegen aller Control Points erfolgt das Definieren des Verhaltens der Fahrzeuge, indem man zu jedem Control Point eine Liste erstellt, die beschreibt, zu welchem Control Point das Fahrzeug nach Erledigen der jeweiligen Aufgabe als nächstes fährt. Das Festlegen von Fahrzeuganzahl und entsprechenden Attributen, wie beispielsweise Geschwindigkeit stellt den letzten Schritt bei der Erstellung des Movementsystems dar. In diesem Beispiel wurden fünf Fahrzeuge mit einer Geschwindigkeit von 1 m/s eingesetzt. Angesteuert wird das Movementsystem über die Anweisungen in der Source File.

Nach dem Definieren und Parametrisieren aller Bausteine sowie des Transport-

systems folgt das Erstellen der einzelnen Prozesse im Prozesssystem. Das vorliegende Beispiel verwendet vier Prozesse, welche in Tabelle 2 zusammengestellt sind.

Tabelle 2: Prozesse

Bezeichnung	Prozessbeschreibung
P_Variante	Zuweisung der Ausprägungen entsprechend des Kundenauftrags
P_Lack	Lackierung der Autos
P_Montage	Montage der Autos
P_Prüfung	Prüfung der Autos

Vor Definition der Prozessschritte in der Source File, muss das Load mit den vom Kunden gewünschten Attributen versehen werden, was ebenfalls im Prozesssystem erfolgt. Bei der Erstellung der Loads wird festgelegt, in welchem zeitlichen Abstand und in welchem der vier Prozesse ein neues Load in das Modell eintritt. In diesem Beispiel tritt ein Load alle 30 Sekunden in den Prozess P_Variante ein. Bei den zugehörigen Attributen wird der entsprechende Datentyp ausgewählt. Tabelle 3 zeigt die notwendigen Loads mit entsprechenden Attributen.

Tabelle 3: Loads und Attribute

Bezeichnung	Bedeutung	Datentyp
L_Auto	Beschreibt das Gut	-
A_Farbe	Speichert die Ausprägung des Attributs Farbe	Integer
A_Dach	Speichert die Ausprägung des Attributs Dach	Integer
A_Status	Speichert den Bearbeitungsstatus des Guts	String

Zusätzlich zu den beschriebenen Bausteinen wird eine Variable des Typs Integer namens *V_Count* definiert, die dazu dient, den in Tabelle 3 genannten Attributen zufällige Werte zuzuweisen. Diese stochastische Zuweisung der Attribute beschreibt die nicht vorhersehbaren Kundenwünsche und wird nachfolgend beschrieben. Nachdem das Load in den ersten Prozess eingetreten ist, wird der Variable *V_Count* über eine Gleichverteilung mit Mittelwert fünf und Abweichung

drei eine zufällige Zahl zugeordnet. Ist diese Zahl kleiner fünf, setzt Automod das Attribut A_Farbe auf „1“, wobei „1“ für die Farbe Blau steht. Eine Zahl größer oder gleich fünf ordnet dem Attribut den Wert „0“ zu, für die Farbe Rot.

Im nächsten Schritt wird der Variable V_Count nochmals über die gleiche Zufallsverteilung eine Zahl zugewiesen. Diesmal wird das Attribut A_Dach gesetzt. Ein Wert kleiner fünf ordnet dem Attribut die Zahl 1 zu, die für das Cabrio steht, ein Wert größer oder gleich fünf den Wert 0, welcher die Limousine darstellt.

Im letzten Schritt der Modellerstellung erfolgt die Beschreibung der Anweisungen der einzelnen Prozesse in Automod Syntax in der sogenannten *Source File*. Die dort abgefassten Anweisungen richten sich an das Load. Bei der Source File handelt es sich um eine Textdatei, die zum Modell gehört. Um die Auslesevorgänge der RFID-Tags zu modellieren, wurden in der Source File bedingte Abfragen verwendet. In den Abfragen werden die jeweiligen Attribute abgefragt und, wenn ein Attribut wahr ist, die entsprechend beschriebenen Anweisungen vom Load durchgeführt. Die Source File zu diesem Modell ist im Anhang zu finden.

4.4.2 Systembeispiel 2: Kommissionierung

Prozessbeschreibung

Das zweite fiktive System beschreibt einen Kommissioniervorgang, bei welchem drei verschiedene Teile in eine Box kommissioniert werden. Ein Kommissionierauftrag kann aus mindestens einer und maximal drei Positionen bestehen. Jedes Teil wird dabei maximal einmal bestellt. Es ergeben sich sieben verschiedene Möglichkeiten Teile zu Aufträgen zu kommissionieren, welche in Tabelle 4 dargestellt werden.

Tabelle 4: Auftragsvariation

Art	Teil 1	Teil 2	Teil 3
1	1	0	0
2	0	1	0
3	0	0	1
4	1	1	0
5	1	0	1
6	0	1	1
7	1	1	1

Die Lagerung der zu kommissionierenden Teile erfolgt artikelrein in Regalen. Jede Kommissionierbox ist mit einem RFID-Tag versehen, auf dem zu Beginn in der Kommissioniervorbereitung die im Kommissionierauftrag vorhandenen Teile gespeichert werden. Des Weiteren speichert der Informationsträger den aktuellen Auftragsstatus. In der Vorbereitung wird dieser auf „leer“ gesetzt.

Zur Kommissionierung fährt die Box über ein Förderband zu dem jeweiligen Regal. Das Förderband ist so angeordnet, dass die Box an allen drei Regalen nacheinander vorbeifährt. Am ersten Regal wird der RFID-Tag an der Box vom Fördersystem gescannt und wenn der Auftrag Artikel 1 umfasst, wird die Box vom Fördersystem auf eine zum Regal gehörige Abstellfläche geschoben. Ein Mitarbeiter legt das Teil in die Box und quittiert den Vorgang, welcher durchschnittlich 10 Sekunden dauert, indem er den RFID-Tag scannt. Der Status wird durch die Quittierung auf „Artikel 1 eingeladen“ gesetzt. Anschließend setzt der Mitarbeiter die Box wieder auf den Förderer. Das Förderband transportiert sie zum nächsten Regal, wo sich der Vorgang wiederholt. Nach dem dritten Regal wird die Box zum Warenausgang gefördert.

Dort entnimmt ein Mitarbeiter die in der Box zusammengestellten Teile und quittiert dieses durch Scannen des RFID-Tags. Dadurch werden die Auftragsdaten gelöscht, die Box für den nächsten Kommissionierauftrag freigegeben und über das Förderband zurück in die Vorbereitung gefahren. Der Entnahmeprozess durch den Mitarbeiter dauert durchschnittlich 10 Sekunden.

Modellierung und Simulation

Zur Darstellung der Artikel-Regale mit jeweiliger Abstellfläche für eine Kommissionier-Box wurden Queues genutzt. Die Verwendung einer Ressource war nicht notwendig, da die Beladevorgänge durch den Mitarbeiter für die Modellierung der Selbststeuerung nicht relevant sind. Die Zeit zur Beladung der Boxen wird als Normalverteilung mit einem Mittelwert von 10 Sekunden und einer Abweichung von 2 Sekunden angesetzt, da nicht jeder Mitarbeiter gleich schnell arbeitet.

Vor der Kommissionierung werden die Boxen vorbereitet. Die Modellierung der Kommissioniervorbereitung sowie des Warenausgangs erfolgte als Queue. Die Entnahmezeit der Teile wird durch eine Normalverteilung mit Mittelwert 10 Sekunden und einer Abweichung von 3 Sekunden dargestellt. Die Abweichung ist bewusst größer gewählt als die des Beladevorgangs an den Regalen, da die Entnahmemenge variiert. Die Box selbst wurde durch ein Load abgebildet, da es sich um die bewegliche Einheit in diesem Materialfluss handelt. Die zu speichernden Informationen ließen sich durch Load-Attribute umsetzen. Für die drei verschiedenen möglichen Teile wurde jeweils ein Attribut des Typs Integer gewählt. Das Attribut kann die Werte 1 für „Teil im Auftrag enthalten“ oder 0 „Teil

nicht enthalten“ annehmen. Es wurde der Typ Integer gewählt, da Boolean-Attribute in Automod nicht vorhanden sind. Als viertes Attribut wurde ein Attribut des Typs String für den Bearbeitungsstatus ausgewählt. Damit das zufällige Eintreffen der verschiedenen Aufträge modelliert werden konnte, wurde eine Variable namens V_Count eingeführt. Die Zuweisung der Werte zu den Attributen ist nachfolgend erklärt: Zuerst wird der Variable V_Count ein zufälliger Wert über eine Gleichverteilung mit Mittelwert fünf und einer Abweichung von drei zugewiesen. Bei einem Wert kleiner fünf speichert die Maschine die Zahl „0“ für das Attribut A_1, bei einem Wert größer oder gleich fünf die Zahl „1“. Dieser Vorgang wiederholt sich für alle drei Attribute. Die Zuweisung der Variablen findet in dem Prozess P_Vorbereitung statt, die Kommissionierung im Prozess P_Kommissionierung. Alle verwendeten Bausteine können Tabelle 5 entnommen werden:

Tabelle 5: Notwendige Bausteine

Bausteinbezeichnung	Bedeutung
Q_Vorbereitung	Puffer, an welchem Auftragsdaten auf dem Tag gespeichert werden
Q_Teil1	Regal von Teil 1
Q_Teil2	Regal von Teil 2
Q_Teil3	Regal von Teil 3
Q_WA	Warenausgang
L_Box	Transportbox, bewegliche Entität
A_Teil1	Attribut welches speichert ob Teil 1 kommissioniert werden muss
A_Teil2	Attribut welches speichert ob Teil 2 kommissioniert werden muss
A_Teil3	Attribut welches speichert ob Teil 3 kommissioniert werden muss
A_Status	Bearbeitungsstatus des Auftrags
V_Count	Hilfsvariable um stochastische Einflüsse darzustellen
P_Vorbereitung	Zuweisung der Attributwerte
P_Kommissionierung	Anweisungen der Kommissionierung

Der Transport der Boxen erfolgt durch ein Förderband, welches ein weggebundenes, stetiges Transportmittel darstellt und in Automod als Movementsystem Conveyor modelliert wurde. Ähnlich wie bei einem Path Mover erfolgt auch bei einem Conveyor zuerst die Definition des Weges. Dazu wurden die Förderbandabschnitte in der Edit Environment platziert und anschließend verbunden. Danach wurden sogenannte *Stations* festgelegt, an denen ein Load auf das Förderband geladen oder vom Förderband genommen werden kann. Dieses Beispiel benötigt neun Stations, jeweils einen Abgabe- und Aufgabepunkt an den Regalen und am Warenausgang sowie eine Station an der Vorbereitung.

Die Einbindung des Movementsystems in den Prozess erfolgte in der Source File. Die Gestaltung der Auslesevorgänge der RFID-Tags ließ sich über bedingte Abfragen in der Source File gestalten. Die Source File mit den entsprechenden Abweisungen ist im Anhang zu finden.

5 Zusammenfassung und Ausblick

Logistische Systeme werden aufgrund der fortschreitenden Globalisierung und steigender Marktdynamik zunehmend komplexer und sind dadurch mit wachsenden Ansprüchen an Flexibilität, Robustheit und Anpassungsfähigkeit konfrontiert. Der Wechsel von einer zentralen Steuerung zu einer Selbststeuerung bietet die Möglichkeit diesen Anforderungen gerecht zu werden.

Die vorliegende Projektarbeit zielt daher auf die Entwicklung und Umsetzung eines Konzepts zur Modellierung und Simulation selbstgesteuerter produktionslogistischer Systeme in dem Materialflussprogramm Automod ab.

Durch ausführliche Literaturrecherchen ließen sich Systeme abgrenzen, auf welche das entwickelte Konzept anwendbar ist. Im Einzelnen wurden Definition und Abgrenzung des Begriffs „produktionslogistisches System“ sowie Erläuterung, Voraussetzungen und Möglichkeiten von Selbststeuerungen aufgezeigt. Weiterhin wurde die RFID-Technologie als Möglichkeit zur Umsetzung logistischer Selbststeuerungen als besonders geeignet identifiziert und dargelegt. Für eine Basis der späteren Konzeptentwicklung lieferten weitere Recherchen Merkmale von Modellierung und Simulation sowie Anforderungen und Besonderheiten bei deren Einsatz. Als weitere Grundlage wurde der Aufbau des Simulationstools Automod, sowie darin verwendbare Bausteine beschrieben.

Auf Basis der Software Automod wurde eine Vorgehensweise zu Modellierung und Simulation selbststeuernder produktionslogistischer Systeme entwickelt. Das Vorgehen bezieht sich auf eine RFID-basierte Selbststeuerung. Zu diesem Zweck erfolgte die Abbildung von RFID-Tags in Automod durch loadspezifische Attribute, denen zu Beginn der Simulation zufällige Werte zugewiesen wurden. Die Entscheidungsvorgänge ließen sich über bedingte Abfragen der Attributwerte in der Source File modellieren, welche entsprechend Handlungsanweisungen auslösten.

Im Zuge einer beispielhaften Realisierung des Konzeptes wurden zwei fiktive Systeme entworfen. Das erste System beschreibt einen mehrstufigen Prozess aus der Automobilfertigung, welcher eine selbstgesteuerte Maschine sowie ein autonom agierendes fahrerloses Transportsystem enthält. Das zweite System stellt einen Kommissioniervorgang mit einem selbstgesteuerten Förderband dar. Für eine dezentrale Datenhaltung und Entscheidungsfindung erfolgte die Ausstattung der Produkte in beiden Anwendungsbeispielen mit einem RFID-Tag. Die gespeicherten Daten wurden von den selbstgesteuerten Maschinen und Transportsystemen ausgelesen, um autonome Entscheidungen zu ermöglichen.

Im Zuge der Umsetzung des Konzeptes stellte sich heraus, dass beide Systeme sowie deren Selbststeuerung gut anhand der in Automod verfügbaren Bausteine

und Transportsysteme modelliert und simuliert werden konnten. Folglich ist festzuhalten, dass Automod zur Modellierung und Simulation einfacher selbststeuernder produktionslogistischer Systeme gut geeignet ist.

Aus den Ergebnissen dieser Arbeit leiten sich zwei weiterführende Fragestellungen ab. Zum einen sollte vertiefend geprüft werden, inwiefern Automod zur Untersuchung komplexerer Systeme, zum Beispiel Systemen mit unterschiedlichen Load-Typen, geeignet ist. Zum anderen stellt sich die Frage, wie sich andere Simulationsprogramme, beispielsweise AnyLogic oder Plant Simulation, in Bezug auf die Modellierung und Simulation von Selbststeuerungen von Automod unterscheiden. Darüber hinaus sind deren spezifische Vor- und Nachteile für die Anwendung in der industriellen Praxis von Interesse.

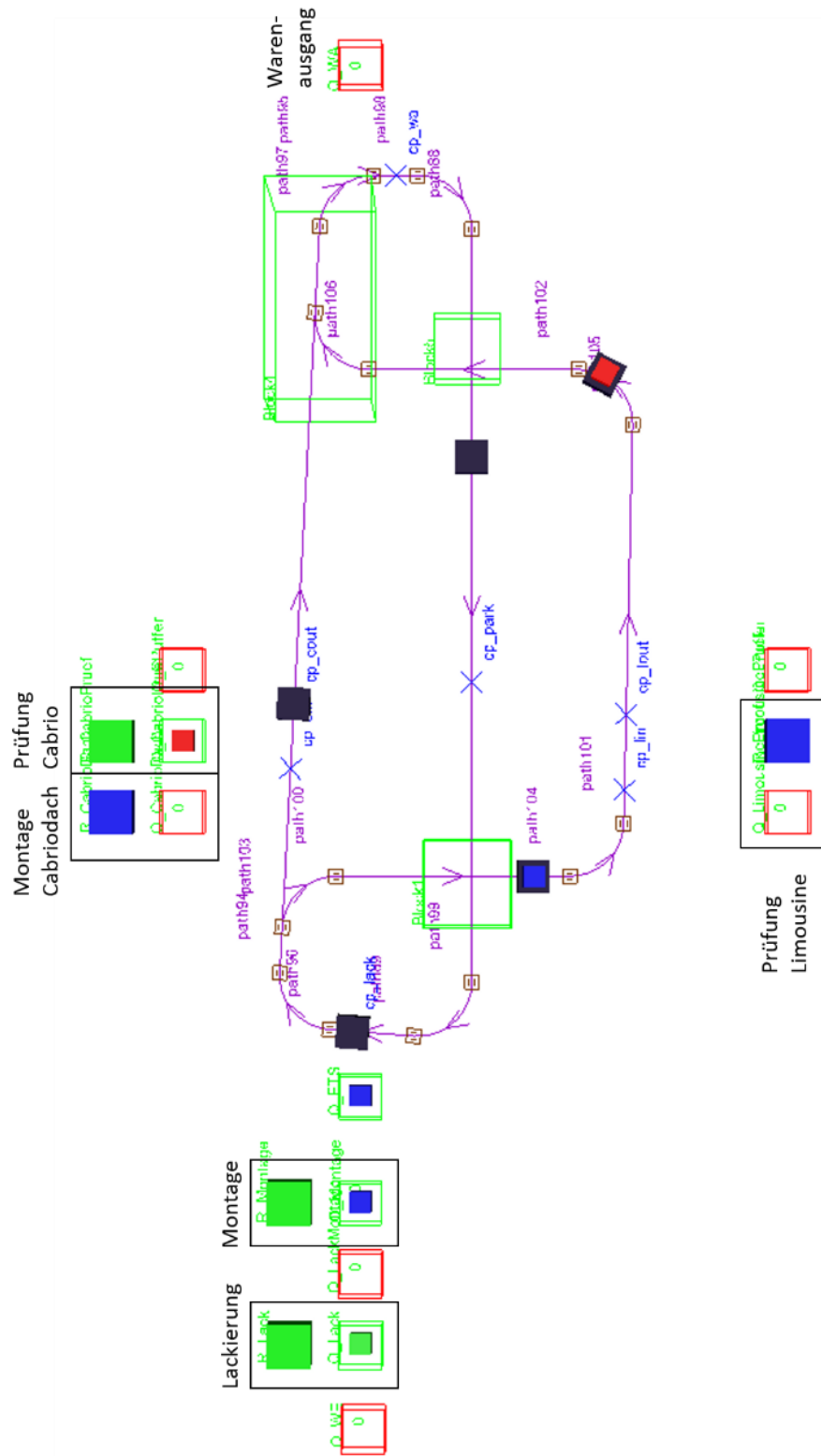
Literaturverzeichnis

- Arnold, D.; Isermann, H.; Kuhn, A.; Tempelmeier, H.; Furmans, K.: Handbuch Logistik, 3. Edition. Berlin: Springer 2008.
- Banks, J.: Discrete-event system simulation, 4. Edition. Upper Saddle River, NJ: Pearson Prentice Hall 2005.
- Becker, J.; Rosemann, M.; Schütte, R.: Grundsätze ordnungsgemäßer Modellierung. *Wirtschaftsinformatik* (1995) 37, S. 435–445.
- Bloech, J.; Bogaschewsky, R.; Buscher, U.; Daub, A.; Götze, U.; Roland, F.: Einführung in die Produktion, 7. Edition. Berlin: Gabler 2014.
- Böse, F.: Selbststeuerung in der Fahrzeuglogistik: Modellierung und Analyse selbststeuernder logistischer Prozesse in der Auftragsabwicklung von Automobilterminals. Berlin: GITO-Verl. 2012.
- Buchholz, P., 2016: Modellgestützte Analyse und Optimierung. Dortmund. Online verfügbar unter <http://ls4-www.cs.tu-dortmund.de/download/buchholz-/MAO/skript.pdf>, zuletzt geprüft am 27.07.2017.
- de Beer, C.: Untersuchung der Dynamik von selbststeuernden Prozessen in produktionslogistischen Systemen anhand ereignisdiskreter Simulationsmodelle. Berlin: Gito 2008.
- Elbert, R.: Möglichkeiten und Grenzen der Selbststeuerung in der Logistik. In: Pfohl, H.-C. (Hrsg.): *Wissenschaft und Praxis im Dialog: Theorie geleitete Analyse der Voraussetzungen für Integration und Koordination zur Selbststeuerung*. Hamburg: Dt. Verkehrs-Verlag 2006, S. 55–74.
- Eley, M.: Simulation in der Logistik: Einführung in die Erstellung ereignisdiskreter Modelle unter Verwendung des Werkzeuges "Plant Simulation". Berlin, Heidelberg: Springer 2012.
- Finkenzeller, K.: RFID-Handbuch: Grundlagen und praktische Anwendungen von Transpondern, kontaktlosen Chipkarten und NFC, 7. Edition. München: Hanser 2015.
- Franke, W.; Dangelmaier, W.: RFID-Leitfaden für die Logistik: Anwendungsgebiete, Einsatzmöglichkeiten, Integration, Praxisbeispiele. Wiesbaden: Gabler 2006.
- Freitag, M.; Herzog, O.; Scholz-Reiter, B.: Selbststeuerung logistischer Prozesse. *Industriemanagement* (2004) 20, S. 23–27.
- Günther, W.A.; Kuzmany, F.; Chisu, R.: Selbststeuernde Materialflussmittel. In: Pfohl, H.-C. (Hrsg.): *Wissenschaft und Praxis im Dialog: Ein*

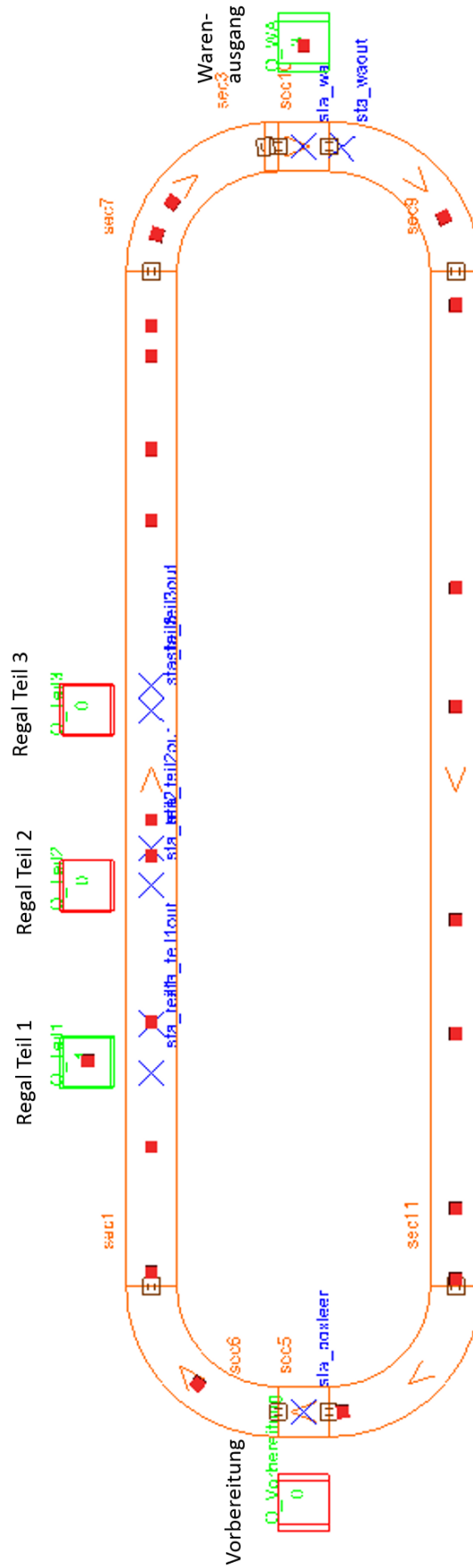
- Technologiebaukasten für das Internet der Dinge. Hamburg: Dt. Verkehrs-Verlag 2006, S. 307–319.
- Kolditz, J.: Fachkonzeption für selbststeuernde logistische Prozesse. Berlin: Gito 2009.
- Krallmann, H.; Bobrik, A.; Levina, O.: Systemanalyse im Unternehmen: Prozessorientierte Methoden der Wirtschaftsinformatik, 6. Edition. München: Oldenbourg-Verlag 2013.
- Lampe, M.; Flörkemeier, C.; Haller, S.: Einführung in die RFID-Technologie. In: Fleisch, E. (Hrsg.): Das Internet der Dinge. Berlin: Springer 2005, S. 69–86.
- Pawellek, G.: Produktionslogistik. In: Klaus, P.; Krieger, W.; Krupp, M. (Hrsg.): Gabler Lexikon Logistik. Wiesbaden: Springer Gabler 2012, S. 466–471.
- Rose, O.; März, L.: Simulation. In: März, L.; Krug, W.; Rose, O.; Weigert, G. (Hrsg.): Simulation und Optimierung in Produktion und Logistic. Heidelberg: Springer 2011, S. 13–22.
- Scholz-Reiter, B.; Freitag, M.; Rekersbrink, H.; Wenning B.L.; Gorltdt, C.; Echelmeyer, W.: Auf dem Weg zur Selbststeuerung in der Logistik. In: Wäschner, G. (Hrsg.): Begleitband zur 11. Magdeburger Logistiktagung: Grundlagenforschung und Praxisbeispiele. Magdeburg: Logisch-Verlag 2005, S. 166–180.
- Scholz-Reiter, B.; Philipp, T.; de Beer, C.; Windt, K.; Freitag, M.: Einfluss der strukturellen Komplexität auf den Einsatz von selbststeuernden logistischen Prozessen. In: Pfohl, H.-C. (Hrsg.): Wissenschaft und Praxis im Dialog. Hamburg: Dt. Verkehrs-Verlag 2006, S. 11-25.
- Schuh, G.: Produktionsplanung und -steuerung: Grundlagen, Gestaltung Und Konzepte, 3. Edition. Dordrecht: Springer 2007.
- Tentrop, F.: Entwicklung eines integrierten Gestaltungsansatzes der Produktionslogistik. Berlin: Universitätsverlag TU Berlin 2011.
- Töllner, A.; Jungmann, T.; Bücker, M.; Brutscheck, T.: Modelle und Modellierung. In: Bandow, G.; Holzmüller, H.H. (Hrsg.): „Das ist gar kein Modell!“. Terminologie, Funktionen und Nutzung. Wiesbaden: Gabler 2009, S. 3–12.
- Voigt, K.-I.; Saatmann, M.; Schorr, S.: Einsatzfelder und Grenzen der Selbststeuerung in flexiblen Logistiksystemen. In: Pfohl, H.-C. (Hrsg.): Wissenschaft und Praxis im Dialog. Hamburg: Dt. Verkehrs-Verlag 2006, S. 96–115.
- Wannenwetsch, H.: Integrierte Materialwirtschaft, Logistik und Beschaffung, 5. Edition. Berlin: Springer Vieweg 2014.

Anhang

Anhang A: Modell Systembeispiel 1



Anhang B: Modell Systembeispiel 1



Anhang C: Source File Systembeispiel 1

```
begin P_Variante arriving procedure

    set V_Count = uniform 5,3 //Verteilung um Attribute zufällig zu
wählen

    if V_Count < 5 then begin //Attribut Farbe wird auf blau gesetzt
        set A_Farbe = 1
    end

    if V_Count >= 5 then begin //Attribut Farbe wird auf rot gesetzt
        set A_Farbe = 0
    end

    set V_Count = uniform 5,3 //Verteilung um Attribute zufällig zu
wählen

    if V_Count < 5 then begin //Attribut Dach wird auf Cabrio gesetzt
        set A_Dach = 1
    end

    if V_Count >= 5 then begin //Attribut Dach wird auf Limousine
gesetzt
        set A_Dach = 0
    end

    set A_Status = "Bereit zur Lackierung"

    move into Q_WE

send to P_Lack
end

begin P_Lack arriving procedure

    move into Q_Lack

    if A_Farbe = 1 then //Selbststeuerung Part 1 Unterscheidung nach
Farbe
    begin
        set A_Status = "In Lackierung"
        use R_Lack for 10
        set color = blue
        set A_Status = "Bereit zur Montage"
    end
    if A_Farbe = 0 then
    begin
        set A_Status = "In Lackierung"
        use R_Lack for 11
        set color = red
        set A_Status = "Bereit zur Montage"
    end

    move into Q_LackMontage

send to P_Montage
end

begin P_Montage arriving procedure
    set A_Status = "Montage"
    move into Q_Montage
```

```
    use R_Montage for 600
    set A_Status = "Fertig Montiert"
    move into Q_FTS

send to P_Pruef
end

begin P_Pruef arriving procedure

    move into FTS.cp_lack

    if A_Dach = 1 then //Selbststeuerung Part 2, Unterscheidung nach
Dachform
    begin
        travel to FTS.cp_cin

        move into Q_CabrioDach
        set A_Status = "In Dachmontage"
        use R_CabrioDach for 20
        set A_Status = "Dach montiert"
        move into Q_CabrioPruef
        set A_Status = "In Prüfung"
        use R_CabrioPruef for 25

        set A_Status = "verkaufsbereit"

        move into Q_CPuffer

        move into FTS.cp_cout
        if A_Status = "verkaufsbereit" then travel to FTS.cp_wa
    end

    if A_Dach = 0 then
    begin
        travel to FTS.cp_lin

        move into Q_LimousinePruef
        set A_Status = "In Prüfung"
        use R_LimousinePruef for 25

        set A_Status = "verkaufsbereit"

        move into Q_LPuffer

        move into FTS.cp_lout
        if A_Status = "verkaufsbereit" then travel to FTS.cp_wa
    end

    move into Q_WA

send to die
end
```


Anhang D: Source File Systembeispiel 2

```
move into Q_Vorbereitung

    set V_Count = uniform 5,3
    if V_Count <5 then set A_Teil1 = 0 //Teil 1 nicht in Auftrag
vorhanden
    if V_Count >=5 then set A_Teil1 = 1 //Teil 1 in Auftrag vorhanden

    set V_Count = uniform 5,3
    if V_Count <5 then set A_Teil2 = 0 //Teil 2 nicht in Auftrag
vorhanden
    if V_Count >=5 then set A_Teil2 = 1 //Teil 2 in Auftrag vorhanden

    set V_Count = uniform 5,3
    if V_Count <5 then set A_Teil3 = 0 //Teil 3 nicht in Auftrag
vorhanden
    if V_Count >=5 then set A_Teil3 = 1 //Teil 3 in Auftrag vorhanden

    set A_Status = "leer"

send to P_Kommissionierung

end

begin P_Kommissionierung arriving procedure
move into foerderer.sta_boxleer

    if A_Teil1 = 1 then // Scannvorgang des RFID-Tag
        begin
            travel to foerderer.sta_teil1
            move into Q_Teil1
            wait for normal 10,2
            set A_Status = "Teil1 eingeladen"
            move into foerderer.sta_teil1out
            end

    if A_Teil2 = 1 then // Scannvorgang des RFID-Tag
        begin
            travel to foerderer.sta_teil2
            move into Q_Teil2
            wait for normal 10,1
            set A_Status = "Teil2 eingeladen"
            move into foerderer.sta_teil2out
            end

    if A_Teil3 = 1 then // Scannvorgang des RFID-Tag
        begin
            travel to foerderer.sta_teil3
            move into Q_Teil3
            wait for normal 10, 2
            set A_Status = "Teil3 eingeladen"
            move into foerderer.sta_teil3out
            end

travel to foerderer.sta_wa

set A_Status = "versandbereit"
```

```
move into Q_WA
wait for normal 10, 3

move into foerderer.sta_waout //Rückweg der leeren Box
travel to foerderer.sta_boxleer

send to die
end
```

Eidesstattliche Versicherung

Büchner, Lena

Matr.-Nr.: 166659

Ich versichere hiermit an Eides statt, dass ich die vorliegende fachwissenschaftliche Projektarbeit mit dem Titel:

Modellierung und Simulation selbststeuernder miniaturisierter Logistiksysteme

selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Dortmund, 08.08.2017

Unterschrift

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG)

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird gfls. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Dortmund, 08.08.2017

Unterschrift