

Technische Universität Dortmund
Fakultät Maschinenbau
Fachgebiet für IT in Produktion und Logistik

Bachelorarbeit

im Studiengang Wirtschaftsingenieurwesen - Schwerpunkt
Produktionsmanagement

zur Erlangung des akademischen Grades
Bachelor of Science

Thema: Data Farming Konzept in
Tecnomatix Plant Simulation

Autor: Erkut Baydar erkut.baydar@tu-dortmund.de
Matrikelnummer: 150553

Ausgegeben am: 17.12.2015

Eingereicht am: 22.02.2016

Betreuer: Univ.-Prof. Dr.-Ing. Markus Rabe
Dipl.-Inf. Anne Antonia Scheidler

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
Abkürzungsverzeichnis	V
1 Einleitung	1
2 Ereignisdiskrete Simulation	3
2.1 Grundlagen und Nutzen der ereignisdiskreten Simulation	4
2.1.1 Die Simulationsstudie	8
2.2 Werkzeuge zur ereignisdiskreten Simulation	15
2.2.1 Zufallszahlen und Simulationswerkzeuge	17
2.3 Tecnomatix Plant Simulation	19
2.3.1 Datentypen und Elemente in Plant Simulation	23
2.4 Supply-Chains	25
2.5 Data Farming	30
3 Modellierung des Data-Farming-Konzepts	36
3.1 Die Simulationsstudie	36
3.1.1 Zielbeschreibung	36
3.1.2 Aufgabendefinition	37
3.1.3 Datenbeschaffung	40
3.1.4 Datenaufbereitung	41
3.1.5 Systemanalyse	43
3.1.6 Modellformalisierung	48
3.1.7 V & V, Implementierung, Experimente und Analyse	53
3.2 Entwicklung des Simulationsmodells	53
3.3 Verifikation des Simulationsmodells durch funktionalen Test	63
3.4 Ergebnisse der Simulation	68
4 Zusammenfassung und Fazit	73
Anhang A Quellcodes	75
Anhang B Sonstiges	79
Literaturverzeichnis	84
Eidesstattliche Versicherung	88

Abbildungsverzeichnis

2.1	Vorgehensmodell Simulationsstudie nach [Rabe et al., 2008]	14
2.2	Prozesse, Aktivitäten und Ereignisse nach [Eley, 2012]	15
2.3	Beispielhafte Simulationswerkzeugen für Produktion und Logistik nach [Eley, 2012]	18
2.4	Einsatzgebiete der Simulation nach [Pawellek, 2008]	19
2.5	Benutzeroberfläche von Plant Simulation	21
2.6	Der Ereignisverwalter in Plant Simulation	22
2.7	Objekteigenschaften einer Einzelstation in Plant Simulation	23
2.8	Hierarchie einer Supply-Chain nach [Gehr u. Hellingrath, 2007]	27
2.9	Supply-Chain eines Endproduzenten nach [Waters, 2009]	29
3.1	Systemgrenzen	38
3.2	Beispielhafte technische Werte für einen Kühlschrank	41
3.3	Konzeptstruktur	46
3.4	Formales Modell Grafisch	52
3.5	Übersicht des Simulationsmodells in Plant Simulation	54
3.6	Import von Echtdateien in das Modell	57
3.7	Verifikation des Attributs Lieferant	65
3.8	Verifikation der Bearbeitungszeit der Einzelstation Transport_OEM	66
3.9	Ergebnisse der Tabelle Output mit einem Seedwert von 1	70
3.10	Ergebnisse der Tabelle Output mit einem Seedwert von 21	72
B.1	Zielbeschreibung der Arbeit	79
B.2	Aufgabendefinition nach [Wenzel et al., 2008]	80
B.3	Systemanalyse nach [Wenzel et al., 2008]	81
B.4	Modellformalisierung nach [Wenzel et al., 2008]	82
B.5	Implementierung nach [Wenzel et al., 2008]	83

Tabellenverzeichnis

2.1	Typische Ziele und zugehörige Kennzahlen zur Verbesserung von Prozessen nach [Hedtstück, 2013]	8
2.2	Verteilungsfunktionen in Simtalk nach [Eley, 2012]	17
2.3	Vorteile von Plant Simulation nach [Siemens Industry Software GmbH, 2015]	20
2.4	Beispielhafte Datentypen in SimTalk und Plant Simulation nach [Eley, 2012; Siemens PLM Software, 2015]	24
2.5	Elemente in Plant Simulation nach [Siemens PLM Software, 2015]	25
3.1	Aufgabenspezifikation	40
3.2	Intervalle der benutzerdefinierten Attribute	43
3.3	Konzeptmodell	48
3.4	Normalverteilungen der Lieferanten	49
3.5	Transportzeit des Logistikdienstleisters	50
3.6	Zugehörigkeit zu Produktlinien anhand des Attributs Qualität	51
3.7	Verteilung der Lieferabweichung	51
3.8	Formales Modell	52

Abkürzungsverzeichnis

BE	Bewegliche Einheit
HPC	High Performance Computing
KPI	Key Performance Indicator
OEM	Original Equipment Manufacturer
SC	Supply-Chain
SCM	Supply Chain Management
V & V	Verifikation & Validierung
vgl.	Vergleiche
z.B.	Zum Beispiel

1 Einleitung

Die hohe Komplexität der logistischen Systeme, sowie der wachsende Wettbewerb machen die Implementierung von effizienten Prozessen sowohl in Unternehmen als auch in Lieferketten essentiell. Im Zuge der Globalisierung ist der Wettbewerb zwischen Unternehmen gewachsen, wodurch die stetige Verbesserung von Prozessen eine große Rolle in der Unternehmensführung spielt [Siemens Industry Software GmbH, 2015]. Die Realisierung von kurzen Lieferzeiten und das Einhalten von zugesagten Terminen ist für Industrieunternehmen mittlerweile genauso von Wichtigkeit, wie der Stand der Technik ihrer Produkte. Die Planung von Prozessen, der Produktion und von Lieferwegen gewinnt damit zunehmend an Bedeutung [März et al., 2011]. Gerade Planungsprobleme in der Logistik sind aufgrund ihrer Komplexität und ihrem Aufwand für den Einsatz der Simulation geeignet. Hierbei können ganze Standorte, Unternehmen oder SCs (Supply-Chains) als abzubildende Systeme gewählt werden [Eley, 2012].

Die Simulation ist ein Verfahren, welches ein reales oder imaginäres System in einem Modell abbildet, welches experimentell untersucht werden kann, um zu neuen Erkenntnissen über das abgebildete System zu gelangen [Hedtstück, 2013]. Mit Hilfe der Simulation können real existierende Produktionsprozesse, Materialflüsse oder Geschäftsprozesse, verbessert werden. Zudem ist es mit Hilfe der Simulation möglich, Systeme und Prozesse, welche noch nicht existieren, im Vorhinein bestmöglich zu planen. Es wird versucht anhand der Abbildung des realen Systems ein Verständnis für die Problemstellung und die Gegebenheiten zu gewinnen. Gerade um die (internationale) Wettbewerbsfähigkeit eines Unternehmens garantieren zu können, muss in vielen Fällen ein bestmöglicher Betriebszustand erreicht werden [März et al., 2011]. Da die Simulation heutzutage fast ausschließlich rechnergestützt durchgeführt wird, ist die benötigte Zeit für die Analyse eines Systems gering, wodurch schnell Ausgabedaten erzeugt werden können [Eley, 2012].

Die vorliegende Bachelorarbeit beschäftigt sich mit dem Ziel, eine Data-Farming-Umgebung innerhalb einer SC aus der Weißwarenindustrie mit Hilfe der Simulationssoftware „Technomatrix Plant Simulation“ zu erstellen. Primär verfolgt die Arbeit das Ziel mit Hilfe des erzeugten Simulationsmodells so viele Daten wie möglich zu erzeugen. Dies erfolgt durch die Manipulation der Simulationsparameter. Je ausführlicher die erzeugte Datenlandschaft mittels Data Farming ist, desto größer sind die gewonnenen Erkenntnisse.

Zunächst wird der Stand der Technik der ereignisdiskreten Simulation und in SCs aufbereitet. Anschließend wird die zu verwendende Simulationssoftware vorgestellt, eine Übersicht über die gängigsten Elemente erstellt und deren Funktionsweisen erklärt. Nachdem das

Simulationswerkzeug erläutert worden ist, wird auf die Grundlagen des Data Farmings eingegangen. Mit Hilfe der gewonnenen Informationen wird anschließend ein Simulationsmodell erstellt. Es wird erläutert, weshalb die jeweiligen Elemente gewählt worden sind und wie die Auswahl der Parameter zu Stande gekommen ist.

Der Fokus der Entwicklung des Modells liegt darin, ein Data-Farming-Konzept in „Tecnomatix Plant Simulation“ zwecks Entscheidungsunterstützungen zu erstellen. Die Verifikation des Modells erfolgt durch funktionale Tests, welche protokolliert werden. Die Ergebnisse der Arbeit werden im Anschluss zusammengefasst und es wird im Fazit ein Ausblick über Zukunftschancen gegeben.

2 Ereignisdiskrete Simulation

Der deutschen Wirtschaft gehen jährlich Milliardenbeträge durch nicht genutzte Verbesserungspotentiale verloren. Durch den permanenten und rasanten Wandel der Industrie ist die Anpassung der Prozesse der Produktion und Logistik eine komplexe Herausforderung. Diese brachliegenden Milliardenbeträge sorgen im internationalen Vergleich für einen Wettbewerbsnachteil der deutschen Industrie gegenüber den ausländischen Mitbewerbern. Zur Aufdeckung von Möglichkeiten zur Effizienzsteigerung spielt die Simulation eine bedeutende Rolle [Rabe, 2001]. Wollen Wirtschaftsunternehmen an der Spitze der technischen Entwicklung mitwirken, so müssen sie sich permanent modernisieren und den technischen Fortschritt suchen. Die Nutzenproduktivität der Ressourcen muss sich erhöhen, damit die Unternehmen langfristig bestehen können [Rabe, 2003]. Bei der Simulation handelt es sich um eine effektive Problemlösungsmethode die häufig eingesetzt wird, wenn weitreichende Entscheidungen getroffen werden müssen, deren Konsequenzen nicht auf Anhieb ersichtlich sind [Rabe et al., 2008; Wenzel et al., 2008]. Der Begriff der Simulation ist in VDI-Richtlinie 3633 [2013] wie folgt definiert:

„Simulation ist das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. Insbesondere werden die Prozesse über die Zeit entwickelt. Im weiteren Sinne wird unter Simulation das Vorbereiten, Durchführen und Auswerten gezielter Experimente mit einem Simulationsmodell verstanden.“

Die Simulation erlaubt durch ihre planungsunterstützende Eigenschaft, eine Untersuchung der komplexen Dynamik logistischer Systeme, sowie die Bewertung umgestalteter Systeme im „Einsatz vor dem Einsatz“. Aus diesem Grund hat sich die Simulation zu einem essentiellen Planungswerkzeug entwickelt, ohne das eine objektive Analyse des Gesamtprozesses kaum noch möglich ist. Sie stellt als Technologie eine fundierte Entscheidungsunterstützung zur Absicherung der Planungsergebnisse dar. Der Begriff der Simulation ist mittlerweile zu einem Synonym zur Bewältigung etlicher unternehmerischer Probleme geworden und ihre Wirtschaftlichkeit als Methode ist seit langem gut belegt [Rabe, 2001, 2003; Siemens Industry Software GmbH, 2015].

2.1 Grundlagen und Nutzen der ereignisdiskreten Simulation

Die Simulation ist ein Verfahren, bei dem ein Modell, welches auf einem realen oder imaginären System basiert, erstellt und experimentell untersucht wird, um zu neuen Erkenntnissen über das System zu gelangen. Im weitesten Sinne bedeutet Simulation, Experimente an einer geeigneten Nachbildung durchzuführen, statt am Original [Arnold u. Furmans, 2009]. In der digitalen Simulation wird ein System als Computermodell innerhalb einer entsprechenden Software, einem so genannten Simulationswerkzeug, abgebildet [Hedtstück, 2013]. Für den Einsatz von Computern muss das Simulationsmodell in mathematisch-logischer Form vorliegen und in das Simulationswerkzeug implementiert werden [Eley, 2012]. Der Einsatz von Computersimulationsmodellen bietet nach Rabe [2001] etliche Vorteile. Einerseits kann unabhängig von der Art des betrachteten Systems auf eine Vielzahl von universell einsetzbaren Simulationsprogrammen zurückgegriffen werden. Andererseits stellen die Kosten für den Einsatz der Simulation und Modellbildung nur einen Bruchteil dessen dar, was die Untersuchung am realen System betragen würde. Die Möglichkeit der variablen Einstellung des zeitlichen Ablaufs der Betrachtung bietet zudem den Vorteil, dass Beobachtungen ermöglicht werden, die im realen System der menschlichen Wahrnehmung entgehen würden. So können sehr schnell ablaufende Prozesse und Vorgänge stark gedehnt werden, um eine ausführliche Untersuchung zu ermöglichen. Als letzten Vorteil der Computersimulation nennt Rabe [2001], dass das bereits real existierende System keinem Risiko unterzogen wird. Die vorliegende Bachelorarbeit beschäftigt sich ausschließlich mit der digitalen Simulation und daher wird der Zusatz *digital* fort folgend weggelassen.

In Zusammenhang mit der Simulation ist häufig die Rede von Systemen, Modellen und Prozessen. Ein System bezeichnet hierbei eine Menge von Objekten innerhalb einer abgegrenzten Umgebung, welche Attribute besitzen sowie zueinander in Beziehung stehen und sich gegenseitig beeinflussen [Eley, 2012; Košturiak u. Gregor, 1995].

Ist der Zustand des Systems zeitlich veränderbar, so handelt es sich um ein dynamisches System. Diese Zustandsänderungen eines Systems können zu kontinuierlichen oder diskreten Zeitpunkten erfolgen. Aus diesem Grund wird zwischen *kontinuierlichen* und *diskreten* Systemen unterschieden, wobei der Fokus dieser Arbeit auf diskreten Systemen liegt [Hedtstück, 2013; Košturiak u. Gregor, 1995]. Zustandsänderungen werden innerhalb eines dynamischen Systems durch Ereignisse hervorgerufen, welche zeitlich punktuelle Vorgänge darstellen. Der Zeitpunkt, zu dem ein Ereignis eintritt ist oft nicht vorhersehbar, weshalb die in der Praxis vorkommenden Prozesse meist stochastischer Natur sind. Für die Untersuchung eines solchen Prozesses stehen somit nur experimentelle Methoden zur Verfügung. Ist kein Zeitpunkt vorhanden, in dem der Zustand des betrachteten System vom Zufall abhängig ist, so handelt es sich um ein deterministisches System, welches das Gegenstück zu einem stochastischen System darstellt [Hedtstück, 2013]. Folglich wird das

ereignisdiskrete Simulationsmodell nur zu jenen Zeitpunkten betrachtet, in denen sich der Modellzustand ändert [Eley, 2012].

In dem System vorhandene Objekte werden in *temporäre* und *permanente* unterschieden. Temporäre Objekte sind in diesem Zusammenhang Objekte, die nur zeitweise in dem System vorhanden sind, wie z.B. (zum Beispiel) Werkstücke oder Bestellungen. Bei permanenten Objekten handelt es sich um Objekte, die dauerhaft in dem System vorhanden sind. Beispiele für permanente Objekte sind Maschinen, Lager oder Verkehrswege [Hedtstück, 2013].

Der Vorgang des Nachbildens eines Systems wird als *Simulation* bezeichnet, während das dazugehörige Modell *Simulationsmodell* genannt wird [Eley, 2012]. Fortan wird mit dem Begriff *Modell* ein Simulationsmodell bezeichnet. Ein Modell spiegelt das zeitliche und logisch-strukturelle Verhalten eines Systems wider und stellt ein abstraktes Abbild eines Systems dar. Mithilfe der Durchführung von *Experimenten* an einem Modell können unbekannte Eigenschaften oder Gesetzmäßigkeiten des Systems ermittelt werden. Hierzu reicht meist ein vereinfachtes Abbild des zu betrachtenden Systems [Hedtstück, 2013; Eley, 2012; Rabe, 2003].

Oftmals sind die zu untersuchenden Systeme zu komplex, um vollständig erfasst und abgebildet zu werden. Aus diesem Grund findet beim Modellbildungsprozess eine Reduktion auf die wesentlichen Elemente und Beziehungen des Systems statt. Somit kann es sein, dass sich das Modell hinsichtlich der zu untersuchenden Eigenschaften innerhalb eines vom Untersuchungsziel abhängigen Toleranzrahmens unterscheidet [Eley, 2012; Rabe, 2003]. Die Modellbildung, bei der ein reales System in ein Modellsystem überführt wird, erfolgt mit der Hilfe von Beschreibungsmitteln. Diese bestehen aus einem Satz von Konstrukten, welche Elemente und Regeln zur Verknüpfung dieser Elemente umfassen. Die erstellten Modelle werden meist Top-Down eingesetzt, wobei unterschiedliche Betrachtungsebenen mit unterschiedlichen Abstraktionsgraden verfolgt werden. Schneider [2008] nennt unterschiedliche Betrachtungsebenen von Simulationsmodellen:

- Strategische Simulation auf Konzernebene (z.B. Lieferketten, Werksverbund)
- Konzeptionelle Simulation innerhalb der Produktionsstätte (z.B. Zusammenwirken verschiedener Stationen)
- Operative Simulation zur Fertigungs- und Logistikstruktur
- Geometriebezogene Simulation (z.B. Kollisionsbetrachtungen in einer Roboterzelle)
- Prozessbezogene Simulation (z.B. Untersuchungen von Fügeprozessen)

Durch den Einsatz von Simulation soll die Planungssicherheit erhöht und Planungsfehler vermieden werden. Außerdem sollen die dynamischen Zusammenhänge des realen Systems besser verstanden werden, um die Auswirkungen bestimmter Maßnahmen genau abschätzen zu können. Gerade bei einem hohen Investitionsvolumen für den geplanten Prozess

beziehungsweise das Projekt in Kombination mit der hohen Komplexität dessen, hat der Einsatz der Simulation einen hohen Nutzen. Wichtig für die Qualität der Ergebnisse ist die Qualität der Eingangsdaten, wobei die Datenversorgung ein großes Problem der Simulation darstellt [Schneider, 2008].

Laut Eley [2012] sind während der Erstellung eines Modelles nicht alle benötigten Eingabedaten verfügbar. Ist eine empirische Erhebung der Daten nicht möglich, so müssen statistische Verteilungen verwendet werden. Es ist anzumerken, dass ein Simulationsmodell eine sorgfältige Planung nicht ersetzen kann und somit lediglich als Unterstützung innerhalb der Planung angesehen werden sollte.

Branchenunabhängig bietet die Simulation bei dem Einsatz in Planung und Optimierung logistischer Systeme viele Vorteile. Zum einen können Ideen analysiert werden, ohne dass in den laufenden Prozess eingegriffen werden muss [Siemens Industry Software GmbH, 2015]. Außerdem können alternative Vorgehensweisen auf ihre Qualität und Quantität untersucht werden und mithilfe von Kennzahlen beziehungsweise KPI (Key Performance Indicator) auf ihre Leistungsfähigkeit beurteilt werden (vgl. (vergleiche) Abschnitt 2.1). Die mittels der Simulation gewonnenen KPI sind stets mit der für das Modell gewählten Abstraktion in Verhältnis zu setzen. Somit sind Aussagen, die durch Experimente am Modell gewonnen worden sind, nicht unbedingt zeitgleich auch Aussagen über das dem Modell zugrunde liegenden Systems. Aus diesem Grund müssen die ermittelten Aussagen unbedingt auf ihre Übertragbarkeit zu überprüfen. Es ist zudem zu bestimmen, welche Eingangsgrößen das Ergebnis beeinflussen und in welchem Maß diese Größen innerhalb des Modells berücksichtigt worden sind [Kuhn, 1998].

Neben der Analyse von Alternativen bietet die Simulation eine objektive Grundlage für Entscheidungen und die Möglichkeit auf Veränderungen schnell zu reagieren [Eley, 2012]. Die Simulation kommt gerade dann zum Einsatz, wenn es nicht möglich ist ein Problem mit analytischen Werkzeugen zu lösen [Klug, 2010; Arnold u. Furmans, 2009].

Im Gegensatz zu analytischen Methoden werden bei der Simulation die systemspezifischen dynamischen Abhängigkeiten und Wechselwirkungen über die Zeit modelliert sowie stochastische Einflüsse berücksichtigt. Dies stellt einen besonderen Vorteil gegenüber den analytischen Methoden dar [Wenzel et al., 2008].

Über die Jahre haben sich Simulationssysteme durchgesetzt, die mit Hilfe von Methodenbausteinen arbeiten. Hierbei können vom Anwender Bausteine über eine grafische Oberfläche ausgewählt und miteinander verbunden werden. Der Aufwand der Simulation wird hierdurch deutlich verringert und die Einstiegsbarriere in das Themengebiet der Simulation wird erheblich verringert, wodurch auch weniger erfahrene Anwender mit akzeptablem Einarbeitungsaufwand erste Modelle erstellen können. Außerdem wird die einfache Anpassung an unterschiedliche Fertigungsarten sowie Detaillierungstiefen durch eine graphische Repräsentation ermöglicht. Auch das in der vorliegenden Arbeit verwendete Simulationswerkzeug „Tecnomatix Plant Simulation“ arbeitet nach dem Prinzip der Funktionsbausteine [Rabe, 2003; Bangsow, 2011].

Der Begriff der *Steuerung* bezeichnet in dem in dieser Arbeit betrachteten Rahmen zum einen die Vorrichtungen, mit denen der Bewegungsablauf eines Betriebsmittels gelenkt wird und zum anderen die allgemeine Lenkung der Produktionsdurchführung. Die Übergänge zwischen den beiden Bedeutungen können fließend sein. Steuerungen lassen sich in *dispositive* und *operative* Steuerung untergliedern. Mit einer dispositiven Steuerung ist die klassische Fertigungssteuerung gemeint, während mit dem Begriff der operativen Steuerung die Fähigkeit gemeint ist, dass der Ablauf vollständig über Algorithmen formulierbar ist. Für automatisch ablaufende Simulationen ist die operative Steuerung unabdingbar [Rabe, 2003]. Der Begriff der Steuerung wird in [DIN 19226, 1968] wie folgt definiert:

„Das Steuern - die Steuerung - ist der Vorgang in einem System, bei dem eine oder mehrere Größen als Eingangsgrößen andere Größen als Ausgangsgrößen auf Grund der dem System eigentümlichen Gesetzmäßigkeit beeinflussen. Kennzeichen für das Steuern ist der offene Wirkungsablauf über das einzelne Übertragungsglied oder die Steuerkette.“

Charakteristisch für eine Steuerung ist der offene Wirkungsweg, wodurch die von den Eingangsgrößen beeinflussten Ausgangsgrößen nicht fortlaufend und nicht wieder über dieselben Eingangsgrößen auf sich selbst wirken können [Rabe, 2003]. Ist der Wirkungsweg geschlossen, so handelt es sich um eine Regelung, die in [DIN 19226, 1968] folgendermaßen definiert wird:

„Das Regeln - die Regelung - ist ein Vorgang bei dem eine Größe, die zu regelnde Größe, fortlaufend erfasst, mit einer anderen Größe, der Führungsgröße, verglichen und abhängig vom Ergebnis dieses Vergleichs im Sinne einer Angleichung an die Führungsgröße beeinflusst wird. Der sich dabei ergebende Wirkungsablauf findet in einem geschlossenen Kreis, dem Regelkreis, statt.“

Im Vergleich zu einer Steuerung erfolgt somit bei einer Regelung eine Rückkopplung. Unter dem Begriff der Steuerung kann auch jede Vorrichtung und jeder Vorgang verstanden werden, in denen Entscheidungen hinsichtlich der Durchführung von Abläufen getroffen werden. In Abhängigkeit vom Systemzustand können somit Zustandsveränderungen hervorgerufen werden, welche keine Rückkopplung erfordern. Gerade diese Definition verdeutlicht die Bedeutung des Begriffes der Steuerung innerhalb der ereignisdiskreten Simulation, da die Begriffe Steuerung, Zustandsänderung und Ereignis miteinander verbunden werden. In diesem Zusammenhang ist noch der Begriff der *Steuerungsregel* zu erwähnen, welche eine Bedingung als Funktion des Systemzustandes sowie eine Ausführungsanweisung, die eine Änderung des Systemzustandes bewirkt. Sowohl Bedingung als auch Ausführungsanweisung können hierarchisch aus mehreren Einzelbedingungen zusammengesetzt sein. Diese Steuerungsregeln betreffen nach Rabe [2003] Entscheidungen zwischen zwei Bausteinen innerhalb des Simulationsmodells.

Im folgenden Abschnitt werden die Simulationsstudie und das in dieser Arbeit verwendete Vorgehensmodell näher betrachtet.

2.1.1 Die Simulationsstudie

Mit dem Begriff der Simulation ist im Allgemeinen eine ausführliche Simulationsstudie gemeint. Sie stellt heutzutage die häufigste Form des Einsatzes der Simulation dar. Die Studie kann sowohl in ein größeres Projekt eingebunden sein oder als alleinstehendes Projekt durchgeführt werden. Hierbei wird ein Prozess als allgemeines Ablaufschema aufgefasst und mithilfe wissenschaftlicher Methoden versucht eine weitestgehend zufriedenstellende Konfiguration zu ermitteln.

Die Ziele einer Simulationsstudie sind unter anderem die Prozessverbesserung, die Beantwortung vorab festgelegter Fragestellungen und die Bearbeitung anstehender Aufgaben der Produktions- und Logistiksystemplanung [Hedtstück, 2013; Wenzel et al., 2008; Pawellek, 2008]. Im Rahmen der Prozessverbesserung werden Kenngrößen beziehungsweise so genannte KPI festgelegt, welche der Messung der Leistungsfähigkeit von Prozessen dienen. Tabelle 2.1 listet einige KPI auf:

Zielsetzung	Kennzahlen
Niedrige Kosten	Energiekosten, Lagerhaltungskosten, Personalkosten
Geringe Zeit	Rüstzeit, Durchlaufzeit
Hohe Kapazität	Auslastung
Hohe Qualität	Fehlerquote, Lebensdauer

Tabelle 2.1: Typische Ziele und zugehörige Kennzahlen zur Verbesserung von Prozessen nach [Hedtstück, 2013]

Im Normalfall ist es nicht möglich, dass alle Ziele gleichzeitig erreicht werden können, da es zwischen einigen von ihnen zu Zielkonflikten kommt. Simulationswerkzeuge stellen hervorragende Analyseinstrumente zur Ermittlung von Kennzahlen von Prozessen dar. Mithilfe dieser Werkzeuge ist es möglich, verschiedene Szenarien durchzuspielen und die Auswirkungen auf die Kennzahlen zu beurteilen [Hedtstück, 2013].

Im Rahmen der Simulation und der Prozessverbesserung fällt oft der Begriff der Optimierung, welcher einer Abgrenzung bedarf. Mit dem Begriff der Optimierung in Zusammenhang mit Simulationsstudien nicht die Suche nach einem absoluten Optimum gemeint. Der Begriff der Optimierung bezeichnet in diesem Fall die Verbesserung eines Prozesses. Ist mit dem Begriff der Optimierung eine Verbesserung des Ist-Zustands gemeint, so kann in diesem Falle die Simulation zur Optimierung genutzt werden, um die beste Alternative aus mehreren Lösungen zu ermitteln. Mithilfe geeigneter Methoden versucht man systematisch die Kenngrößen ausreichend genug an die Zielvorgaben anzunähern. Dies erfolgt durch die geeignete Wahl der Parameter und deren Variation. Im Normalfall sind hierzu mehrere Verbesserungsschritte notwendig, weshalb die Simulation den experimentellen

Optimierungsverfahren zugeordnet wird [Pawellek, 2008; Schneider, 2008; Eley, 2012]. Bei der Suche nach der *besten* Parametereinstellung kann die Simulation den Planer unterstützen. Das Modell gibt Auskunft über die kritischen Grenzen der Parameter und ermöglicht auf Basis von „Wenn-Dann-Szenarien“ eine Wirtschaftlichkeitsbetrachtung des Gesamtprozesses [Klug, 2010]. Um einen Prozess optimieren zu können, wird vorausgesetzt, dass die Erreichung der Ziele messbar ist. Nur so kann beurteilt werden, wann das Ziel ausreichend erreicht worden ist [Hedtstück, 2013].

Klassischerweise ist jedoch mit dem Begriff der Optimierung die Anwendung eines mathematischen Optimierungsverfahrens gemeint. Bei genauerer Betrachtung dieses Umstands wird der Unterschied zwischen den beiden Begriffen deutlich. Die Simulation dient der Überprüfung der Leistungsfähigkeit, sowie der experimentellen Verbesserung eines Systems, während bei der klassischen Optimierung mithilfe eines Optimierungsverfahrens die *mathematisch optimale* Lösung zu einem logistischen Problem ermittelt wird. Im Gegensatz zu mathematischen Optimierungsverfahren ist die Simulation kein Verfahren, welches aus sich heraus ein Optimum liefern kann. Sie dient dem Simulationsanwender lediglich zur Beurteilung eines gegebenen Szenarios, indem Kennzahlen erzeugt werden, welche sonst nur im realen System ermittelbar wären. Darüber hinaus können auch Kennzahlen erzeugt werden, die in der Realität aufgrund fehlender Automatisierung und durch organisatorische Probleme nicht ermittelbar wären [Kuhn, 1998]. Simulation und die klassische Optimierung schließen sich jedoch nicht aus, sondern können sich vielmehr ergänzen. Das Optimierungsverfahren kann in das Simulationsmodell integriert werden oder umgekehrt [Eley, 2012]. Abbildung 2.1 stellt ein einfaches und überschaubares Vorgehensmodell zur Durchführung einer Simulationsstudie unter Einbindung von V & V-Techniken nach Rabe et al. [2008] dar.

Die Dokumentation der Ergebnisse wird nicht explizit als Phase ausgewiesen, da diese in *jeder* Phase vorhanden ist. Das Vorgehensmodell betrachtet nur die Aufgaben, die *nach* der Beauftragung der Simulationsstudie anfallen. Im Gegensatz zu anderen Vorgehensmodellen ist dieses Modell durch die kontinuierliche und konsequente Einführung von Phasenergebnissen gekennzeichnet. Die einzelnen Projektphasen werden als Ellipsen dargestellt. Jeder Projektphase des Vorgehensmodells wird ein Phasenergebnis zugeordnet, welche durch Rechtecke dargestellt werden. Bei den ermittelten Phasenergebnissen kann es sich um Dokumente und Modelle handeln, oder eine Kombination von beidem. Rücksprünge zu vorherigen Phasen und Iterationen sind prinzipiell möglich, da das vorgestellte Vorgehensmodell eine prinzipielle Abfolge zur Bearbeitung der jeweiligen Phasen darstellt [Wenzel et al., 2008].

In der Definitions- und Angebotsphase wird eine zunächst grobe *Zielbeschreibung* sukzessive vervollständigt und dient als Orientierung. In der Zielbeschreibung wird zunächst das zu Untersuchende System mit seinen typischen Ausprägungen kompakt beschrieben. Im Anschluss werden Aufgabenstellung und der Zweck des Simulationsmodells formuliert. Bei mehreren konkurrierenden Ziel, müssen diese priorisiert werden. Diese Phase gilt als

Einstieg in das eigentliche Projekt [Rabe et al., 2008; Wenzel et al., 2008].

Die Durchführung der eigentlichen Simulationsstudie beginnt mit der Konkretisierung der Aufgabe im Rahmen der *Aufgabendefinition*. Es wird die Zielbeschreibung und die zu untersuchenden Systeme und Teilsysteme konkretisiert. Hierbei werden unter anderem die Systemgrenzen festgelegt, das grobe Verhalten des Systems beschrieben und relevante Systemkomponenten aufgelistet. Zudem wird der Detaillierungsgrad des zu erstellenden Modells festgelegt. Hierbei müssen die zu beschaffenden Daten hinsichtlich ihrer Art, ihrem Umfang und ihrer Granularität beschrieben werden. Eine weitere Aufgabe der Aufgabenspezifikation ist die Benennung der zu verwendenden Informations- und Datenquellen und die Nutzung von externen Schnittstellen. Der Modellzweck und dessen Nachnutzung werden konkretisiert und es werden die Anforderungen an die Lösungsmethode festgelegt [Wenzel et al., 2008]. Das Ziel dieser Phase ist die *Aufgabenspezifikation*, welche sich dadurch auszeichnet, dass eine von dem Auftragsgeber und -nehmer gemeinsam verstandene und detaillierte Aufgabenbeschreibung vorliegt. Es wird ein Grundverständnis erzeugt, ob und wie die Aufgabe zu lösen ist [Rabe et al., 2008].

Anschließend folgt die Phase der *Systemanalyse*. Hierbei wird aus dem realen System das *Konzeptmodell* abgeleitet und wesentliche Ziele zur Erstellung des Simulationsmodells festgelegt. Das Ziel dieser Phase ist die Festlegung, welche Elemente des realen Systems in welcher Detaillierung und unter Verwendung welcher Mechanismen modelliert werden sollen. Die Berücksichtigung von irrelevanten Aspekten senkt die Effizienz der Modellbildung und -ausführung. Hingegen entsteht bei der fehlenden Berücksichtigung von relevanten Zusammenhängen ein für die Aufgabenstellung ungeeignetes Modell. Dies birgt die Gefahr, dass aus dem Simulationsmodell falsche Ergebnisse entstehen, welche zu falschen Entscheidungen führen können [Rabe et al., 2008]. Im entstehenden Konzeptmodell werden die Strukturen des Modells sowie der Teilmodelle beschrieben und vorzugsweise grafisch dargestellt. Dies kann in Form von Flussdiagrammen, Netzplänen oder Prozesskettenmodellen erfolgen. Die grafische Darstellung enthält zudem auch die Material- und Informationsflüsse des Systems mit ihren Eingabe- und Ausgabegrößen [Wenzel et al., 2008]. Laut Rabe et al. [2008] ist der Begriff des Konzeptmodells in der Literatur nicht klar definiert. Jedoch ist man sich einig, dass das Konzeptmodell eine Definition erstellen soll, was zu modellieren ist und wie diese Modellierung umzusetzen ist. Es müssen jedoch keine Aussagen darüber getroffen werden, wie das ausführbare Simulationsmodell in das verwendete Simulationswerkzeug zu implementieren ist.

Um ein entsprechendes Simulationsmodell erstellen zu können, müssen im Rahmen der *Datenbeschaffung* so genannte *Rohdaten* ermittelt werden. Diese Daten und Informationen werden im Rahmen einer Simulationsstudie vom Auftragsgeber zu Verfügung gestellt und es kann sich hierbei um z.B. Lagerbestandslisten, Auftragslisten oder Leistungsdaten handeln. Ergänzend zu den Angaben aus der Aufgabenspezifikation, muss das konkrete Vorgehen der Datenbeschaffung dokumentiert werden [Wenzel et al., 2008].

Die Rohdaten werden im Rahmen der *Datenaufbereitung* in *aufbereitete Daten* überführt.

Die aufbereiteten Daten entstehen durch die Bereinigung, Strukturierung und Verdichtung der Rohdaten. Fehlerhafte Datensätze sind mittels Plausibilitäts- und Konsistenzprüfungen zu ermitteln und gegebenenfalls zu korrigieren. Fehlende Daten müssen neu beschafft werden. Die Verdichtung der Rohdaten kann in Form von statistischen Verteilungen erfolgen, wobei die Verteilungsfunktion, die Koeffizienten sowie die eingesetzten statistischen Verfahren zu dokumentieren sind. Das Ergebnis der Verdichtung ist ebenfalls zu protokollieren. Gerade bei Planungsstudien kann es dazu kommen, dass bestimmte Eigenschaften des Systems unbekannt sind, weshalb gemeinsam von Auftragsgeber und -nehmer Annahmen zu treffen sind [Wenzel et al., 2008].

In der Phase der *Modellformalisierung* wird das Konzeptmodell in das *formale Modell* überführt, welches sich im Idealfall ohne weiteres implementieren lässt. In der Praxis entsteht ein formales Modell nur selten, da im Rahmen der Systemanalyse, das Konzeptmodell derart formalisiert wird und auch das Modellierungskonzept der Simulationssoftware mit einschließt. Im Sinne der V & V ist diese Phase dennoch von Bedeutung, weshalb Rabe et al. [2008] davon abrät, auf sie zu verzichten. Als eine der wichtigen Aufgaben der Modellformalisierung, gilt die Gewährleistung der Implementierbarkeit der im Konzeptmodell beschriebenen Abläufe. Elemente und Beziehungen aus dem Konzeptmodell werden weiter in Richtung eines formalen Entwurfes entwickelt, so dass Simulationsfachleute ohne weitere Analysen, die Implementierung vornehmen können. Zusätzlich steigert die Durchführung der Modellformalisierung die Qualität der Simulationsstudie und es kann bei späteren Unstimmigkeiten festgestellt werden, ob das falsche Modell erstellt oder die Implementierung fehlerhaft durchgeführt worden ist. [Rabe et al., 2008; Wenzel et al., 2008].

Im Rahmen der *Implementierung* entsteht als Ergebnis das *ausführbare Modell*, welches von Simulationsfachleuten oder Softwareexperten durchgeführt wird. Abhängig vom Modellierungskonzept des verwendeten Simulationswerkzeuges ist die Nutzung einer Programmiersprache nicht zwingend notwendig. So können vordefinierte Bausteine genutzt werden, um reale Elemente abzubilden [Rabe et al., 2008]. In dieser Phase muss der Auftragsnehmer das formale Modell in eine tatsächliche Implementierung umsetzen. Aus der Praxis ist hierbei die so genannte 80/20-Regel hervorgegangen, die besagt, dass in der Regel circa 80% der benötigten Komponenten im Simulationswerkzeug enthalten sind und 20% durch die Schaffung zusätzlicher Komponenten oder der Manipulation bestehender erstellt werden müssen. Die Nutzung von vorgegebenen Bausteinen hat den Vorteil, dass diese bereits durch den Entwickler des Simulationswerkzeuges hinsichtlich ihrer Qualität geprüft worden sind und somit von einer mehr oder weniger guten Qualität ausgegangen werden kann. Diese Qualitätsaussage ist nur auf bereits bestehende Bausteine bezogen, weshalb beachtet werden muss, dass die Entwicklung von neuen Erweiterungen und Komponenten Qualitätsprobleme aufweisen kann. Aus diesem Grund ist die ausführliche Dokumentation der neuen Komponenten und der entsprechend getroffenen Annahmen wichtig [Wenzel et al., 2008; Rabe et al., 2008].

Der letzte Schritt des Vorgehensmodells ist die Phase *Experimente und Analyse* und stellt den eigentlichen Nutzen der Simulationsstudie dar. Die Voraussetzung zur Durchführung dieser Phase ist das Vorhandensein der aufbereiteten Daten und des ausführbaren Modells, welche zusammengeführt werden. Diese Phase enthält mehrere Schritte, welche iterativ durchgeführt werden. Zum einen werden Experimentenpläne erstellt und Hypothesen festgelegt, die überprüft werden sollen. Zum anderen werden die Experimente durchgeführt und die Ergebnisse geordnet dokumentiert. Der letzte Schritt dieser Phase ist die Ableitung von Schlussfolgerungen über das reale System, welche die festgelegten Hypothesen bestätigen oder widerlegen und quantitative Aussagen liefern. Die Ergebnisse der Experimente können zudem Hinweise zur Verbesserung des Systems liefern [Rabe et al., 2008]. Experimente werden in der VDI-Richtlinie 3633, Blatt 3 [2013] als „[...] eine Anzahl von Simulationsläufen mit systematischer Parametervariation“ definiert. Ein Simulationslauf oder ein Experiment bildet das Verhalten eines Systems über einen bestimmten Zeitraum nach und ist ein zielgerichtetes Ausprobieren [Kuhn, 1998]. Das Modell wird pro Simulationslauf über eine bestimmte Zeitspanne genau *einmal* ausgeführt. Als *Simulationszeitraum* wird die Zeitspanne, über die das System analysiert wird bezeichnet. Die *Simulationszeit* hingegen, beschreibt die im Simulationsmodell fortschreitende Zeit. Die Simulationszeit stellt das Gegenstück zur *Rechenzeit* dar. Diese stellt die Zeit dar, die für die Analyse eines Systems vom Computer benötigt worden ist. Da stochastische Aspekte eine große Rolle in Simulationen spielen können, ist in solchen Fällen eine hinreichende Anzahl an Wiederholungen eines Simulationslaufs mit verschiedenen Startwerten der Zufallsverteilungen notwendig. Das Simulationsmodell verhält sich dann bei jedem erneuten Start anders und führt zu unterschiedlichen Ergebnissen (vgl. Abschnitt 2.2.1). Bei einem deterministischen Simulationsmodell ist nur ein einziger Simulationslauf nötig, um Aussagen über das betrachtete System treffen zu können [Eley, 2012].

Zur Modellherstellung müssen die Korrektheit und Gültigkeit des Modells geprüft werden. Die Korrektheit bedeutet in diesem Zusammenhang, dass das Modell nach den vorgegebenen syntaktischen Regeln umgesetzt worden ist. Zudem muss das Modell semantisch Korrekt sein und darf keine logischen Fehler enthalten. Überprüft wird die Korrektheit eines Modells mit der Verifikation. Ein Modell wird als gültig bezeichnet, wenn es sich zeitlich so verhält wie das reale System und mit diesem in den strukturellen Eigenschaften übereinstimmt. Die Gültigkeit eines Modells wird mit der Validierung überprüft.

Die Besonderheit des in Abbildung 2.1 gezeigten Vorgehensmodells ist die in jeder Phase stattfindende V & V, weshalb keine eigene Phase der V & V vorhanden ist. Die V & V der Daten und Modelle ist in diesem Vorgehensmodell phasenbegleitend und somit kein einmaliger Vorgang kurz vor dem Ende der Simulationsstudie. Ergebnisse und Teilergebnisse einer Projektphase werden mithilfe von V & V-Methoden auf Eignung, Plausibilität und Vollständigkeit geprüft. Zudem erfolgt eine Überprüfung der Richtigkeit und Vollständigkeit der Übernahme der Ergebnisse aus dem vorherigen Schritt. Es kann somit durchaus vorkommen, dass ein Rücksprung zu einer vorherigen Projektphase erforderlich wird. Wird

die V & V erst nach dem Abschluss der Modellbildung durchgeführt, können Fehler nicht frühzeitig genug erkannt und behoben werden [Rabe et al., 2008; Wenzel et al., 2008]. In Zusammenhang mit der V & V von Simulationsmodellen sind *Tests* zu erwähnen. Tests stellen Mittel zur Verifikation und Validierung von Modellen dar und werden von Beginn der Entwicklung bis hin zur Fertigstellung eines Modells eingesetzt. Mit Hilfe von Tests soll sichergestellt werden, dass das zu untersuchende System hinreichend präzise wiedergegeben wird (*sachliche Korrektheit*), das Modell die für die Zielstellung erforderlichen Funktionen beinhaltet (*funktionale Korrektheit*) und die erforderlichen Randbedingungen erfüllt (*technische Korrektheit*). Der Großteil von Tests lässt sich nicht eindeutig der V & V zuordnen, da sie eng mit beiden Verfahren verbunden sind. Ein einziger Test mit negativem Ergebnis weist nach, dass das untersuchte Modell nicht für die Aufgabenstellung verwendet werden kann, während ein Test mit positivem Ergebnis nur nachweist, dass das Modell unter den Testbedingungen in erwarteter Weise reagiert. Aus diesem Grund, kann mit Hilfe von Tests die Gültigkeit eines Modells niemals nachgewiesen werden. Durch eine Anzahl sorgfältig ausgewählter Testmethoden kann jedoch die Gültigkeit wahrscheinlich gemacht werden. Für die Durchführung von Tests sind geeignete Techniken erforderlich, wobei ein Test auch mehrere Techniken gleichzeitig verwenden kann. Beispiele für Techniken sind z.B. der *Vergleich* eines Modells mit einem anderen, gültigen Modell, die Beobachtung des Modells in der *Animation* und die Überprüfung des Modells in Grenzsituationen (*Grenzwerttest*). Der *Grenzwerttest* setzt voraus, dass die Ergebnisse des Simulationsmodells auch für Kombinationen extremer Eingabedaten und Parameter plausibel sein müssen. Eingabegrößen und Parameter werden so gesetzt, dass das Verhalten des Modells besser vorhersagbar wird. Z.B. können hierbei alle Losgrößen in einem Simulationsmodell für ein Produktionssystem auf den kleinsten vorhergesehenen Wert gesetzt werden, so dass Rüst- und Bearbeitungsvorgänge besser abgeschätzt werden und mit den Zeiten im realen System verglichen werden können.

Mit der *Beobachtung des Modells in der Animation* kann nachgewiesen werden, dass das Verhalten des Modells in bestimmten Situationen *ungültig* ist. Es wird beobachtet, ob sich das Simulationsmodell in dem betrachteten Zeitabschnitt plausibel verhält und ob Unterschiede zum realen System auftreten. Selten auftretende Fehler werden mit hoher Wahrscheinlichkeit nicht erkannt und es ist auch kein Rückschluss auf das über den betrachteten Zeitraum hinausgehende Verhalten möglich. Durch geeignete Maßnahmen kann die Wirksamkeit der Animation verbessert werden. Zum einen kann das System bereits bei der Modellerstellung mit dem Ziel erstellt werden, das Systemverhalten transparent darzustellen. Zum anderen kann die Simulation mit verlangsamer Geschwindigkeit durchgeführt werden, so dass es dem Beobachter möglich ist, wesentliche Vorgänge zu verfolgen. Die Stärke der Animation liegt somit in der Überprüfung des Modellverhaltens über kurze Zeitabschnitte.

Bei dem *Vergleich mit anderen Modellen* werden die Ergebnisse des Modells für festgelegte Eingabedaten mit den Ergebnissen des einfacheren Modells verglichen. Aufgrund des

hohen Aufwands ist die Entwicklung von zwei Simulationsmodellen für das selbe System selten. Häufig wird diese Technik in Kombination mit weiteren Techniken, wie z.B. dem Grenzwerttest, durchgeführt [Rabe et al., 2008].

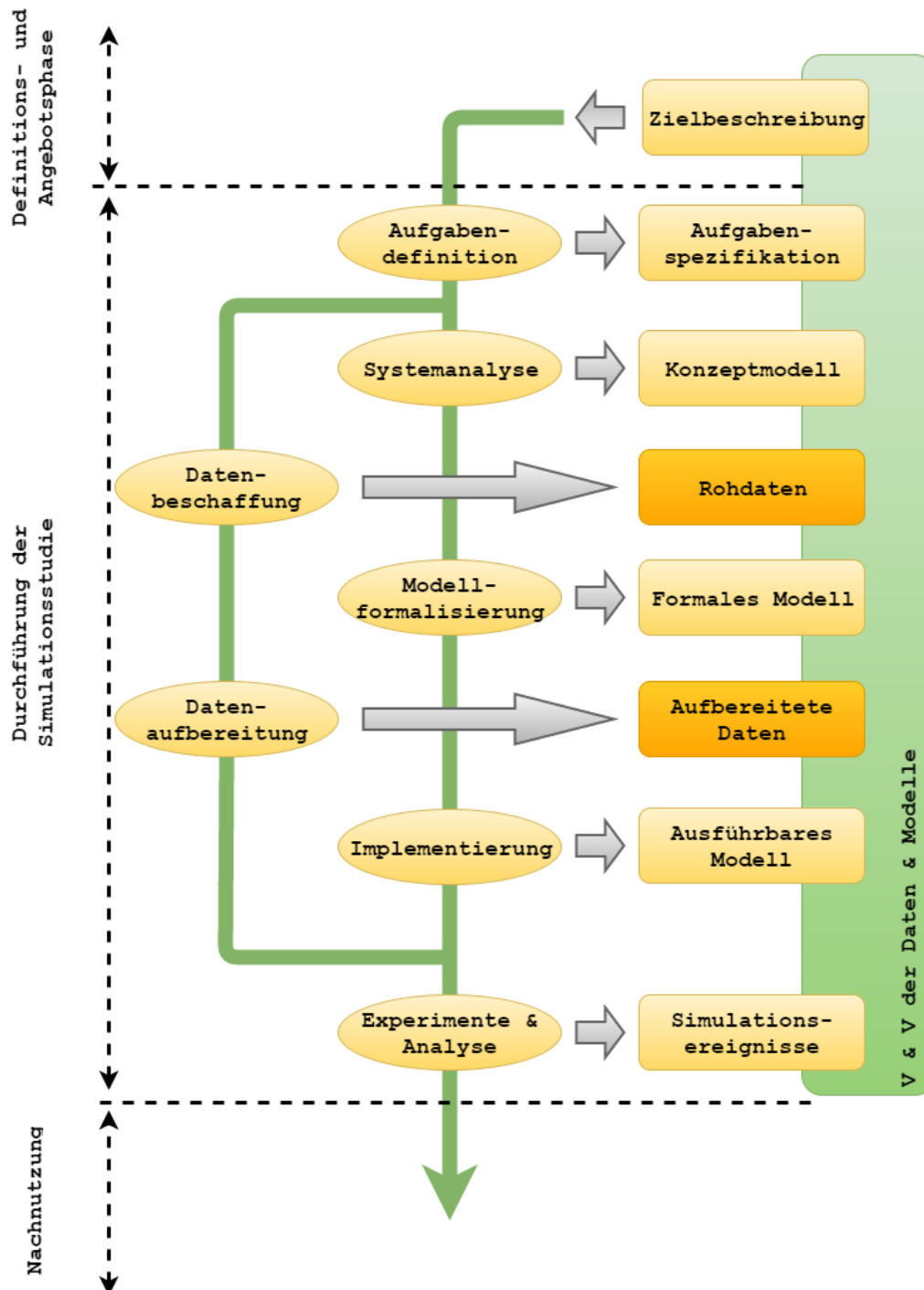


Abbildung 2.1: Vorgehensmodell Simulationsstudie nach [Rabe et al., 2008]

Für die Durchführung solcher Simulationsprojekte werden Simulationswerkzeuge benötigt, welche im folgenden Abschnitt erklärt werden.

2.2 Werkzeuge zur ereignisdiskreten Simulation

Wie bereits in Abschnitt 2.1 erläutert, kann zwischen *kontinuierlichen* und *ereignisdiskreten* Simulationsmodellen unterschieden werden. Die vorliegende Bachelorarbeit beschäftigt sich ausschließlich mit der ereignisdiskreten Simulation, da diese besonders geeignet ist, die komplexen Eigenschaften von logistischen Systemen abzubilden. Die Mehrzahl der Simulationseinsätze in der Produktionslogistik basiert auf der ereignisdiskreten Simulation. Bei der ereignisdiskreten Simulation ändern sich die Modelle nur zu bestimmten, diskreten Zeitpunkten. Aus diesem Grund wird das Simulationsmodell nur zu diesen diskreten Zeitpunkten betrachtet [Rabe, 2003].

Zur Veranschaulichung der ereignisdiskreten Simulation, wird ein Beispiel aus der Realität betrachtet. Bei diesem Beispiel wird ein Werkstück auf einer Maschine bearbeitet. Hierzu wird das Werkstück zum Zeitpunkt t_1 auf die Maschine umgelagert und auf dieser Δt Zeiteinheiten bearbeitet. In der ereignisdiskreten Simulation sind für dieses Beispiel nur die Zeitpunkte von Bedeutung, in denen das Werkstück auf die Maschine und von der Maschine umgelagert werden, sowie der Zeitpunkt zu dem die Bearbeitung des Werkstücks beendet ist. Der nächste Zeitpunkt der von Interesse ist, ist, wenn die Bearbeitung des Werkstücks beendet ist, also der Zeitpunkt $t_2 = t_1 + \Delta t$. Erst zu diesem Zeitpunkt verändert sich das System wieder. Die Zeitpunkte t_1, t_2, \dots , in denen Änderungen der Modellzustände stattfinden nennt *Ereignisse*. Ereignisse lösen *Aktivitäten* aus oder beenden sie. In dem gewählten Beispiel ist die Bearbeitung des Werkstücks eine *Aktivität* [Eley, 2012]. Abbildung 2.2 verdeutlicht diesen Zusammenhang. Die Abbildung zeigt einen Prozess, der aus der Ankunft eines Auftrags, der Durchführung von zwei Arbeitsgängen und der Beendigung dieser besteht. Es finden zwei Aktivitäten sowie fünf Ereignisse statt, die zu dem Prozess zusammengefasst werden.

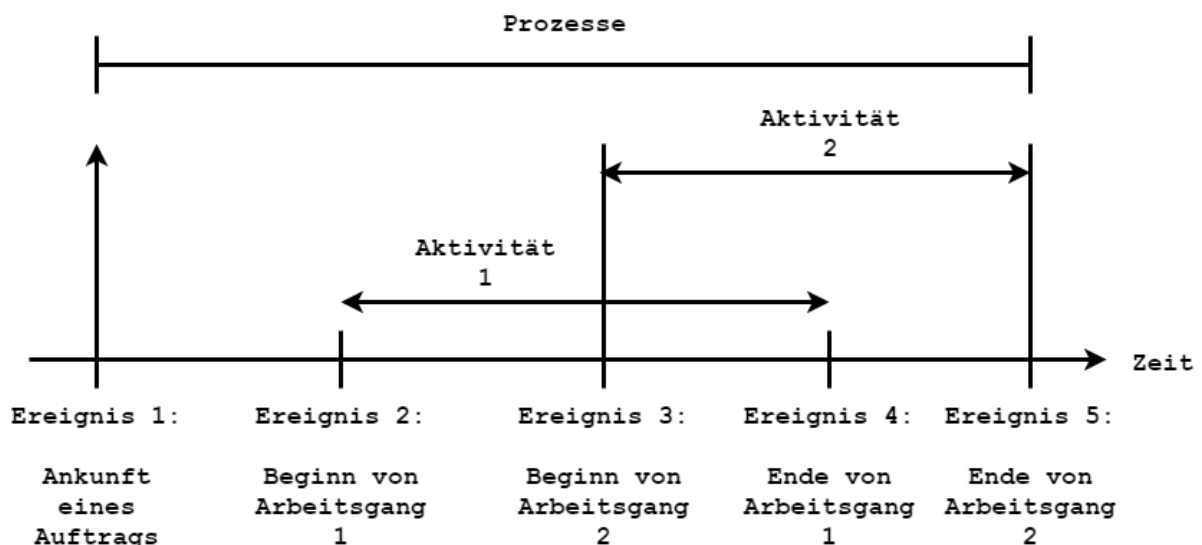


Abbildung 2.2: Prozesse, Aktivitäten und Ereignisse nach [Eley, 2012]

Für die ereignisdiskrete Simulation werden Softwarewerkzeuge verwendet, welche die auszuführenden Ereignisse in einer Zeitliste verwalten. Diese Zeitliste wird auch Ereignisliste, Ereignisverwalter oder Simulationsuhr genannt und ist aufsteigend nach der zugehörigen Simulationszeit der Ereignisse sortiert. Sobald ein Ereignis beendet ist, wird dieses aus der Zeitliste entfernt. Es entstehen während der Durchführung des Simulationslaufs neue dynamische Ereignisse, welche entsprechend ihrer Simulationszeit, in die Zeitliste einsortiert werden. Wenn die Zeitliste leer ist oder aber die Simulationsuhr das Ende des Simulationszeitraums erreicht hat, endet der Simulationslauf. Die Softwarewerkzeuge enthalten neben der Zeitliste noch weitere Elemente, auf die im folgenden näher eingegangen wird. Es wird bei diesen Elementen zwischen *materiellen* und *immateriellen* Elementen unterschieden. Bei den materiellen Elementen handelt es sich um im realen System physisch vorhandene Elemente, wie z.B. Maschinen oder andere physisch vorhandene Elemente des realen Systems [Eley, 2012]. Eley [2012] fasst die wichtigsten immateriellen Elemente wie folgt zusammen:

- Entity: Bei Entities handelt es sich um Objekte, die sich und ihren Standort während eines Simulationslaufs ändern können. Im deutschen Sprachraum werden sie auch *bewegliche* Elemente genannt. Sie dienen dazu um Objekte wie Werkstücke oder Transportbehälter, welche durch das System wandern, abzubilden. Entities werden vom Simulationsprogramm erzeugt und können auch von diesem vernichtet werden.
- Ressourcen: Entities nehmen Ressourcen innerhalb des Modells in Anspruch. Ressourcen werden unter anderem auch *unbewegliche* Objekte genannt. Beispiele für Ressourcen sind Montage- und Bearbeitungsstationen. Ressourcen sind während des ganzen Simulationslaufs vorhanden aber nicht durchgehend verfügbar, da sie belegt sein können.
- Attribute: Attribute stellen die Eigenschaften von Entities und Ressourcen dar, die ihr Verhalten festlegen. Ein beispielhaftes Attribut *belegt* mit dem Wert *true* sorgt dafür, dass die Ressource keine weiteren Entities mehr aufnimmt und diese somit in der Warteschlange verweilen.
- Methoden: Methoden werden mithilfe einer Programmiersprache programmiert. Es handelt sich um Prozeduren, die in den Simulationsablauf eingreifen und Attributwerte ändern oder unter bestimmten Voraussetzungen neue Entities erschaffen. Durch Ereignisse ausgelöste Zustandsänderungen können ebenfalls mit Methoden gesteuert werden.
- Variablen: In Variablen werden Informationen und Daten abgespeichert, die im weiteren Simulationslauf noch benötigt werden. Es können ganzzahlige Werte oder aber auch komplexe Datenstrukturen in Variablen gespeichert werden.

- **Zufallszahlen:** Zufallszahlen dienen dazu, um Zufallsprozesse zu erzeugen. Gerade in einem stochastischen System sind sie von großer Bedeutung, da sie gemäß vorgegebener statistischer Verteilungen erzeugt werden.

Die einzelnen Elemente werden innerhalb der Simulationswerkzeuge visuell durch veränderbare und animierte Symbole dargestellt, um dem Anwender ein besseres Verständnis über das Simulationsmodell zu vermitteln.

Im folgenden Abschnitt wird der Vorgang der Erzeugung von Zufallszahlen erklärt und einige beispielhafte Simulationswerkzeuge vorgestellt.

2.2.1 Zufallszahlen und Simulationswerkzeuge

Es ist anzumerken, dass in Plant Simulation wirklicher Zufall kaum umgesetzt werden kann und dies auch nicht erwünscht wird [Eley, 2012]. Es kommen Pseudozufallszahlenfolgen zum Einsatz, welche in ihren Eigenschaften kaum von echten Zufallszahlenfolgen zu unterscheiden sind. Sie zeichnen sich dadurch aus, dass sie stets periodisch sind. Das bedeutet, dass sobald eine Zufallszahl zum zweiten Mal ermittelt wird, stimmen alle folgenden Zahlen mit den vorher berechneten überein [Werners, 2013]. Eine gute Pseudozufallszahlenfolge zeichnet sich somit durch eine lange Periodenlänge aus, die so lang ist, dass die Periodizität nicht beobachtbar ist. Die Reproduzierbarkeit von Zufallszahlen ist eine gewünschte Eigenschaft von Pseudozufallszahlen, was zwar der Zufälligkeit widerspricht, aber das Testen von Algorithmen ermöglicht. Bei einer wirklich zufälligen Zahlenfolge wäre nicht ersichtlich, ob eine Abweichung durch die Zahlenfolge oder die Implementierung des Algorithmus zustande kommt [Jetzke, 2007]. Bei der Berechnung eines Pseudozufallszahlenstroms wird der Startwert auch *Seedwert* genannt. Der Pseudozufallszahlenstrom ist bei einem Neustart des Simulationsmodells stets der gleiche, sofern sich dieser Seedwert nicht ändert. Erst bei einem anderen Seedwert entsteht ein neuer Pseudozufallszahlenstrom.

Tabelle 2.2 listet die gängigsten Verteilungsfunktionen in SimTalk auf, wobei in dieser Arbeit lediglich die Gleich- und Normalverteilung eingesetzt wird.

Verteilung	Funktion
Gleichverteilung	<code>z_gleich(s,a,b)</code>
Dreiecksverteilung	<code>triangle(s,a,b,m)</code>
Normalverteilung	<code>z_normal(s,μ,σ)</code>
neg. Exponentialverteilung	<code>z_negexp(s,λ)</code>
Binomialverteilung	<code>z_gleich(s,n,p)</code>

Tabelle 2.2: Verteilungsfunktionen in Simtalk nach [Eley, 2012]

Abbildung 2.3 zeigt die Vielfalt an etablierten Simulationswerkzeuge der vergangenen

Jahre im Bereich der Produktion und Logistik. Die Werkzeuge sind nach ihrem Spezialisierungsgrad sortiert. Die obere Hälfte der Abbildung zeigt Werkzeuge, die nur für spezielle Anwendungsbereiche geeignet sind, während die Werkzeuge in dem unteren Bereich allgemeine Programmiersprachen darstellen.

Simulationsprachen bilden einen speziellen Bereich der Programmiersprachen dar und sind so genannte *formale Sprachen*. Formale Sprachen bieten sich für den Einsatz in der Simulation an, da sie eine Reihe vorgefertigter Elemente und Funktionen für die Modellformulierung anbieten. Zu erwähnende Simulationsprachen sind unter anderem: *SIMSCRIPT*, *Simula*, *SimTalk* und *SLAM*. In [VDI-Richtlinie 3633, 2013] werden Programmiersprachen in anweisungsorientierte, objektorientierte und blockorientierte unterteilt. Nur objekt- und blockorientierte Programmiersprachen können als Simulationsprachen verstanden werden. Das im Rahmen dieser Bachelorarbeit eingesetzte Werkzeug „Tecnomatix Plant Simulation“ ist in der Mitte der Abbildung 2.3 angesiedelt und stellt somit einen Kompromiss beider Extrema dar [Eley, 2012].

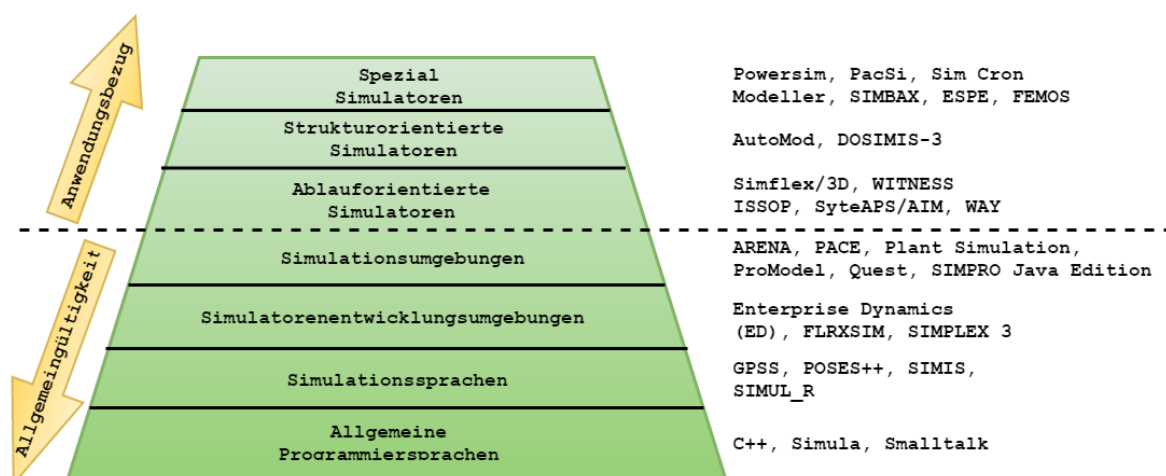


Abbildung 2.3: Beispielfhafte Simulationswerkzeugen für Produktion und Logistik nach [Eley, 2012]

Plant Simulation ist eines der meistverwendeten Simulationswerkzeuge in der Industrie. Das Haupteinsatzgebiet der Simulation ist der Betrieb von Fabrikssystemen, sowie deren Planung. Bei der Simulation von Materialflüssen zeichnet sich der Trend, dass das zu Planung erstellte Modell nach einigen Modellanpassungen ebenfalls zur Materialflusststeuerung genutzt wird [Pawellek, 2008].

Abbildung 2.4 veranschaulicht die Entwicklung der Einsatzgebiete der Simulation. Hierbei wird ersichtlich, dass der Trend immer weiter zum Einsatz der Simulation sowohl in der Materialflussplanung, -steuerung, -gestaltung und -regelung geht.

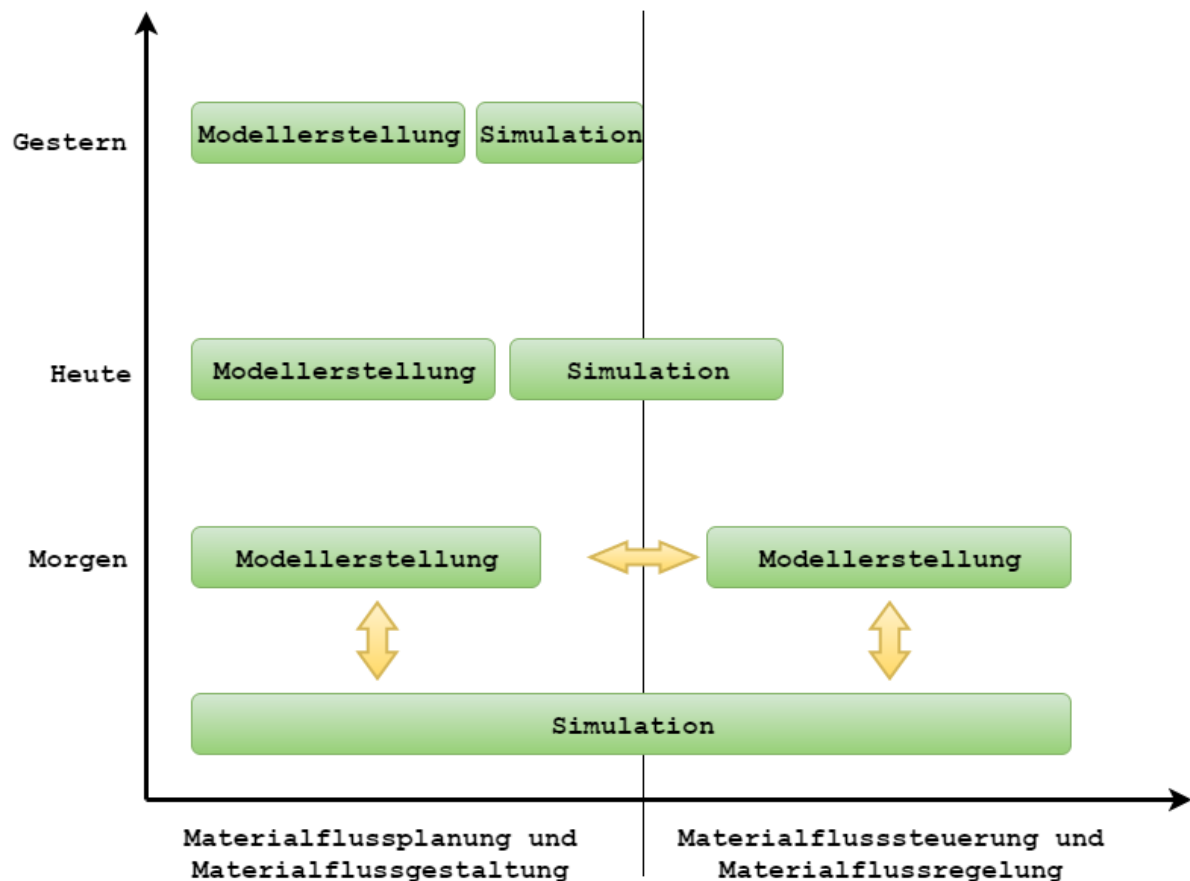


Abbildung 2.4: Einsatzgebiete der Simulation nach [Pawellek, 2008]

Plant Simulation nutzt in seiner Programmierschnittstelle die Simulationssprache *SimTalk*, um auch erfahrene Simulationsanwender anzusprechen. Hierbei handelt es sich um eine objektorientierte Sprache, die die Modifizierung der Grundbausteine und die Realisierung von komplexen und realitätsgetreuen Systemmodellen ermöglicht [Bangsow, 2011]. Die Funktionsweise, Benutzeroberfläche und die gängigsten Elemente von Plant Simulation werden im folgenden Abschnitt erklärt.

2.3 Tecnomatix Plant Simulation

Bei Plant Simulation handelt es sich um ein kommerzielles Simulationswerkzeug zur ereignisdiskreten Simulation. Die Funktionsweise dieser Werkzeuge wurde bereits in den Abschnitten 2.1 und 2.2 näher erläutert, weshalb an dieser Stelle nur auf die Besonderheiten von Plant Simulation eingegangen wird.

Gerade die einfache Zugänglichkeit und die leichte Bedienung von Plant Simulation machen die Software zu dem führenden Simulationswerkzeug für Materialflusssimulationen in der Industrie und wird in fast allen Wirtschafts- und Industriezweigen eingesetzt. Besonders die grafische Darstellung einzelner Elemente des Simulationsprojekts sorgt für eine leichte

Verständlichkeit. Dieses Modellprinzip wird *bausteinbezogene Modellierung* genannt. Die bereits im Vorfeld der Simulationsstudie ermittelten Informationen werden im Anschluss an Plant Simulation übertragen und entsprechend abgebildet, wobei zu erwähnen ist, dass je komplexer die Simulationsstudie ist, desto komplexer die Umsetzung in Plant Simulation ist [Bangsow, 2011, 2008; EDAG Production Solutions GmbH & Co. KG, 2015].

Die große Bedeutung wird zudem deutlich, wenn man sich die Vorteile von Plant Simulation betrachtet. So wirbt der Hersteller Siemens unter anderem damit, dass Einsparungen bei Erstinvestitionen, die Steigerung der Produktivität und die Systemoptimierung für geringeren Energieverbrauch mit Plant Simulation möglich ist. Tabelle 2.3 listet die Vorteile von Plant Simulation auf.

Verbesserung	Ausprägung
Einsparungen bei Erstinvestitionen	3-6%
Steigerung der Produktivität bestehender Systeme	20%
Senkung der Kosten für Neusysteme	20%
Verkleinerung des Lagerbestands	um bis zu 60%
Verkürzung der Durchsatzzeiten	um bis zu 50%

Tabelle 2.3: Vorteile von Plant Simulation nach [Siemens Industry Software GmbH, 2015]

Die Benutzeroberfläche von Plant Simulation wird in Abbildung 2.5 dargestellt, wobei die Hauptelemente gesondert geschildert sind.

Die Simulation läuft zum größten Teil in dieser Ansicht ab und die meisten Funktionen sind direkt aufrufbar. Im linken Teil der Benutzeroberfläche ist die *Klassenbibliothek* positioniert, welche alle für die Simulation nutzbaren Elemente beinhaltet. Das Anlegen eigener Ordner, für einen schnelleren Zugriff auf einzelne Elemente, das Duplizieren und Ableiten von Elementen sowie das Anlegen von Netzwerken ist ebenfalls über die Klassenbibliothek möglich [Eley, 2012; Bangsow, 2011].

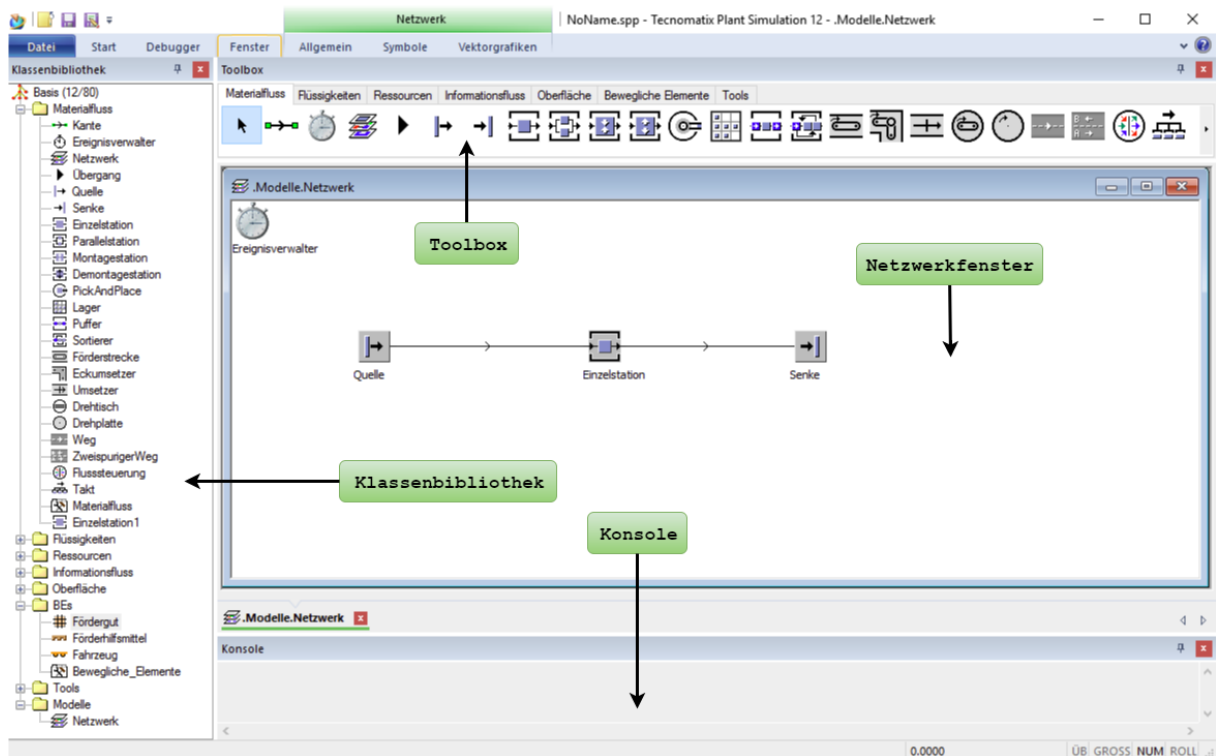


Abbildung 2.5: Benutzeroberfläche von Plant Simulation

Die *Konsole* gibt Informationen, wie z.B. Fehlermeldungen aus, die im Verlauf des Simulationslaufs entstehen. Gerade bei der Fehlersuche in komplexen Simulationssystemen ist die Konsole ein unerlässliches Hilfswerkzeug [Eley, 2012; Siemens PLM Software, 2015]. Die *Toolbox* dient dem Schnelzugriff auf Elemente aus der Klassenbibliothek. Eigene Elemente können zur Toolbox hinzugefügt werden, um die eigene Produktivität zu erhöhen und die Software auf die eigenen Bedürfnisse anzupassen [Eley, 2012; Siemens PLM Software, 2015].

Das *Netzwerkfenster* stellt die Arbeitsfläche für die Simulation dar. Hier werden alle Elemente von Plant Simulation abgelegt und mit Hilfe von Kanten miteinander verbunden. Es erlaubt ein einfaches Experimentieren mit dem Simulationsmodell während es pausiert ist und gibt beim Simulationslauf Fehlermeldungen aus. Die Elemente werden mittels *Drag&Drop* aus der Klassenbibliothek in das Netzwerkfenster gezogen und abgelegt [Eley, 2012; Siemens PLM Software, 2015].

Der *Ereignisverwalter* ist ein Werkzeug zur Manipulation der Simulationsuhr. Mit Hilfe dieses Werkzeugs kann die Simulation gestartet, gestoppt und zurückgesetzt werden. Die Materialbewegungen werden beim Start der Simulation grafisch angezeigt. In diesem Zusammenhang werden die Fördergüter in Plant Simulation BE (bewegliche Einheit) genannt. BEs können keine Teile aufnehmen. Außerdem ermöglicht der Ereignisverwalter eine Anpassung der Simulationsgeschwindigkeit beziehungsweise die Simulationszeit. Dieses Hilfsmittel ist besonders für Simulationen unerlässlich, die ein Modell betrachten, welches

über eine große Zeitspanne in der Realität abläuft. Sehr schnelle Vorgänge können durch das Verlangsamen der Simulationsgeschwindigkeit ebenfalls besser untersucht werden. Mit dem Kontrollhäkchen *Echtzeit*, kann die Simulationsgeschwindigkeit mit Hilfe eines Multiplikators exakt an die in der Realität vergehende Zeit angepasst werden. Neben der Manipulation der Schnelligkeit ist zudem die Option der *Einzelstschritt-Simulation* vorhanden. Jeder klick auf diesen Button führt exakt einen Schritt innerhalb der Simulation aus. Somit wird immer das nachfolgende Ereignis innerhalb des Modells ausgeführt. Durch den Zurücksetzen Button wird das Simulationsmodell wieder in den Anfangszustand versetzt, wobei weitere Methoden erforderlich sein können, um Variablen und Tabellenwerte zurückzusetzen [Eley, 2012; Siemens PLM Software, 2015; Bangsow, 2011]. Der Ereignisverwalter ist in Abbildung 2.6 abgebildet.

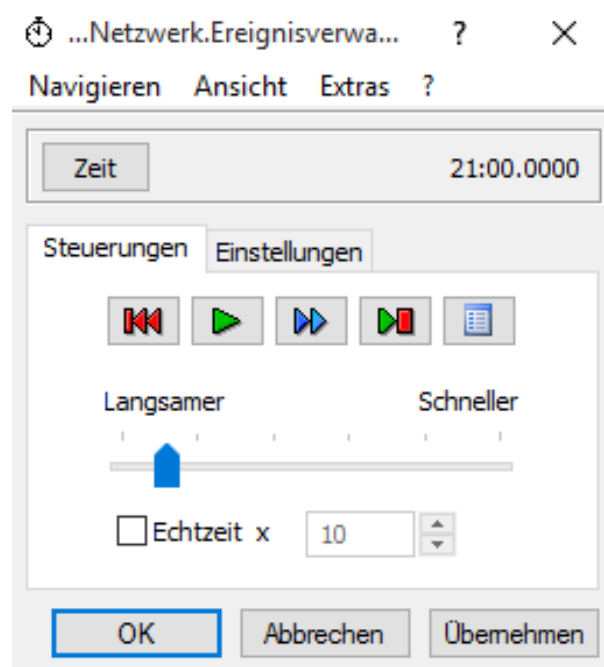


Abbildung 2.6: Der Ereignisverwalter in Plant Simulation

Mit einem Doppelklick auf die einzelnen Elemente innerhalb des Netzwerkfensters gelangt man in die *Objekteigenschaften* des jeweiligen Elements. In Abbildung 2.7 sind die Objekteigenschaften einer beispielhaften Einzelstation abgebildet. Die Objekteigenschaften erlauben es dem Anwender, die benutzten Objekte realitätsgetreu abzubilden. Es können Werte wie die Bearbeitungszeit, Rüstzeit, Erholzeit oder Zykluszeit modifiziert und mit stochastischen Funktionen und Verteilungen versehen werden. Da es in der Realität immer wieder zu Störungen im Produktions- und Lieferprozess kommen kann, können auch Störungen simuliert werden. Die Bearbeitungszeit legt in den Objekteigenschaften eines Elements fest, wie lange sich ein BE auf der jeweiligen Station aufhält, bis es an den Nachfolger übergeben wird. Die Rüstzeit stellt in diesem Zusammenhang die Zeit dar, welche benötigt wird, um ein Arbeitssystem (z.B. Maschine, Anlage, Fertigungsstelle)

für einen bestimmten Vorgang einzurichten. Die Erholzeit gibt die Zeit an, in der keine weiteren BE die Station betreten können, nachdem ein BE ausgetreten ist. Mit Hilfe der Zykluszeit wird festgelegt, in welchem Zeitraster die BE eintreten können [Siemens PLM Software, 2015].

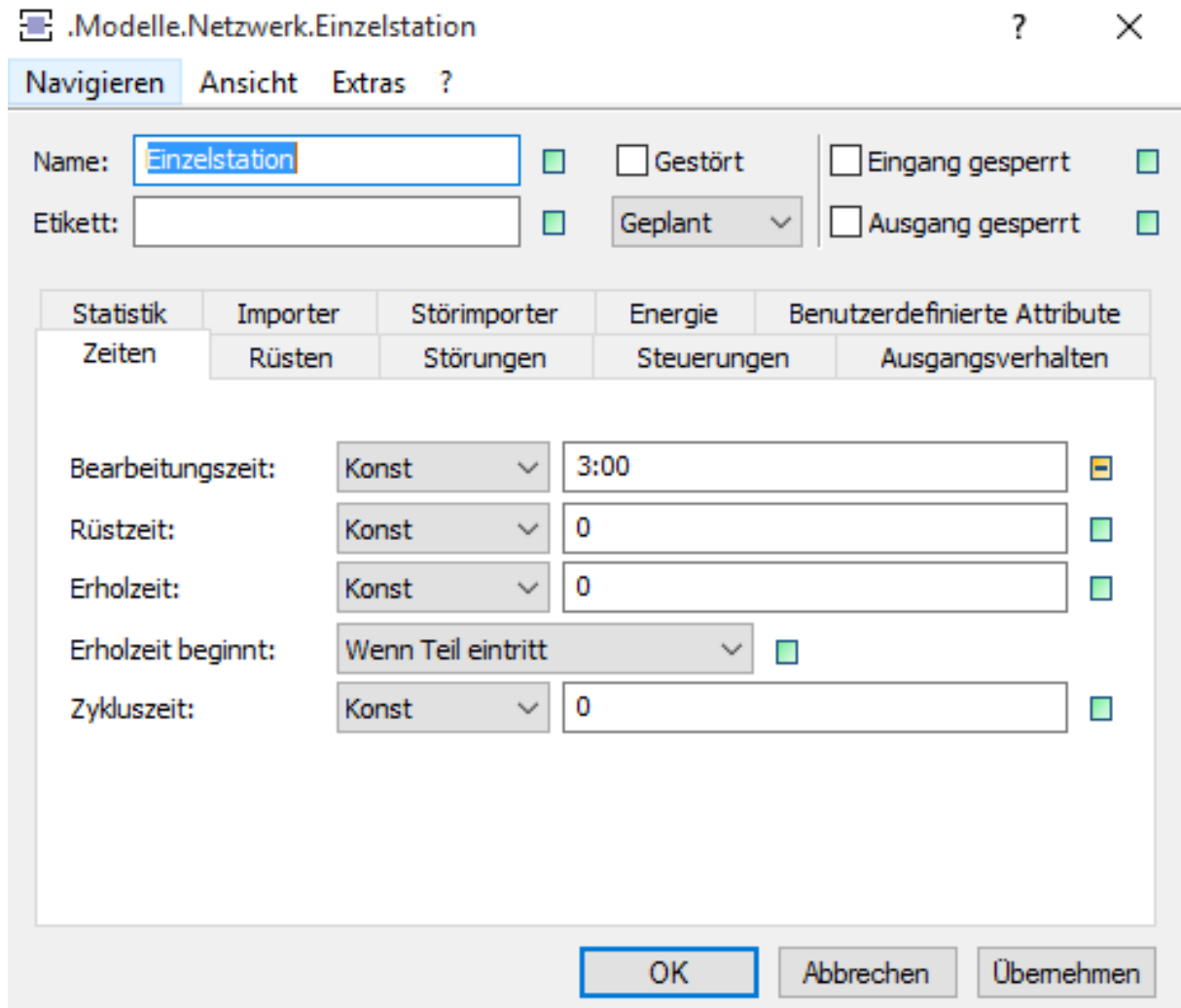


Abbildung 2.7: Objekteigenschaften einer Einzelstation in Plant Simulation

Diese Eigenschaften der Objekte werden auch *Attribute* genannt (vgl. Abschnitt 2.2), welche mit unterschiedlichen Datentypen beschrieben werden.

Im folgenden Abschnitt werden die verschiedenen wählbaren Datentypen und Elemente erklärt.

2.3.1 Datentypen und Elemente in Plant Simulation




Eine Auswahl von verschiedenen, im Rahmen dieser Arbeit verwendeten, Datentypen ist in Tabelle 2.4 zu sehen. Diese Datentypen dienen der Deklaration von Variablen und Attributen der Elemente und stimmen mit denen aus anderen Programmiersprachen überein.

Es ist außerdem möglich Datentypen in beschränktem Maße ineinander umzuwandeln [Bangsow, 2011].

Typ	Bezeichnung	Wertebereich	Mögliche Operationen
Integer	Ganze Zahlen	-2147483648 bis 2147483647	+, -, *, <, >, =, Division mit Rest und Modulo
String	Zeichenkette	a bis z, A bis Z, Sonderzeichen, Ziffern 0 bis 9	<, >, =
Boolean	Logische Werte	true / false	NOT, AND, XOR, NOR, NAND, OR, =, ≠
Date und DateTime	Datumsangaben	01.01.1970 bis 31.12.9999	Eingeschränkte Addition und Subtraktion

Tabelle 2.4: Beispielhafte Datentypen in SimTalk und Plant Simulation nach [Eley, 2012; Siemens PLM Software, 2015]

In Tabelle 2.5 werden die Bausteine aufgeführt, die in dem Hauptteil dieser Arbeit verwendet worden sind [Siemens PLM Software, 2015].

Name	Funktion	Symbol
Quelle	Eine <i>Quelle</i> produziert BEs. Mit der Quelle können verschiedene Teiletypen hintereinander oder in gemischter Reihenfolge produziert werden. Eine Quelle hat <i>keine</i> Bearbeitungszeit und ihre Kapazität beträgt <i>1</i> .	
Senke	Die <i>Senke</i> entfernt BEs aus dem Simulationsmodell, nachdem diese alle Bearbeitungsstationen durchlaufen haben. Somit gibt die Senke im Gegensatz zu anderen Elementen die bearbeiteten BEs <i>nicht</i> an den Nachfolger ab. In der Realität stellt sie den Warenausgang oder den Versand einer Fabrik dar.	
BE	<i>BEs</i> repräsentieren Fördergüter innerhalb des Simulationsmodells. BEs können produziert, bearbeitet und transportiert werden. BEs besitzen keine Ladekapazität und sind nicht am Transport anderer Teile beteiligt.	

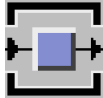



Einzelstation	Eine <i>Einzelstation</i> besitzt einen Arbeitsplatz, um ankommende Teile zu bearbeiten und im Anschluss (nach Ablauf der Rüst- und Bearbeitungszeit) an die nachfolgende Station weiterzugeben. Sie ist ein Materialflussbaustein, die eine Maschine mit der Kapazität 1 abbildet.	
Kante	Eine <i>Kante</i> stellt eine Verbindung zwischen zwei Elementen innerhalb des gleichen Netzwerkes her. Die Richtung der Verbindung wird über die Pfeilspitze auf der Verbindungslinie verdeutlicht.	
Methode	Das Informationsflussobjekt <i>Methode</i> dient der Erstellung von Steuerungen, welche dann von anderen Objekten aufgerufen werden können. Methoden werden unter Verwendung der Programmiersprache <i>SimTalk</i> erstellt. Methoden können automatisch bei dem Zurücksetzen (<i>reset-Methode</i>) und beim Start des Simulationsmodells (<i>init-Methode</i>) ausgeführt werden.	
Tabelle	Eine <i>Tabelle</i> stellt eine Liste mit mehreren Spalten dar. Mit Hilfe von Methoden können Werte in Tabellen geschrieben und aus Tabellen gelesen werden.	

Tabelle 2.5: Elemente in Plant Simulation nach [Siemens PLM Software, 2015]

In dem in dieser Arbeit zu erstellenden Simulationsmodell handelt es sich um einen Materialfluss innerhalb der Weißwarenindustrie. Dieser Materialfluss findet in einem Netzwerk zwischen einem First-Tier Zulieferer und dem OEM (Original Equipment Manufacturer) statt. Mit Hilfe der Simulation sollen innerhalb dieses Zulieferernetzwerks mit Hilfe des Data Farmings Daten erzeugt werden.

Um ein Entsprechendes Simulationsmodell erstellen zu können, müssen zunächst die Grundlagen von SCs erklärt werden, da sich die vorliegende Arbeit mit der Supply Chain Simulation beschäftigt.

2.4 Supply-Chains

Innerhalb einer SC konzentrieren sich die Maßnahmen vorwiegend auf nachhaltige Prozessoptimierungen, die Standardisierung von Produktionssystemen sowie die Arbeitsteilung zwischen OEM und Zulieferern, da sich der Endkunde sich bei seinem Kauf nur am Gesamtwert des Gutes orientiert. Durch die neue Ordnung der Arbeitsteilung wird zunehmend mehr Verantwortung auf die Zulieferer übertragen, damit sich der OEM weitestgehend auf seine eigenen Kernkompetenzen fokussieren kann. Jede wertschöpfende Leistung und jedes Teil, was der Kunde *nicht* mit der Marke des OEM verbindet, steht Lieferanten und

Logistikdienstleistern zur Disposition [Gehr u. Hellingrath, 2007; Heiserich et al., 2011]. Heiserich et al. [2011] nennen hierbei drei Gründe für eine unternehmensübergreifende Arbeitsteilung. Zum einen liegt der Grund bei unzureichendem *Know-How* und mangelnder *Qualität*, da ein einzelnes Unternehmen kaum noch in der Lage ist, alles selbst herzustellen. Zum anderen sind wie bereits erwähnt die *Kosten* der Unternehmung ein großer Faktor, der bei einer großen Anzahl von benötigten Technologien und Prozessen eine Wirtschaftlichkeit ausschließt. Als letzten Grund nennt Heiserich et al. [2011] die zu geringe *Flexibilität* eines allein agierenden Unternehmens, die nötig ist um schnell auf schwankende Kundenwünsche zu reagieren.

Eine SC erstreckt sich über mehrere Wertschöpfungsglieder. Sie reicht von der *Source of Supply* bis zum *Point of Consumption* [Werner, 2014]. Durch die Nutzung kostengünstiger Standorte und der Bedarfszusammenführung können Mengendegressionseffekte erreicht werden, die zu Kosteneinsparungen für alle beteiligten Parteien führen. Trotz gewachsener Strukturen können selbst die an erster Stelle stehenden Lieferanten nicht allen Anforderungen durch den OEM nachkommen und es folgt eine weitere Verlagerung auf nachfolgende Lieferkettenpartner. Dieser Prozess führt zu immer tiefer gestaffelten Lieferketten. Ein bekanntes Beispiel aus der Automobilindustrie, welches enorme Ausmaß diese ständige Verlagerung annehmen kann, sind die Lederteile für die Türinnenverkleidung der Mercedes-Benz S-Klasse, mit einer achtstufigen Lieferkette [Gehr u. Hellingrath, 2007]. Der stetig wachsende Druck zu hoher Lieferflexibilität bei höchster Qualität und Wirtschaftlichkeit kann nicht mehr von den Unternehmen alleine bewältigt werden. Netzwerke treten, statt singulärer Kunden-Lieferanten-Beziehungen, immer mehr in den Vordergrund. Die OEM drängen immer mehr auf eine Transformation von Lieferketten zu Netzwerken. In vielen Fällen wird auf zwei Lieferanten für die gleichen Teile gesetzt, um die Lieferflexibilität im Falle eines Lieferantenausfalls oder fehlenden Kapazitäten garantieren zu können. Die kontraktierten Logistikdienstleister sind keine reinen Transporteure mehr wie einst, sondern haben sich zu flexiblen Dienstleistern mit einem großen Spektrum an ergänzenden Aufgaben entwickelt. Ihre Einbeziehung ist derweil unumgänglich geworden, bringt aber auch neue Herausforderungen mit sich [Gehr u. Hellingrath, 2007].

Die Zusammenarbeit in Zulieferernetzwerken erfolgt heutzutage fast ausschließlich nach einem hierarchischen Prinzip. Der dominierende Akteur ist der OEM [Gehr u. Hellingrath, 2007]. Die nachfolgende Abbildung 2.8 zeigt die klassische Hierarchie in Zulieferernetzwerken.

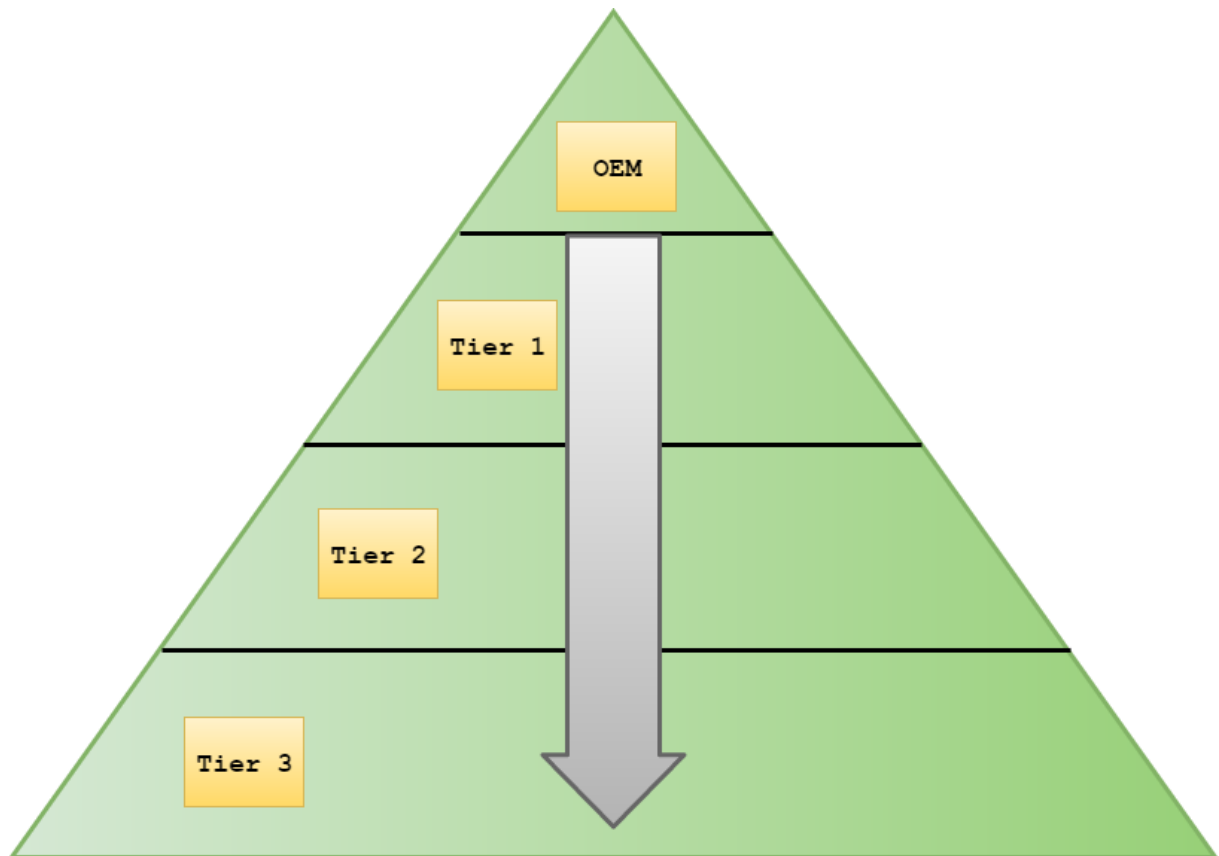


Abbildung 2.8: Hierarchie einer Supply-Chain nach [Gehr u. Hellingrath, 2007]

Der *OEM* hat die größten wirtschaftlichen Möglichkeiten und somit stehen diesem eine leistungsfähige Prozess- und IT-Infrastruktur zu Verfügung. Bei den IT-Lösungen handelt es sich noch überwiegend um Individuallösungen, jedoch hat die Einführung von Standardsoftware begonnen (z.B. SAP Logistik-Module). Diese Standardsoftware deckt derzeit noch nicht alle Umfänge der Produktion und Logistik ab. Das größte Interesse der OEM innerhalb des Zulieferernetzwerkes ist die rechtzeitige Erkennung potenzieller Lieferengpässe sowie die Gewährleistung der Transparenz über das Netzwerk. Das primäre Ziel ist die durchgängige Versorgungssicherheit der Montagelinien mit Teilen. Ein Performance-Monitoring ermöglicht zudem Erkenntnisse über die Kostenrelevanz der Netzwerks, im Hinblick auf eine Kostenverbesserung. Der OEM ist der Erzeuger von Enderzeugnissen, welche für den Endkunden bestimmt sind [Heiserich et al., 2011].

Der zweite Akteur einer SC ist der *Logistikdienstleister*, der ein Bindeglied im Netzwerk darstellt. Er stellt flexible teure Transportkapazitäten zur Verfügung und übernimmt noch ein großes Spektrum ein weiteren steuernden Aufgaben. Die rechtzeitige Kenntnis über Bedarfsprofile und geplante Lieferungen ist für sie von großem Interesse, da diese zur optimalen Planung der Transport-, Lager- und Logistikkapazitäten dienen. Jetzke [2007] nennt einige Anforderungen, die an Lager gestellt werden. Zum einen soll mit Hilfe von Lagern die Schwankung und Asynchronität von Angebot und Nachfrage ausgeglichen

werden. Außerdem helfen Lager bei der Ermöglichung von Mengeneffekten, da Waren in größeren Mengen eingekauft werden können, um niedrigere Preise zu erzielen.

Der *First-Tier Lieferant* ist der wichtigste Vertreter der Zulieferer und stellt einen System- bzw. Modulzulieferer dar. Er trägt die Verantwortung über die termin- und mengengerechte Belieferung des OEM. Diese Versorgungssicherheit ist meist vertraglich geregelt. Ihr Ziel ist ebenfalls die Sicherstellung einer Transparenz innerhalb der SC sowie die frühzeitige Erhaltung von Bedarfskapazitätenprofile. Diese Kenntnis ermöglicht ein proaktives Handeln und die rechtzeitige Erkennung von Lieferengpässen. Die optimale Auslastung ihrer kritischen Ressourcen (auch Engpassressourcen) im Sinne der Versorgungssicherheit und Kostenrelevanz ist ein weiteres Ziel dieses Akteurs.

N-Tier Lieferanten sind Zulieferer der zweiten oder einer nachfolgenden Stufe. Bei diesen Zulieferern handelt es sich meistens um kleine und mittelständische Unternehmen, welche entsprechend ihrer wirtschaftlichen Möglichkeiten nur selten über die notwendigen IT-Strukturen verfügen, um eine datentechnische Anbindung an den 1st-Tier-Zulieferer oder den OEM zu haben. Aus diesem Grund ist ein Planungsfenster für diesen Partner eine SC von hoher Bedeutung. Ohne dieses Planungsfenster ist die Teilnahme am Netzwerk nicht möglich [Gehr u. Hellingrath, 2007; Waters, 2009].

Es handelt sich somit um ein heterogenes Interessenbündel von Akteuren, weshalb Spannungsverhältnisse nicht verwunderlich sind. Zum einen erhoffen sich die teilnehmenden Akteure eine erhöhte Wettbewerbsfähigkeit, zum anderen streben sie nach Autonomie. Im Rahmen des SCM (Supply Chain Management) gilt es, diesen Balanceakt so auszuführen, dass alle teilnehmenden Partner zufrieden sind. Das SCM ist erforderlich, da keine leitende und übergeordnete Instanz innerhalb einer SC existiert. Weisungen, Pläne und Programme sind innerhalb einer SC schwächer ausgeprägt, als innerhalb eines einzelwirtschaftlichen Unternehmens. Zudem muss ein stetiger Konsens vorherrschen, um die langfristige Bestehen des Netzwerkes zu ermöglichen [Werner, 2014].

Jetzke [2007] definiert den Begriff der SC wie folgt:

„Die Lieferkette ist die geordnete Folge von Tätigkeiten, die zur Erfüllung eines Kundenwunsches notwendig sind. Der Anfang der Güterflusses in einer Lieferkette soll dort sein, wo die einzusetzenden Produkte austauschbar sind. Das Ende dort, wo der weitere Verbleib keine Rolle spielt.“

SCs können in zwei Typen strukturiert werden. Zum einen gibt es die *hierarchisch pyramidalen Supply-Chains* und zum anderen die *polyzentrischen Supply-Chains*. Bei der ersteren Art handelt es sich um eine SC mit einer strategisch relevanten Unternehmung im Mittelpunkt. Alle beteiligten Akteure der Lieferkette richten ihre Aktivitäten nach diesem Unternehmen. Das dominierende Unternehmen beherrscht die SC durch ihre Größe, finanziellen Mittel oder ihr Wissenspotential. Die Partner dieser Art von Lieferketten orientieren sich demnach an der Marktmacht des dominierenden Unternehmens, welches seine Partner durch langfristige Verträge an sich bindet. Im Gegenzug dazu herrschen bei

polyzentrischen SCs homogene wechselseitige Beziehungen vor. Führung und Dominanz innerhalb dieser SC-Art werden durch regelmäßige Verhandlungen oft neu verteilt und die Entscheidungskompetenzen sowie die Koordinationsaufgaben sind gleichmäßig auf die Partner verteilt [Werner, 2014].

Die Abbildung 2.9 verdeutlicht die Beziehungen zwischen Lieferanten, OEM und Kunden. Es ist zu erwähnen, dass in dieser Arbeit lediglich die Verbindung zwischen den First-Tier Lieferanten und dem OEM betrachtet werden und die vor- und nachgelagerten Stufen einer SC nicht im Fokus stehen [Waters, 2009].

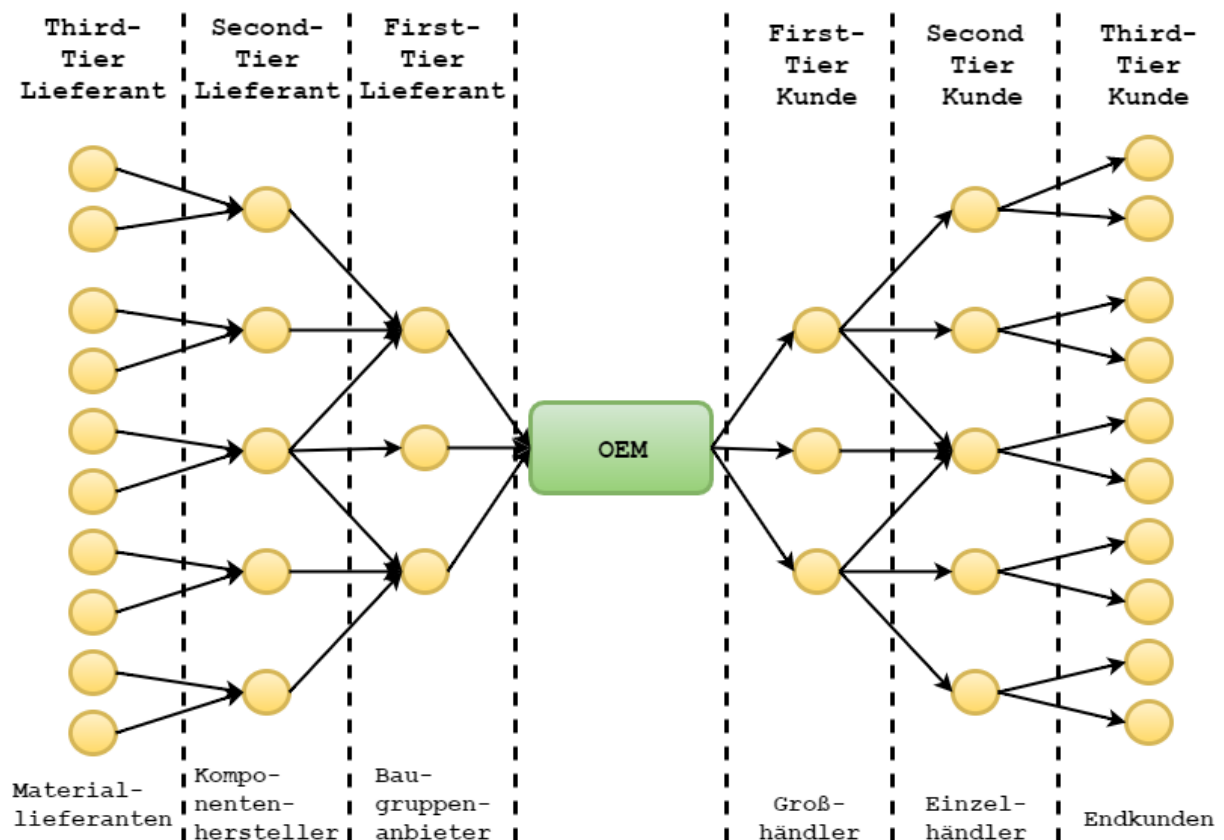


Abbildung 2.9: Supply-Chain eines Endproduzenten nach [Waters, 2009]

In Zusammenhang mit dem Begriff der SC stößt man oft auf den Begriff *Materialfluss*. Rabe [2003] definiert diesen Begriff wie folgt:

Materialfluss ist die Verkettung aller Vorgänge beim Gewinnen, Be- und Verarbeiten, sowie bei der Verteilung von stofflichen Gütern innerhalb festgelegter Bereiche. Der Begriff Materialfluss schließt mehrere Teilfunktionen ein und bezieht sich auf alle Materialbewegungen innerhalb oder zwischen umgrenzten Betriebseinheiten in ihrer räumlichen, zeitlichen und organisatorischen Verknüpfung mit den betrieblichen Aufgaben.

Die SC stellt eine wichtige Prozesskette für Unternehmen dar, da an der Schnittstelle

zum Kunden die Leistungsfähigkeit des Unternehmens demonstriert wird. Daher ist die SC ein Zentrales Element der Kernprozesse eines Unternehmens. Der Fokus einer SC ist die bestmögliche Zufriedenstellung des Kunden, weshalb die Verbesserung dessen eine sehr hohe Bedeutung besitzt. Die SC-Prozesse Strategie, Prozesse, Kundenanforderungen Einrichtungen, Kennzahlen, Datenverarbeitungssysteme sowie Organisation stehen im Mittelpunkt des SCM [Becker, 2008]. Müller [2005] beschreibt den Begriff der SCM wie folgt:

Als SCM kann die Planung, Steuerung und Kontrolle des gesamten Material- und Dienstleistungsflusses, einschließlich der damit verbundene Informations- und Geldflüsse, innerhalb eines Netzwerks von Unternehmen und deren Bereichen verstanden werden, die im Rahmen von aufeinanderfolgenden Stufen der Wertschöpfungskette an der Entwicklung, Erstellung und Verwertung von Sachgütern und/oder Dienstleistungen partnerschaftlich zusammenarbeiten, um Effektivitäts- und Effizienzsteigerungen zu erreichen.

Durch die Entstehung vieler Wertschöpfungsketten zur Ausnutzung von Optimierungseffekten, haben sich die Konkurrenzbeziehungen am Markt verändert. Es konkurrieren mittlerweile nicht mehr einzelne Unternehmen miteinander, sondern ganze SCs. Aus diesem Grund ist es auch erwiesen, dass ein Unternehmen nur dann erfolgreich am Markt konkurrieren kann, wenn es eine bessere Kooperation mit seinen Partnern aufweist, als die Konkurrenz [Arndt, 2015]. Somit hat sich das SCM von einem Randthema zu einer zentralen Aufgabe entwickelt. Unternehmen wie *smart*, *Zara*, *Microsoft* und *Herman Miller* konnten durch den Einsatz innovativer SCM-Konzepte ihre Effektivität sowie ihre Effizienz verbessern. Alle Unternehmen erzielten Einsparungen, bessere Lieferzeiten sowie eine Verbesserung des Lieferservice [Heidtmann, 2008].

Nachdem die Grundlagen von SCs aufbereitet worden sind, wird im nächsten Abschnitt das Data Farming erläutert.

2.5 Data Farming

Im Jahr 1998 hat General Charles Krulak, der damalige Kommandant der U.S. Marine Corps, die inhärente Asymmetrie des Krieges erkannt und behauptet, dass viele der existierenden Kampf- und Simulationsmodelle nicht den Bedingungen der Realität entsprechen [Horne u. Meyer, 2004]. Diese Realisation führte dazu, dass man versuchte die stetig voranschreitenden Kapazitäten von EDV-Systemen derart zu nutzen, dass neue Daten generiert und neues Wissen über Abstraktionen dieser Daten gewonnen werden konnte [Horne u. Meyer, 2004; NATO Science and Technology Organization, 2014].

Die Abstraktion der Daten war aufgrund der Komplexität auf dem Schlachtfeld für die Beantwortung vieler militärischer Fragen von Bedeutung. In diesem Sinne hat das Defense Science Board dem U.S. Verteidigungsministerium fundamentale Veränderungen bezüglich

der Simulationsmethoden vorgeschlagen.

Die neu vorgeschlagene Komponente trug den Namen *Data Farming* [Horne u. Brandstein, 1998] und wurde erstmals 1999 auf der *Winter Simulation Conference* in Phoenix (USA) vorgestellt. Laut Horne u. Brandstein [1998] liegt die Motivation Data-Farming-Techniken zu nutzen nicht in den wissenschaftlichen Fortschritten, sondern in der Tatsache, dass die Anzahl der unterschiedlichen Möglichkeiten, die auf dem Schlachtfeld eintreten können, unendlich groß ist. Aus diesem Grund ist es von Interesse, so viele Datenbeispiele und -variationen wie möglich zu betrachten um ein besseres Verständnis über die erzeugte Datenlandschaft zu erlangen [Horne u. Brandstein, 1998; Maryann Lawlor, 2005; Barry u. Koehler, 2004].

Der Begriff des Data Farmings stammt von Bauern, welche ihre Felder *kultivieren*, um ein Ertragsmaximum zu erzielen. Sie manipulieren zudem ihre Umgebung durch Dünger und Mittel zur Schädlingsbekämpfung zu ihrem Vorteil. Ähnlich wie die Bauern, manipulieren Data Farmer ihre Simulationsmodelle zu ihrem Vorteil. Dies erlaubt es ihnen, das Verhalten des Simulationsmodell zu studieren und Daten aus diesen zu *ernten* [Sanchez, 2014; Rabe u. Scheidler, 2015].

Data Farming ermöglicht die Untersuchung massiver Eingabebereiche, das Entdecken bedeutender Merkmale die im Rahmen der Komplexität der Simulation zustande kommen und das Erkennen von kausalen Beziehungen [Sanchez, 2014]. Selbst simple Modelle können einen großen Eingabegrößenbereich aufweisen. Für den Anwender können in dieser Hinsicht viele Parameterkombinationen interessant sein, jedoch kann sich die Einschätzung der Wichtigkeit von Parameterkombinationen im Vorfeld als schwierig bis unmöglich erweisen [Barry u. Koehler, 2004].

Aufzeichnungen historischer Ereignisse waren oftmals sehr detailreich, bildeten jedoch lediglich nur einen Punkt in der unendlich großen Datenlandschaft ab. Die hierfür genutzten Modelle waren sehr detailliert und komplex, so dass die Einrichtung und Parametrisierung mehrere Stunden in Anspruch nahm und das Ergebnis des Modells nur einen Punkt in der gesamten Datenlandschaft darstellte [Horne u. Brandstein, 1998; NATO Science and Technology Organization, 2014]. Sind keine historischen Daten vorhanden, bietet das Data Farming dennoch den Vorteil, dass man durch die Durchgeführten Simulationen eine Reihe von „What-If“-Szenarien effektiv und effizient betrachten kann [Sanchez, 2014; NATO Science and Technology Organization, 2014].

Horne u. Brandstein [1998] nutzte im Rahmen des Data Farmings ein militärisches Simulationsmodell (ISAAC), welches mehrere Male ausgeführt worden ist, um die Gesamtfläche der erzeugten Datenlandschaft zu vergrößern. Das Konzept des Data Farmings wurde in diesem Modell zum ersten Mal angewandt. Hierfür wurde eine Technik entwickelt, welche aus den folgenden vier Komponenten bestand:

- *Düngen*: Expertenbefragung, welche wichtigen Aspekte innerhalb eines Konfliktes wie ermittelt werden könnten, die in der Vergangenheit nur schlecht ermittelt worden sind
- *Bewirtschaften*: Anhand von Expertenwissen entdecken, was in einer gegebenen Situation wichtig sein könnte
- *Pflanzen*: Die erlangten Ideen, soweit wie möglich, in Modelle einpflanzen und die Ausführung des Modells für eine Anzahl an unterschiedlichen Möglichkeiten und Variablen, die von Interesse sind
- *Ernten*: Ernten des Datenoutputs aus dem genutzten Modell durch die Entwicklung von Techniken, die für das Verständnis wissenschaftlicher Daten genutzt werden

Somit kommt Horne u. Brandstein [1998] zu dem Schluss, dass das Data Farming aus dem Verlangen nach Antworten auf Fragen resultiert. Die Modelle müssen demnach fragenorientiert sein. Durch das *Farmen* eines großen Datenvolumens verspricht man sich einen Wissenszuwachs und ein verbessertes Verständnis über vorliegende Fragen und Probleme. Der Nutzen des Data Farmings liegt in der Fähigkeit Modelle schnell erstellen, ausführen und analysieren zu können, um die Ergebnisse zu interpretieren [Horne u. Brandstein, 1998; Barry u. Koehler, 2004].

Daher wird mit dem Data Farming keine Hypothese bestätigt oder widerlegt; die Datenlandschaft wird erkundet um Trends, Sattelpunkte, Scheitelpunkte und Ausreißer zu entdecken. Somit werden dem Anwender nicht nur die wahrscheinlichsten Ergebnisse vorgestellt, sondern gerade auch unwahrscheinliche, die unerwünschte Folgen haben können. Zur Erforschung der Fragen werden immer mehr Daten in den entsprechenden Datenbereichen erzeugt. Die Erzeugung dieser Daten kann unter anderem durch vermehrte Simulationsdurchläufe mit Parametervariation erfolgen. Beim Data Farming handelt es sich um einen Prozess, welcher schnell Antworten auf viele Fragen liefern, aber keinen Ausgang vorhersagen kann. Mittels High Performance Computing (HPC) beziehungsweise dem Einsatz von Hochleistungscomputer können Simulationsmodelle schnell erzeugt und deren Ergebnisse in kurzer Zeit gesammelt werden. Die erzeugten Modelle werden von den Anwendern mehrere Hunderttausend bis Millionen Male ausgeführt, um die Datenlandschaft mit Informationen zu befüllen [Horne u. Meyer, 2004; Maryann Lawlor, 2005; Barry u. Koehler, 2004; NATO Science and Technology Organization, 2014].

Maryann Lawlor [2005] zitiert Horne wie folgt:

„You get a dynamic combination because you can look at literally thousands, tens of thousand, hundreds of thousands, even millions of runs. You can vary the parameters, which are numerous because in today's uncertain world, you're up against so many different factors. You can't really predict anything, but if you look at enough possibilities, you can begin to understand.“

Im Anschluss an die Erkenntnisse aus dem Modell von Horne, hat der U.S. Kongress sein Interesse an einer Kombination von Data-Farming-Techniken mit Hochleistungsrechnern bekundet und *Project Albert* ins Leben gerufen. Das Ziel dieses Projekts war nicht die Vorhersage einer Antwort durch die wiederholte Ausführung unterschiedlicher Modelle, sondern Aufschluss und neues Wissen über die erzeugte Datenlandschaft zu erhalten. Mittels Data Farming können vielfältige Parametervariationen untersucht werden, um Überraschungen vorherzusehen und dem Anwender entsprechende Entscheidungshilfen zu bieten [Horne u. Meyer, 2004; Horne u. Schwierz, 2008; NATO Science and Technology Organization, 2014]. Sanchez [2014] erwähnt in diesem Zusammenhang, dass es wichtig ist, sich stets bewusst zu sein, dass die durchgeführten Simulationen Ergebnisse zu dem Modellverhalten liefern, welche nicht zwingend denen des realen Systems entsprechen. Als Project Albert 1997 gestartet worden ist, ist man noch von der Annahme ausgegangen, dass es möglich sei, alle möglichen Schlachtausgänge nach Charakteristik der Einsatzkräfte und der Operation, zu bestimmen. Man kam schnell zu dem Schluss, dass dies nicht möglich sei und Horne wird wie folgt zitiert:

„[...]You can never cover the landscape because you’re up against virtual infinity.[...] there are so many different factors that we could be up against [...]“

Aus diesem Grund ist zu betonen, dass das Data Farming lediglich zur Entscheidungsunterstützung, nicht jedoch zur Ausgangsvorhersage, beiträgt [Maryann Lawlor, 2005]. Beim Data Farming handelt es sich um eine interdisziplinäre und iterative Gruppenarbeit und Maryann Lawlor [2005] zitiert Horne bezüglich des iterativen Data-Farming-Prozess wie folgt:

„You can grow data in interesting regions then add parameters and components that are important. After that’s done, we run it and that’s the whole metaphor of data farming. You’re not just growing data - a million pieces of data - and mining it. You are going back in and growing more data.“

Ist die Ausführung des Modells beendet, so können die Ergebnisse analysiert werden und anhand dieser Analyse bestimmt werden, ob das Modell korrekt erstellt worden ist. Zudem wird geprüft, ob das erstellte Modell die Essenz der zugrundeliegenden Frage hinreichend widerspiegelt. Ist dies nicht der Fall, so kann das Modell schnell angepasst und erneut ausgeführt werden, was den iterativen Charakter des Data Farmings unterstreicht [Barry u. Koehler, 2004].

Im Rahmen der Parametervariation können Seeds (vgl. Abschnitt 2.2.1) genutzt werden, um die zufällige Veränderung der Parameter zu erwirken. Hierbei sollte laut Horne u. Meyer [2004] darauf geachtet werden, dass bei jeder Modellausführung der Seed für die Bestimmung der Zufallszahlen mit jeder Ausführung von dem initialen Seed abweicht. Die Wissenschaftler nutzen die unterschiedlichen Seedwerte somit, um die Verteilung der

Zufallsergebnisse und den Einfluss des Zufalls auf das Gesamtszenario zu betrachten. Ein Simulationsmodell wird hierzu unter Verwendung der selben Parameter mehrere Male mit unterschiedlichen Seeds ausgeführt [Maryann Lawlor, 2005]. Durch das Speichern der verwendeten Seedwerte ist es dem Anwender möglich, ein Experiment unter den gleichen Zufallseinflüssen erneut auszuführen und dieses um weitere Eingabegrößen zu erweitern. Somit ist es möglich, neue Daten aus dem ursprünglichen Experiment, mit den selben Zufallswerten, zu farmen. Zudem ermöglicht die Speicherung der Seedwerte die erneute Ausführung von Simulationsmodellen, um die Ursachen für ein anormales Verhalten feststellen zu können [Sanchez, 2014]. Barry u. Koehler [2004] nennt das Ausführen eines Modells mit den selben Parametern, jedoch mit variierenden Seedwerten als eine Möglichkeit, wie ein Modell innerhalb einer Data-Farming-Umgebung ausgeführt werden kann.

Horne u. Meyer [2004] nennt zwei Gründe, wie das Data Farming bei der Simulation und der Entscheidungsfindung unterstützen kann. Zum einen erlaubt das Data Farming die V & V des erzeugten Modells. Die Reaktion des Modells auf viele zufällige Eingabegrößen und unvorhersehbarer Kombinationen dieser stellt einen funktionalen Test (vgl. Abschnitt 2.1.1) für das Modell dar. Die ermittelten Ergebnisse können analysiert und zur Überprüfung der verwendeten Algorithmen genutzt werden. Zum anderen nennt Horne u. Meyer [2004] die Szenarioanalyse als einen Vorteil des Data Farmings. Hierbei werden die Daten analysiert, die durch das Ausführen des Simulationsmodells entstehen. Anhand dieser Analyse können neue Informationen gewonnen und Fragestellungen aus der Realität beantwortet werden. Das Data Farming erlaubt es dem Anwender, das Modell über eine große Anzahl von Parametern auszuführen, welche mittels zufälliger Variationen derart verändert werden, dass der Anwender eine bessere Sicht auf die möglichen Szenarien bekommt. Diese Analyse und Variation des Modells erlaubt es zudem, die Systemdynamik besser zu verstehen.

Damit Modelle in eine Data-Farming-Umgebung eingepflegt werden können, müssen sie ein Basisszenario mittels einer XML-Textdatei vorweisen und ein textbasiertes Outputformat nutzen [Horne u. Meyer, 2004]. Sanchez [2014] schlägt vor, den Prozess der Datensammlung zu automatisieren und so oft wie möglich durchzuführen.

Sanchez [2014] spricht im Rahmen des Data Farmings von den „3 F des Data Farmings“. Die F stehen für *factors*, *features* und *flexibility*.

Mit *factors* sind die Inputs des Simulationsmodells gemeint, die bei Variation das Modell derartig beeinflussen, dass der Anwender neues Wissen über das Systemverhalten erhält. Sanchez [2014] schlägt vor, dass die betrachteten *factors* über vergleichsweise große Intervalle betrachtet werden sollen, statt auf kleine limitiert zu sein.

Features bezeichnen Simulationsrückmeldungen. In diesem Zusammenhang nennt Sanchez [2014] so genannte *thresholds*, ab denen das Systemverhalten sich plötzlich ändert, wie z.B. eine Parameterkombination, ab der eine Schlacht gewonnen oder verloren wird.

Flexibility bezieht sich auf den Wunsch, viele Fragen über das Simulationsmodell beantworten zu können, selbst wenn die Fragen zum Zeitpunkt der Modellherstellung nicht bekannt sind. Somit ist ein restriktives Modelldesign von Nachteil und die Erweiterbarkeit

des Modells von hoher Bedeutung.

Sanchez u. Wan [2012] nennt zwei Simulationsdesigns, die im Rahmen des Data Farmings vermieden werden sollten. Der erste Fall ergibt sich, wenn viele am Projekt beteiligte Personen interessante Parameterkombinationen vorschlagen, welche im Anschluss analysiert werden. Der Fokus in diesen Studien liegt somit in einigen gewissen Szenarien, wo sich viele Faktoren gleichzeitig verändern und die Ursache für Veränderungen nicht mehr eindeutig ersichtlich wird. Sanchez [2014] betont ausdrücklich die Wichtigkeit der *smarten* Datengenerierung, aus der ein klareres Bild der Datenlandschaft entsteht und der Anwender zu nützlichen Erkenntnissen der zugrundeliegenden kausalen Beziehungen gelangt. Der zweite Fall, der nach [Sanchez u. Wan, 2012] vermieden werden sollte ist, wenn ein Basisszenario ausgewählt und die vorhandenen Parameter einzeln verändert werden. Die Erkennung einer Interaktion zwischen den Parametern ist somit nicht mehr möglich, was sich negativ auf die Vollständigkeit der Datenlandschaft auswirkt.

Nachdem die theoretischen Grundlagen für die vorliegende Arbeit erklärt worden sind, folgt im nächsten Kapitel die Erstellung eines Simulationsmodells, um Daten zu *farmen*.

3 Modellierung des Data-Farming-Konzepts

Dieses Kapitel beinhaltet den Hauptteil dieser Arbeit. Das Ziel, ein Data-Farming-Konzept zu erstellen, wird im Rahmen einer Simulationsstudie erläutert und die Ergebnisse dokumentiert. Im Anschluss folgt eine Verifikation des erstellten Simulationsmodells. Danach werden die erzeugten Ergebnisse untersucht.

3.1 Die Simulationsstudie

Zur Durchführung der Simulationsstudie wird das Vorgehensmodell von Rabe et al. [2008] genutzt (vgl. Abbildung 2.1). Hierzu wird auf Grundlage der Zielbeschreibung, der Aufgabenspezifikation und der Systemanalyse ein Konzeptmodell erarbeitet. Anschließend wird dieses Konzeptmodell in ein formales Modell überführt. Mittels Implementierung dieses formalen Modells entsteht das ausführbare Modell in Plant Simulation. Im folgenden werden die Ergebnisse und die Vorgehensweise für jeden der Schritte aus dem Vorgehensmodell beschrieben. Die jeweiligen Phasenergebnisse bauen auf den Erkenntnissen aus Abschnitt 2.1.1 und den Checklisten aus Anhang A auf.

3.1.1 Zielbeschreibung

Die Zielbeschreibung ist der einzige Schritt der Definitions- und Angebotsphase des Vorgehensmodells. Im Rahmen der Vergabe dieser Arbeit wurde bereits mit den Betreuern eine Zielbeschreibung erstellt (vgl. Abbildung B.1), welche die Erstellung eines Simulationsmodells in Plant Simulation und die anschließende Entwicklung eines zugehörigen Data-Farming-Konzepts vorsieht. Da die Ziele der Arbeit mit Hilfe der ereignisdiskreten Simulation erreicht werden sollen, stimmt die Zielsetzung der vorliegenden Arbeit größtenteils mit der Zielbeschreibung der durchzuführenden Simulationsstudie überein (vgl. Abbildung B.1). Somit werden in diesem Abschnitt die Ziele der Simulationsstudie auf Basis der Zielbeschreibung der vorliegenden Arbeit definiert.

Zur Erhöhung der Aussagekraft des Modells und der Ergebnisse sollen Durchläufe unter Parametervariation stattfinden. Um diese Parametervariation durchführen zu können, sollen die BEs des entstandenen Simulationsmodells benutzerdefinierte Attribute erhalten, welche die Parameter der realen Objekte darstellen. Da es sich beim Data Farming um

eine Technik handelt, die eine möglichst große Datenlandschaft generieren soll (vgl. Abschnitt 2.5), ist die Qualität und ausreichende Verfügbarkeit von Echtdateien elementar. Das zu entwickelnde Konzept soll für eine SC anwendbar sein und eine Datenlandschaft generieren, die den Transportvorgang von Kühlschränken zwischen First-Tier-Lieferanten und dem OEM darstellt. Hierzu soll eine fiktive SC modelliert werden, die für die Vorstellung des Konzepts als Grundlage dient. Da das Ziel einer SC die Erreichung des besten Ergebnisses im gesamten Netzwerk darstellt, ist die Kenntnis über *alle* möglichen Szenarien von Bedeutung (vgl. Abschnitt 2.4). Der Einsatz eines Data-Farming-Konzepts kann gerade in diesem Bereich mit „What-If-Szenarien“ bei der Entscheidungsunterstützung von Nutzen sein (vgl. Abschnitt 2.5). Auch zur Gewährleistung der Transparenz über das Netzwerk sind die Ergebnisse des Data Farmings von Bedeutung, da in den nachfolgenden Schritten der Wissensgewinnung neue Zusammenhänge ersichtlich werden können, die dem OEM ein besseres Verständnis über die gesamte Wertschöpfungskette verschaffen (vgl. Abschnitt 2.4). Das Ziel von First-Tier-Lieferanten innerhalb von SCs ist neben einer transparenten SC, die Möglichkeit proaktiv handeln zu können. Ist eine große Datenlandschaft (vgl. Abschnitt 2.5) erzeugt worden, so sind viele Szenarien bereits identifiziert und dem First-Tier-Lieferanten bekannt, was das frühzeitige Einschreiten ermöglicht (vgl. Abschnitt 2.4). Da es sich um „What-If-Szenarien“ handelt, können im Vorfeld bereits Überlegungen stattfinden und Vorgehensweisen für bestimmte Szenarien festgelegt werden. Die ermittelten Ergebnisse sollen nach einem erfolgreichen Simulationsdurchlauf in einem standardisierten Format für weitere Analysezwecke vorliegen. Hierbei werden oft die Formate „.xml“, „.xlsx“ oder „.txt“ verwendet, da diese standardisiert und verbreitet sind. In nachfolgenden Arbeiten und Studien sollen Ergebnisse, die mit dem in dieser Arbeit erstellten Konzept, analysierbar sein. Die Portabilität des Modells (vgl. Abschnitt 2.5) ist somit eine zusätzliche Voraussetzung.

Nachdem die Modellerstellung abgeschlossen ist, soll das ausführbare Modell mittels funktionaler Tests verifiziert werden, da das Data Farming eine entscheidungsunterstützende Eigenschaft besitzt und ohne die Verifikation falsche Ergebnisse entstehen können.

Im folgenden wird die Zielbeschreibung präziser definiert und die Aufgabendefinition erstellt.

3.1.2 Aufgabendefinition

Die Aufgabendefinition ist der erste Schritt der Durchführung der Simulationsstudie. In diesem Schritt des Vorgehensmodells wird die *Aufgabenspezifikation*, durch Konkretisierung der Zielbeschreibung erzeugt. Die Aufgabenspezifikation dient im Anschluss als Grundlage zur Erzeugung des Konzeptmodells. In Abschnitt 2.1.1 wurden die in dieser Phase durchzuführenden Schritte bereits erläutert.

Der Zweck des zu entwickelnden Simulationsmodells ist die Ermöglichung von Data-Farming-Aktivitäten in Plant Simulation. Somit ist die zu verwendende Lösungsmethode

per Zielbeschreibung festgelegt. Da Plant Simulation häufig für die Modellierung von Materialflüssen und SCs verwendet wird, bietet es sich an, ein Data-Farming-Konzept für eine Lieferkette abzubilden. Hierbei sollen benutzerdefinierte Attribute erzeugt werden, welche als Grundlage der Data-Farming-Aktivitäten dienen und die Datenlandschaft erzeugen. Mittels zufälliger Parametervariation wird die Datenlandschaft erzeugt und aufgespannt. Die Anzahl der deklarierten benutzerdefinierten Attribute bestimmt dabei die Fläche der Datenlandschaft. Um die Verständlichkeit des fiktiven Modells zu gewährleisten, wird die Anzahl daher bewusst niedrig gewählt, da das Konzept und nicht die Ergebnisse des Modells im Vordergrund steht.

Als Systemgrenzen des Modells werden der Warenausgang des First-Tier-Lieferanten und der Wareneingang des OEM gesetzt. Dies geschieht aus dem Grund der Reduktion des Informationsbeschaffungsaufwands. Die dem First-Tier-Lieferanten vorgelagerte SC, als auch die dem OEM nachgelagerte SC (vgl. Abbildung 2.9) werden in diesem Modell nicht weiter betrachtet. Abbildung 3.1 stellt eine grobe Veranschaulichung der fiktiven SC dar, in der das zu entwickelnde Data-Farming-Konzept zum Einsatz kommt. Es soll der Transportvorgang von Kühlschränken zwischen First-Tier-Lieferanten und einem OEM nachgestellt werden.

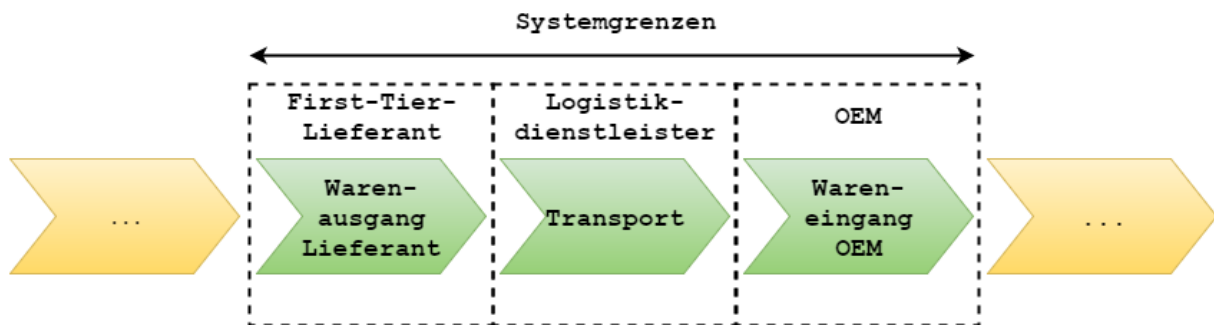


Abbildung 3.1: Systemgrenzen

Die relevanten Systemkomponenten der betrachteten SC beinhalten den OEM, First-Tier-Lieferanten und einen Logistikdienstleister. Der Logistikdienstleister wurde hinzugezogen, um die Transporte zwischen den First-Tier-Lieferanten und dem OEM nachzustellen (vgl. Abschnitt 2.4). Hierzu werden die zu liefernden Produkte am Warenausgang der First-Tier-Lieferanten abgeholt und zum OEM transportiert. Im Rahmen von Abschnitt 2.4 wurde ermittelt, dass Logistikdienstleister in der jüngsten Zeit nicht nur ausschließlich Transportaufgaben übernehmen, sondern auch weitere Dienstleistungen anbieten. Aus diesem Grund wird festgelegt, dass der Logistikdienstleister neben der Transportfunktion auch Qualitätsarbeiten für den OEM übernehmen soll. Hierzu muss ein benutzerdefiniertes Attribut *Qualität* erstellt werden, welche eine Auskunft über die Qualität des jeweiligen Kühlschranks gibt. Für den OEM hat dies den Vorteil, dass er sich auf sein Kerngeschäft fokussieren kann und nur eine kleine Abteilung für Qualitätssicherung und -prüfung be-

nötigt. Der Logistikdienstleister kann hingegen sein Dienstleistungsportfolio ausweiten und neue Geschäftsbereiche am Markt etablieren. Zudem wird festgelegt, dass die Dauer des Transportvorgangs abhängig vom Volumen und Gewicht eines Kühlschranks ist, was die Erzeugung eines benutzerdefinierten Gewichtsattributs erfordert. Das Volumen kann aus den unterschiedlichen Abmessungen der Kühlschränke errechnet werden, welche im Rahmen der Datenbeschaffung als Echtdateien vorliegen.

Da das vorrangige Ziel die Generierung von Daten und die Erzeugung eines Data-Farming-Konzepts ist, ist die Modellierung von mehreren Warenausgängen verschiedener First-Tier-Lieferanten zweckmäßig. Die Erhöhung der Anzahl der Warenausgänge gleicht einer Erhöhung der Quellen (vgl. Tabelle 2.5) in dem Simulationsmodell, wodurch häufiger Bestellungen erzeugt und zum OEM transportiert werden, was wiederum in einer quantitativen Erhöhung der generierten Daten resultiert.

Der OEM soll über verschiedene Wareneingänge verfügen, die nach Produktlinie aufgeteilt sind. Hierzu kann anhand eines Attributs oder dem Zusammenwirken mehrerer Attribute in dem Simulationsmodell eine Unterscheidung getroffen werden, um die Produkte in verschiedene Produktgruppen einzuteilen. Dieser Sachverhalt würde die anfallenden Kommissionierarbeiten für den OEM reduzieren, da der Logistikdienstleister für die korrekte Zuordnung der Waren verantwortlich ist.

Der Detaillierungsgrad des Simulationsmodells ist nicht explizit festgelegt. Da es sich um eine fiktive SC handelt, reicht ein zweckmäßiger Nachbau einer Lieferkette, an dem Lieferant, Logistikdienstleister und OEM beteiligt sind. Ziel und Zweck des Simulationsmodell ist nicht die detailgetreue Modellierung einer echten SC, sondern die beispielhafte Erstellung eines Konzepts, was das Data Farming in Plant Simulation als Simulationswerkzeug ermöglicht.

Um die Benutzerfreundlichkeit sowie die Wiederverwendbarkeit des Data-Farming-Konzepts zu verbessern, sollen externe Schnittstellen eingebaut werden. Hierbei handelt es sich um Import- und Exportschnittstellen. Diese Schnittstellen erlauben dem Anwender die Nutzung eigener Daten und die Untersuchung der Ergebnisse in anderen Applikationen.

Die Abnahmekriterien gemäß der Checkliste in Abbildung B.2 wurden bereits in der Zielbeschreibung definiert und bedürfen keiner weiteren Erklärung. Tabelle 3.1 zeigt die Ergebnisse der Aufgabendefinition basierend auf Abbildung B.2 und Abschnitt 2.1.1 und stellt die Aufgabenspezifikation dar.

Aktivität	Ausprägung
Modellzweck	- Ermöglichung des Data Farmings in Plant Simulation und in SCs
Festlegung der Systemgrenzen	- Warenausgang von First-Tier-Lieferanten - Wareneingang des OEM
Relevante Systemkomponenten	- mehrere First-Tier-Lieferanten - Logistikdienstleister für Transport und Qualitätssicherung - OEM mit mehreren Wareneingängen
Detaillierungsgrad	- für die SC nicht vorgegeben, da diese fiktiv ist
Wiederverwendbarkeit	- Wird durch Import- und Exportfunktion ermöglicht
Lösungsmethode	- Tecnomatix Plant Simulation (Version: 12.0.3)
Schnittstellen zu externen Systemen	- Import und Export von Daten
geplante Modellnutzung	- Durchführung von Data-Farming-Aktivitäten und Analyse der Ergebnisse
Abnahmekriterien	- vorgefertigtes Beispielszenario - Übertragung von benutzerdefinierten Attributen (Parameter) - Parametervariation per Zufall - Importmöglichkeit für Parameter - Exportmöglichkeit der Ergebnisse in einem standardisierten Format - Verifikation des Simulationsmodells durch funktionalen Test

Tabelle 3.1: Aufgabenspezifikation

Im nächsten Schritt wird die Datenbeschaffung durchgeführt, die zur Ermittlung von Rohdaten benötigt wird.

3.1.3 Datenbeschaffung

Die Rohdaten für das Simulationsmodell werden direkt von einem Hersteller für Kühlschränke beschafft. In diesem Fall handelt es sich um Bosch Home (<http://www.bosch->

home.com/de/produkte/kuehlen-gefrieren/kuehlschraenke.html). Hierbei werden die Werte aller Produktlinien für Kühlschränke mit und ohne Gefrierfach betrachtet und protokolliert. Der Hersteller gibt für jeden seiner Kühlschränke verschiedene (konstruktions-)technische Werte an, welche die Grundlage für die Parametervariation in den später folgenden Data-Farming-Aktivitäten darstellen. Abbildung 3.2 zeigt hierbei beispielhaft die vom Hersteller angegebenen Daten für einen Kühlschrank aus der Produktlinie „Serie 6“ (Produktnummer: KIR81AD30). Die Parameter für Höhe, Breite, Tiefe, Energieverbrauch, Nutzinhalt und Geräuschwert aller Kühlschränke aus den genannten Kategorien sind einheitlich auf der Homepage von Bosch Home aufgelistet. Die Daten werden sukzessive protokolliert, um diese im Anschluss im Rahmen der Datenaufbereitung zu verwenden.

Produktbeschreibung

Leistung und Verbrauch
Energie-Effizienz-Klasse: A++
Energieverbrauch: 116 kWh/Jahr
Nutzinhalt gesamt: 319 l
Geräuschwert: 37 dB(A) re 1 pW
Design und Komfort
Kühlteil
Technische Informationen
Türanschlag rechts, wechselbar
Komfortable, einfache Montage
Klimaklasse SN-T
Gerätemaße (H x B x T): 177,2 cm x 55,8 cm x 54,5 cm
Nischenmaße (H x B x T): 177,5 cm x 56,0 cm x 55,0 cm
Anschlusswert: 90 W
220 - 240 V
Zubehör

Abbildung 3.2: Beispielhafte technische Werte für einen Kühlschrank

Es stehen insgesamt 43 Kühlschränke zur Verfügung, für welche Daten protokolliert werden. Einige Kühlschränke weisen die exakt selben Abmessungen auf und unterscheiden sich nur in Werten wie Geräuschwert, Nutzinhalt oder Energieaufnahme. Dies deutet darauf hin, dass es sich bei diesen Kühlschränken um Produktvariationen handelt. Die ermittelten Rohdaten werden in einer Excel-Tabelle festgehalten, welche im Rahmen der folgenden Datenaufbereitung wiederverwendet wird.

3.1.4 Datenaufbereitung

Die Datenaufbereitung erfolgt mit Hilfe von Microsoft Excel. Hierbei werden alle zuvor protokollierten Parameter der Kühlschränke übersichtlich aufbereitet. Es werden hierfür 3 Blätter in der Excel-Arbeitsmappe angelegt, um unter anderem zwischen Kühlschränken

mit und ohne Gefrierfach zu unterscheiden. Im dritten Blatt werden die Ergebnisse zusammengetragen. Es wird innerhalb dieser Blätter ebenfalls zwischen den verschiedenen Produktserien unterschieden. Neben der Zuordnung zu den Serien und den bereits genannten Parametern wird ebenfalls ein Link zu dem entsprechenden Kühlschrank hinterlegt, um die Daten verifizieren zu können.

Hierzu wurden die Daten zu den Abmessungen, dem Energieverbrauch in kWh/Jahr, dem Nutzinhalt in Liter und dem Geräuschwert in db(A) re 1 pW (Geräuschwert bezogen auf 1 Pikowatt Leistung) für insgesamt 43 Kühlschränke ohne Gefrierfach und 36 Kühlschränke mit Gefrierfach auf der Homepage von Bosch Home in Microsoft Excel 2016 ausgewertet. Mittels der Minimum- und Maximum-Funktion von Excel, wurden die unterschiedlichen Minima und Maxima für jeden Parameter ermittelt. Dies erlaubt die Betrachtung der realen Intervalle für die jeweiligen Parameter und ermöglicht weitere Überlegungen zur Anpassung dieser. Anhand der Minima und Maxima aller 43 Kühlschränke und somit unabhängig dessen, ob der Kühlschrank ein Gefrierfach besitzt oder nicht, werden im dritten Arbeitsblatt der Excel-Arbeitsmappe die Ergebnisse zusammengetragen und reale Intervalle definiert. In Tabelle 3.2 werden diese Intervalle dann vergrößert, um die betrachtete Datenlandschaft zu vergrößern und die Granularität der Daten zu verfeinern.

Die Auswertung ergab, dass die minimale Höhe aller von Bosch Home produzierten Kühlschränke bei 82cm und die maximale Höhe bei 187cm liegt. Die Breite der betrachteten Kühlschränke liegt zwischen 54,1cm und 66cm. Die Tiefe der betrachteten Kühlschränke bewegt sich zwischen 54,2cm und 65cm. Der Energieverbrauch aller Kühlschränke liegt zwischen 65kWh/Jahr und 229 kWh/Jahr, während der Nutzinhalt zwischen 123 Liter und 346 Liter beträgt. Die jeweiligen Geräuschwerte bewegen sich in einem Intervall von 33 bis 41 db(A) re 1 pW.

Wie bereits in der Aufgabendefinition erwähnt, sollen zwei weitere Parameter *Qualität* und *Gewicht* betrachtet werden, die das Systemverhalten beeinflussen. Da für diese Parameter keine Echtwerte zur Verfügung stehen, werden plausible (aber fiktive) Werte angenommen. Der Parameter *Qualität* liegt im Intervall von 0 bis 100 und stellt einen Parameter dar, der jedem Kühlschrank einen quantitativen Qualitätswert zuordnet. In der Realität kann sich dieser Parameter durch unterschiedliche Zusammensetzungen und ist von Produkt zu Produkt verschieden. Gerade für die Umsetzung von Qualitätsarbeiten durch den Logistikdienstleister wird dieser Parameter benötigt, da anhand dessen unterschieden werden soll, ob ein Produkt abgewiesen werden soll oder nicht. Die Beschaffung dieses Parameters ist somit aus externer Sicht nur schwer bis gar nicht möglich. Da das Modell jedoch auch eine Importfunktion für Echtdateien aufweisen soll wird dieses Attribut dennoch eingeführt, um dem späteren Anwender eine Möglichkeit zu geben, Produkte anhand ihrer Qualität unterschiedlichen Produktlinien einordnen zu können und qualitativ minderwertige Produkte auszuschließen. Die Ermittlung des Parameters *Gewicht* ist über die Homepage von Bosch Home ebenfalls nicht möglich gewesen, weshalb ein realistisches Intervall von 20kg bis 90kg gewählt worden ist. Anhand dieses Parameters wird im späteren Modellverlauf

die Transportzeit der Waren vom First-Tier-Lieferanten an den OEM modelliert, wobei schwerere Kühlschränke eine längere Lieferzeit als leichtere besitzen.

Tabelle 3.2 zeigt übersichtlich die tatsächlich protokollierten und die anschließend im Simulationsmodell zu verwendenden Parameter. Zur Vergrößerung der betrachteten Datenlandschaft, wurden die tatsächlichen Intervalle vergrößert.

Parameter	Tatsächliches Intervall	Im Modell verwendetes Intervall
Höhe	82-187 cm	70-200 cm
Breite	54,1-66 cm	40-80 cm
Tiefe	54,2-65 cm	40-80 cm
Energieverbrauch	65-229 kWh/Jahr	50-240 kWh/Jahr
Nutzzinhalt	123-346 Liter	110-360 Liter
Geräuschwert	33-41 db(A) re 1 pW	20-60 db(A) re 1 pW
Gewicht	keine Werte	20-90 kg
Qualität	keine Werte	0-100

Tabelle 3.2: Intervalle der benutzerdefinierten Attribute

Die aufbereiteten Daten befinden sich auf dem beigelegten Datenträger im Anhang als Excel-Arbeitsmappe mit dem Namen „Daten Bosch.xlsx“. Im Anschluss an die Datenaufbereitung folgt die Systemanalyse zur Erzeugung eines Konzeptmodells.

3.1.5 Systemanalyse

Wie bereits im Rahmen der Aufgabendefinition erläutert, handelt es sich bei den geplanten Systemgrenzen um die Warenausgänge von First-Tier-Lieferanten und den Wareneingängen eines OEM. Die Prozessschritte vor dem Warenausgang der First-Tier-Lieferanten und nach dem Wareneingang beim OEM werden nicht weiter betrachtet, da eine solche Betrachtung den Detaillierungsgrad erhöhen und den Rahmen dieser Arbeit übersteigen würde. Der Schritt der Datenbeschaffung wäre in diesem Fall aufwändig und in der Realität kaum möglich, da viele unterschiedliche Unternehmen an einem Wertschöpfungsnetzwerk beteiligt sind (vgl. Abschnitt 2.4) und Daten für entsprechend viele Unternehmen beschafft und aufbereitet werden müssten. Daher beschränkt sich diese Simulationsstudie auf den Material- und Informationsfluss zwischen dem Warenausgang von First-Tier-Lieferanten und den Wareneingängen des OEM.

Als weitere Systemgrenze wird das Ausschusslager des Logistikdienstleisters festgelegt. Da dieser Qualitätsarbeiten für den OEM übernehmen soll, müssen die entsprechenden Objekte umgeleitet werden, um nicht an den OEM geliefert zu werden. Prozesse, die nach

dem Ausschuss von Waren stattfinden werden nicht weiter betrachtet.

Da die Erstellung des Data-Farming-Konzeptes im Vordergrund steht, sind interne Wechselwirkungen von Bedeutung. Hierzu sollen die Eingabegrößen innerhalb des Modells als Grundlage verwendet werden, um unterschiedliche Systemantworten zu erzeugen. Auch im Rahmen der „3 F des Data Farmings“ sollen so genannte *thresholds* eingeführt werden (vgl. Abschnitt 2.5). Ab einer bestimmten Parameterkombination soll sich das Modell abweichend vom bisherigen Vorgehen verhalten. Um zu neuen Ergebnissen und Ausgabegrößen zu gelangen, muss ein Teil der Eingabedaten dazu verwendet werden, neue Ausgabedaten zu erzeugen. Hierbei soll die Transportzeit der Waren abhängig vom Volumen und Gewicht des Kühlschranks sein. Das Ergebnis der Multiplikation soll entscheiden, wie lange der Transport von einem First-Tier-Lieferanten zum OEM dauert. Die exakte Festlegung dieser *thresholds* erfolgt im Rahmen der Modellformalisierung.

Außerdem sollen Qualitätsprüfungen von dem Logistikdienstleister durchgeführt werden. Basierend auf dem Parameter *Qualität* sollen Waren unterschiedlich behandelt werden. Waren mit einer niedrigen Qualität sollen direkt als Ausschuss deklariert und nicht an dem OEM weitergeleitet werden. Auch im Rahmen der Einführung von unterschiedlichen Produktlinien ist dieser Parameter von Bedeutung. Es sollen Intervalle definiert werden, die die Zuordnung eines Kühlschranks zu einer der Produktlinien bestimmen.

Des Weiteren sollen im Rahmen des Detaillierungsgrades Lieferverzögerungen einer SC modelliert werden, um die Auswirkungen dieser in die späteren Experimente einbeziehen zu können. [Arndt, 2014] hat hierbei in ihrer Arbeit bereits erste Ansätze vorgestellt, welche als Grundlage zur Umsetzung von Lieferverzögerungen in dieser Arbeit dienen und im Rahmen der Modellformalisierung näher erklärt werden.

Da es sich bei der SC um eine fiktive SC handelt, müssen bestimmte Annahmen getroffen werden. Unter anderem wird davon ausgegangen, dass der Logistikdienstleister jeden zu transportierenden Kühlschrank einzeln liefert. Außerdem wird festgelegt, dass der OEM pro Produktlinie lediglich nur ein BE gleichzeitig aufnehmen kann. Dieses vorgehen würde in der Realität zu hohen Kosten führen, ist im Rahmen der fiktiven SC jedoch zu vernachlässigen. Da das vorrangige Ziel die Erzeugung einer möglichst großen Datenlandschaft ist und nicht die möglichst reale Umsetzung eines Liefervorgangs an einen OEM, spielt diese Annahme für das Data-Farming-Konzept keine große Rolle. Durch spätere Variationen und Modifikationen des Simulationsmodells kann diese Annahme aufgehoben und an die eigenen Bedürfnisse angepasst werden. Da es sich hierbei um einen fiktiven Materialfluss handelt, ist eine Beschaffung von echten Transportzeiten nicht möglich beziehungsweise notwendig.

Es wird auch eine Annahme über das Attribut *Qualität* getroffen, welches für die Qualitätsprüfung durch den Logistikdienstleister verwendet wird. Es wird angenommen, dass dieses Attribut dem Simulationsanwender innerhalb einer Skala von 0 bis 100 zur Verfügung steht. Da die tatsächliche Beurteilung der Qualität eines Kühlschranks komplexer ist, ist es dem Simulationsanwender überlassen, ein eigenes Zuordnungssystem zu entwickeln und seine

eigenen Qualitätsanforderungen in eine solche Skala zu übertragen. Ansonsten muss das Simulationskonzept derart verändert werden, dass die benutzerdefinierte Qualitätsskala eingepflegt werden kann.

Obwohl der Detaillierungsgrad durch diese Annahmen reduziert wird, haben sie keine großen Auswirkungen auf das zu erzeugende Data-Farming-Konzept. Es soll lediglich eine Grundlage für die Verwendung von Echtdateien erzeugt werden, ohne dass diese vorliegen. Aus diesem Grund ist es dem späteren Simulationsanwender überlassen, für die Genauigkeit der Eingabegrößen und die korrekte Darstellung der Transportzeiten zu sorgen. In dieser Arbeit wird ausschließlich ein Konzept zur Umsetzung von Data-Farming-Aktivitäten vorgestellt, weshalb der bestehende Echtdateienmangel keine Auswirkung auf die Ergebnisse der Simulationsstudie hat.

Die Konzeptstruktur für das Simulationsmodell ist in Abbildung 3.3 zu sehen. Die Quellen des Modells stellen den Warenausgang eines Lieferanten dar, während die Senken des Modells einen der Wareneingänge des OEM darstellen. Diese Wareneingänge werden nach Produktlinie aufgeteilt. Die Waren gelangen durch eine Lieferung an den OEM. Die Lieferung wird hierbei vom Logistikdienstleister durchgeführt. Neben dieser Tätigkeit übernimmt der Logistikdienstleister zudem Qualitätsarbeiten für den OEM. Vor einer Lieferung wird die Qualität der Ware geprüft und entsprechend dieser an den jeweiligen Wareneingang beziehungsweise die jeweilige Senke geliefert. Der Materialfluss beginnt somit bei einem First-Tier-Lieferanten und endet entweder beim OEM oder in dem Ausschuss des Logistikdienstleisters. Dieser ist für den hauptsächlichen Materialfluss verantwortlich. Damit der Materialfluss beginnt, muss der OEM bei einem First-Tier-Lieferanten eine Bestellung platzieren. Nur so kann dieser die Ware bereitstellen und dem Logistikdienstleister mitteilen, dass die Waren an den OEM geliefert werden sollen. Informationen über die technischen Daten und der Herkunft des Objekts werden (in Form von benutzerdefinierten Attributen) ebenfalls an den OEM übermittelt.

Als Eingabegrößen werden die im Rahmen der Datenbeschaffung ermittelten Werte gewählt. Diese Werte sollen in zufälligen Kombinationen an die produzierten Kühlschränke übergeben werden, um die mögliche Datenlandschaft so gut wie möglich erzeugen zu können. Die Ausgabegrößen ergeben sich durch die internen Wechselwirkungen zwischen den Komponenten. Abbildung 3.3 zeigt die geplanten Eingabe- und Ausgabegrößen.

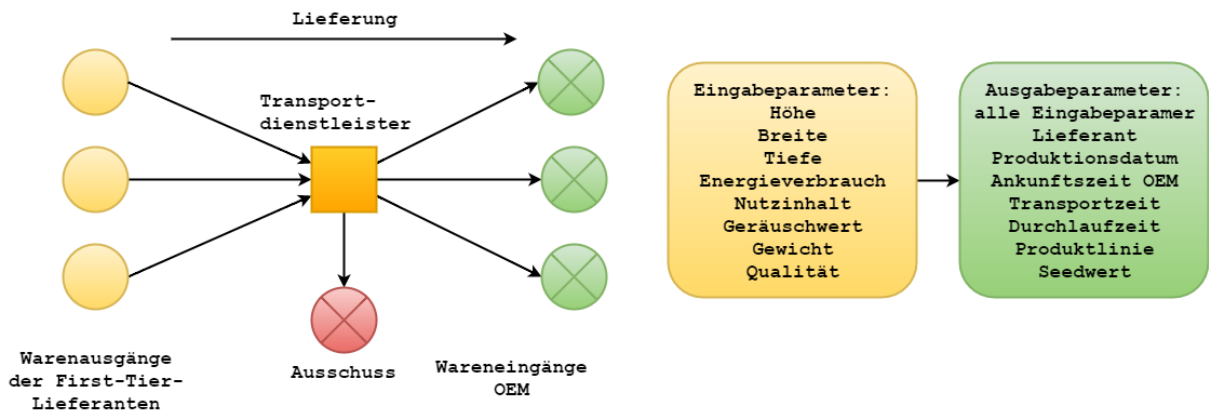


Abbildung 3.3: Konzeptstruktur

Um die Genauigkeit des Simulationsmodells garantieren zu können, muss es dem Anwender möglich sein, Echtdata in das Modell einpflegen zu können. Mit Hilfe einer Importfunktion soll ermöglicht werden, dass Parameter in das Modell eingelesen werden können, die dem Anwender vorliegen. Diese Schnittstelle wird im Rahmen der COM-Schnittstelle von Microsoft Excel realisiert, welche von Plant Simulation unterstützt wird. Mit Hilfe dieser Schnittstelle soll ein Import und Export von Daten ermöglicht werden, um das Modell auf weitere Szenarien übertragen zu können. Für den Import von Daten können z.B. Echtdata aus der laufenden Produktion verwendet werden, mit deren Hilfe die SC betrachtet werden kann. Eine Exportfunktion ermöglicht die Analyse der Ergebnisse und bildet die Grundlage für Data-Mining-Aktivitäten auf der Suche nach neuem Wissen. Gerade für diesen Schritt ist die Qualität der Eingangsdaten von Bedeutung, was mit Hilfe der Importfunktion und Echtdata ermöglicht wird.

Während des Simulationslaufs sollen die Ausgabegrößen für jeden Kühlschrank in einer Tabelle übersichtlich protokolliert werden. Außerdem muss es dem Anwender möglich sein, diese Tabelle in ein standardisiertes Format zu exportieren, um die Ergebnisse in weiteren Systemen und Programmen untersuchen zu können.

Der Detaillierungsgrad des Simulationsmodells beschränkt sich lediglich auf die Plausibilität der verwendeten Attribute und Parameter, da das Ziel der Arbeit die Entwicklung eines Data-Farming-Konzepts ist. Im Rahmen der Konzeptentwicklung standen keine Echtdata zur Verfügung, weshalb Daten ermittelt worden sind, die sich einfach in das Modell implementieren lassen und die Funktionsweise dessen demonstrieren. Die protokollierten Parameter enthalten Werte für *Höhe*, *Breite*, *Tiefe*, *Energieverbrauch*, *Nutzhalt* und *Geräuschwert* der betrachteten Kühlschränke. Darüber hinaus sollen im Rahmen der Modellerstellung die beiden Parameter *Qualität* und *Gewicht* *hinzugefügt* werden, welche auf der Homepage nicht ermittelt werden konnten und daher plausibel definiert werden müssen.

Es werden für die Modellierung dieses Systems keine Teilmodelle benötigt, weshalb die Modellierung in einem einzigen Simulationsmodell stattfindet. Die Verwendung von Stan-

dardbauteilen ist für die Erstellung des Modells ausreichend. Zudem kann die Modellierung in der 2D-Ansicht durchgeführt werden, da die entstandenen Daten am Ende des Simulationsdurchlaufs von Bedeutung sind und nicht die tatsächlichen Arbeitsvorgänge.

Zur Durchführung der Simulation wird Tecnomatix Plant Simulation in Version 12.0.3 unter Verwendung einer Studentenlizenz benutzt. Außerdem kommt ein Apple MacBook Air mit einem Intel Core i5 Prozessor und 8 GB Arbeitsspeicher zum Einsatz, welches mittels Bootcamp unter Windows 10 betrieben wird.

Tabelle 3.3 fasst die im Rahmen der Systemanalyse erarbeiteten, neuen, Ergebnisse zusammen.

Aktivität	Ausprägung
Struktur und Komponenten des Systems	- siehe Abbildung 3.3
Wechselbeziehungen zwischen den Systemkomponenten	- siehe Abbildung 3.3 - Lieferverzögerungen während des Transports möglich
Systemgrenzen	- Warenausgang von 3 First-Tier-Lieferanten - Wareneingang des OEM für 3 verschiedene Produktlinien - Ausschuss des Logistikdienstleisters
Anforderungen an den Detaillierungsgrad sowie Umfang und Granularität der Daten	- Parameter vom Hersteller - fiktive Parameter: Gewicht und Qualität
Material- und Informationsfluss	- Material <i>fließt</i> von Lieferant über Logistikdienstleister zum OEM - Informationsfluss wird durch den OEM mit der Bestellung eines Kühlschranks initiiert - Informationen über technische Aspekte und Herkunft der Waren werden vom Lieferanten mit der bestellten Ware an den OEM übermittelt
Eingangs- und Ausgangsgrößen	- siehe Abbildung 3.3
Detaillierungsgrad	- Nutzung der technischen Daten von Bosch Home - Erweiterung der Parameter für größere Datenlandschaft
Verwendete Daten- und Informationsquellen	Homepage von Bosch Home (http://www.bosch-home.com/de/produkte/kuehlen-gefrieren/kuehlschranke.html)
Auswirkungen von Datenmangel	- keine, da die Konzeptentwicklung im Vordergrund steht
Lösungsmethode	- Tecnomatix Plant Simulation (Version: 12.0.3)

Nicht zu modellierende Systemkomponenten	<ul style="list-style-type: none"> - Prozesse vor dem Warenausgang der First-Tier-Lieferanten - Prozesse nach dem Wareneingang des OEM - Prozesse nachdem der Logistikdienstleister Waren als Ausschuss deklariert hat - Exakte Transport- und Herstellzeiten
Verwendeter Computer	- Apple MacBook Air mit Intel Core i5 4260U Prozessor und 8 GB Arbeitsspeicher und Windows 10 als Betriebssystem (Bootcamp)
Schnittstellen zu externen Systemen	- Import und Export von Daten über COM-Schnittstelle von Microsoft Excel

Tabelle 3.3: Konzeptmodell

Anschließend an diesen Schritt folgt der Prozess der Modellformalisierung, wo das Konzeptmodell in das formale Modell überführt wird.

3.1.6 Modellformalisierung

Gemäß Abbildung B.4 müssen zunächst die in dem Konzeptmodell definierten Systemkomponenten während der Modellformalisierung formal beschrieben werden. Die Hauptkomponenten sind hierbei die First-Tier-Lieferanten, der OEM und der Logistikdienstleister. Die First-Tier-Lieferanten, beziehungsweise deren Warenausgänge, werden in dem Simulationsmodell als Quellen (vgl. Abschnitte 2.3 und 2.3.1) definiert. Dieser Baustein erzeugt innerhalb des Modells neue BEs, welche das abgebildete Netzwerk durchlaufen. Die BEs stellen in diesem Zusammenhang Kühlschränke dar. Der Logistikdienstleister ist für den Übergang der BEs vom Warenausgang der First-Tier-Lieferanten zu den Wareneingängen des OEM verantwortlich. Diese Systemkomponente wird in Form einer Einzelstation (vgl. Abschnitte 2.3 und 2.3.1) dargestellt. Die Bearbeitungsdauer (vgl. Abschnitt 2.3) der Einzelstation soll die Zeit nachmodellieren, die der Logistikdienstleister für den Transport der BEs zu den Wareneingängen des OEM benötigt. Da im Rahmen der Systemanalyse festgelegt worden ist, dass der Logistikdienstleister neben den Transportaufgaben zusätzlich noch Qualitätsarbeiten für den OEM übernimmt, muss eine Senke (vgl. Abschnitte 2.3 und 2.3.1) in das Modell eingefügt werden, welche die BEs vernichtet, die aussortiert werden. Zur Modellierung der tatsächlichen Wareneingänge des OEMs werden ebenfalls Senken verwendet. Wie bereits in Abbildung 3.1 abgebildet, werden die nachfolgenden Prozesse nicht weiter betrachtet, weshalb der Einsatz von Senken an diesem Punkt der SC zweckmäßig ist. Aus dem selben Grund werden Quellen für die Warenausgänge der First-Tier-Lieferanten verwendet.

Lieferant1 soll eine Normalverteilung für den *Abstand* zwischen zwei Bestellungen mit

den Werten „1:04:48:00, 1:09:36:00, 0, 6:00:00:00“ aufweisen, welche im Format μ , σ , *untere Schranke*, *obere Schranke* eingegeben werden. Diese Werte entsprechen einem μ von 1,2, einem σ von 1,4, werden jedoch in das Attributfeld *Abstand* im Format „DD:HH:MM:SS.XXXX“ eingetragen, wobei *DD* für die Anzahl an Tagen, *HH* für die Anzahl an Stunden, *MM* für die Anzahl an Minuten, *SS* für die Anzahl an Sekunden und *XXXX* für Sekundenbruchteile stehen. Die untere Schranke der Normalverteilung liegt bei null Tagen und die obere Schranke bei sechs Tagen.

Lieferant2 soll eine Normalverteilung mit den Werten „2:07:12:00, 2:14:24:00, 0, 7:00:00:00“ aufweisen, was einem μ von 2,3, einem σ von 2,6 entspricht. Die untere Schranke des Lieferanten beträgt wie beim ersten Lieferanten null Tage, während die obere Schranke bei sieben Tagen liegt.

Im Gegenzug dazu soll *Lieferant3* die Werte „3:09:36:00, 3:19:12:00, 0, 8:00:00:00“ aufweisen. Das μ für den dritten Lieferanten liegt somit bei 3,4 und das σ bei 3,8. Auch dieser Lieferant weist somit eine untere Schranke von null Tagen auf, während die obere Schranke mit acht Tagen den höchsten Wert aller drei Lieferanten annimmt. In Tabelle 3.4 sind die unterschiedlichen Parameter für die Normalverteilungen aller Lieferanten übersichtlich aufgeführt. Diese Werte dienen als Grundlage für die drei unterschiedlichen First-Tier-Lieferanten und sind in ihren Einstellungen in diesen Ausprägungen eingestellt.

Lieferant	μ	σ	Obere Schranke	Untere Schranke
1	1,2	1,4	0 Stunden	6 Stunden
2	2,3	2,6	0 Stunden	7 Stunden
3	3,4	3,8	0 Stunden	8 Stunden

Tabelle 3.4: Normalverteilungen der Lieferanten

Neben den Abstandszeiten für die Lieferanten, muss die Transportdauer der Bestellungen bestimmt werden. Wie bereits während der Systemanalyse definiert, soll die Transportzeit zum OEM abhängig von den Parametern *Volumen* und *Gewicht* der zu liefernden Kühl-schränke sein. Hierzu wurden fiktive Intervalle festgelegt, die sich aus der Multiplikation dieser beiden Parameter ergeben. Basierend auf dem Ergebnis der Multiplikation soll der Logistikdienstleister zwischen 2 und 6 Stunden für den Transport benötigen. Die hierfür festgelegten Intervalle sind in Tabelle 3.5 abgebildet und können durch eine Methode (vgl. Abschnitt 2.3.1) als Eingangssteuerung (vgl. Abschnitt 2.1) der Einzelstation umgesetzt werden.

Ergebnis Gewicht x Volumen	Lieferzeit
<= 20.000.000	2 Stunden
<= 40.000.000	3 Stunden
<= 60.000.000	4 Stunden
<= 80.000.000	5 Stunden
> 80.000.000	6 Stunden

Tabelle 3.5: Transportzeit des Logistikdienstleisters

Da der Logistikdienstleister zusätzlich Qualitätsprüfungen übernehmen soll, muss eine Grenze gesetzt werden, ab der BEs abgelehnt werden. Basierend auf diesem Attribut sollen zudem die verschiedenen Wareneingänge des OEMs angesteuert werden, da per Definition festgelegt worden ist, dass der OEM für jede Produktgruppe einen unterschiedlichen Wareneingang besitzt. Damit nicht zu viel Ausschuss generiert wird, werden nur Kühlschränke abgelehnt, die eine Qualität kleiner 10 besitzen. Dieser Wert kann im Nachhinein angepasst werden, falls strengere Anforderungen an die zu liefernden Teile gestellt werden sollten. Wird ein Kühlschrank abgelehnt, so wird dieser zu einer separaten Senke transportiert und vernichtet. Kühlschränke mit einer Qualität zwischen 10 und 40 werden der Produktgruppe *Basic* zugeordnet und an den entsprechenden Wareneingang transportiert. Diese Produktgruppe soll preissensitive Kunden ansprechen. Ist die Ausprägung des Parameters Qualität zwischen 40 und 70, so wird der Kühlschrank der Produktlinie *Advanced* zugeordnet. Kühlschränke der Produktlinie *Advanced* haben als Zielgruppe Kunden, die ein ausgewogenes Preis-Leistungs-Verhältnis suchen. Ist die Qualität des Kühlschranks größer 70, so wird dieser der *Premium*-Linie zugeordnet. Diese Kühlschränke stellen die qualitativ hochwertigsten Produkte des OEM dar und befinden sich im höheren Preissegment. Diese Überlegungen leiten sich aus den Produktlinien von Bosch Home ab. Eine Übersicht der unterschiedlichen Produktlinien, sowie deren Anforderungen, ist in Tabelle 3.6 abgebildet. Durch eine Methode als Eingangssteuerung der Einzelstation kann diese Qualitätsprüfung umgesetzt werden. Es ist dabei zu betonen, dass dieser Prozess innerhalb der Eingangssteuerung stattfinden muss, da für BEs die als Ausschuss deklariert worden sind, keine Transportzeit berechnet und simuliert werden soll und diese direkt vernichtet werden sollen.

Wert des Attributs Qualität	Produktlinie
< 10	Ausschuss
10-40	Basic
40-70	Advanced
> 70	Premium

Tabelle 3.6: Zugehörigkeit zu Produktlinien anhand des Attributs Qualität

Nachdem der Logistikdienstleister die Qualitätsprüfungen erfolgreich abgeschlossen hat, beginnt der Transport der Waren zum Bestimmungsort. Wie bereits im Rahmen der Systemanalyse festgelegt, stellen Lieferverzögerungen des Logistikdienstleisters eine weitere Systemkomponente dar, die in dem Simulationsmodell nachgestellt werden soll. Als Grundlage für das folgende Funktionsprinzip dient [Arndt, 2014]. In ihrer Arbeit hat Arndt [2014] ein Konzept entwickelt, welches die Ist- und Solllieferzeiten eines Lieferanten nachstellt. Hierzu wurden zwei Einzelstationen und eine Senke verwendet.

Arndt [2014] hat hierfür in ihrer Arbeit die Echtzeiten empirisch untersucht und entsprechend den Ergebnissen eine Tabelle erstellt, welche die Verteilungen für Ist- und Solllieferzeiten der Lieferanten enthält. Im Gegensatz zu den Ergebnissen von Arndt [2014], wurden in dieser Arbeit die Lieferzeitabweichungen auf wenige Stunden reduziert, da das Ziel der Arbeit die Erzeugung möglichst vieler Daten ist. Tabelle 3.7 zeigt eine für die Zwecke dieser Arbeit angepasste Version der Lieferabweichungen der Lieferanten. Die absoluten Häufigkeiten wurden nicht verändert und entsprechen den Ergebnissen von Arndt [2014].

Lieferabweichung in Stunden	Häufigkeit
1:00:00.0000	54
2:00:00.0000	23
3:00:00.0000	10
4:00:00.0000	13
5:00:00.0000	3
6:00:00.0000	7
7:00:00.0000	3
8:00:00.0000	4
9:00:00.0000	4

Tabelle 3.7: Verteilung der Lieferabweichung

Die Umsetzung der Lieferverzögerungen wird in Abschnitt 3.2 anschaulich erläutert. Abbildung 3.4 zeigt eine grafische Darstellung des formalen Modells als warteschlangenbasiertes Diagramm.

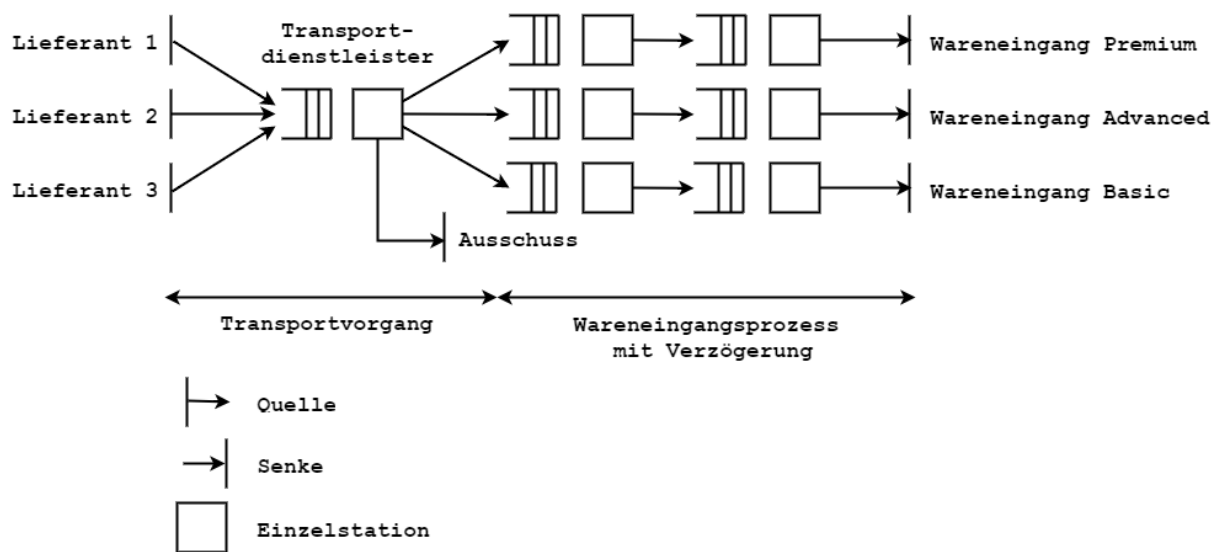


Abbildung 3.4: Formales Modell Grafisch

Tabelle 3.8 fasst, angelehnt an Abbildung B.4, die Ergebnisse aus dem Schritt der Modellformalisierung zusammen.

Aktivität	Ausprägung
Systemkomponenten	<ul style="list-style-type: none"> - 3 First-Tier-Lieferanten mit jeweils einem Warenausgang - 1 OEM mit 3 Wareneingängen - 1 Logistikdienstleister
Steuerungen	<ul style="list-style-type: none"> - Qualitätsprüfung durch den Logistikdienstleister (vgl. Tabelle 3.6) - Transportzeit abhängig von den Parametern <i>Gewicht</i> und <i>Qualität</i> (vgl. Tabelle 3.5) - Abbildung von Lieferverzögerungen (vgl. Tabelle 3.7)
Stochastische Parameter und deren Charakteristika	<ul style="list-style-type: none"> - alle technischen Parameter aus Abbildung 3.3: Gleichverteilung - Lieferverzögerungen: Empirische Verteilung (vgl. Tabelle 3.7)

Tabelle 3.8: Formales Modell

Im Anschluss an dieses Phasenergebnis folgen die Phasen V & V, Implementierung und Experimente und Analyse, welche im folgenden Abschnitt behandelt werden.

3.1.7 V & V, Implementierung, Experimente und Analyse

Im Rahmen der Zielbeschreibung ist nur eine Verifikation des Simulationsmodells mittels funktionaler Tests gefordert (vgl. Abbildung B.1). Das Vorgehen hierzu wird in Abschnitt 3.3 geschildert. Die Verifikation der Eingabedaten und Parameter erfolgte durch Überprüfung der korrekten Übernahme der Werte von der Homepage des Herstellers in die Excel-Arbeitsmappe und in das Simulationsmodell. Eine Validierung des Modells ist nicht gefordert.

Die Implementierung des formalen Modells erfolgt in Abschnitt 3.2, während die Phase der Experimente und Analyse in Abschnitt 3.4 betrachtet und durchgeführt wird.

3.2 Entwicklung des Simulationsmodells

Die Simulation der SC erfolgt, wie bereits in Abschnitt 3.1 festgelegt, im 2D-Modus von Plant Simulation, da es sich um eine fiktive strategische Simulation auf Konzernebene handelt, welche sich auf die Lieferkette eines Unternehmens (in diesem Beispiel ein Hersteller für Weiße Ware) konzentriert (vgl. Abschnitt 2.1). Diese Art der Simulation wird auch SCS genannt (vgl. Abschnitt 2.3). Die Nutzung von Standardelementen ist für das vorgeschlagene Konzept ausreichend (vgl. Abschnitt 3.1).

Abbildung 3.5 zeigt eine Übersicht des in diesem Abschnitt erläuterten Data-Farming-Konzepts in Plant Simulation. Die entsprechende Datei befindet sich auf dem dieser Arbeit beiliegenden Datenträger und trägt den Namen „Datafarming.spp“.

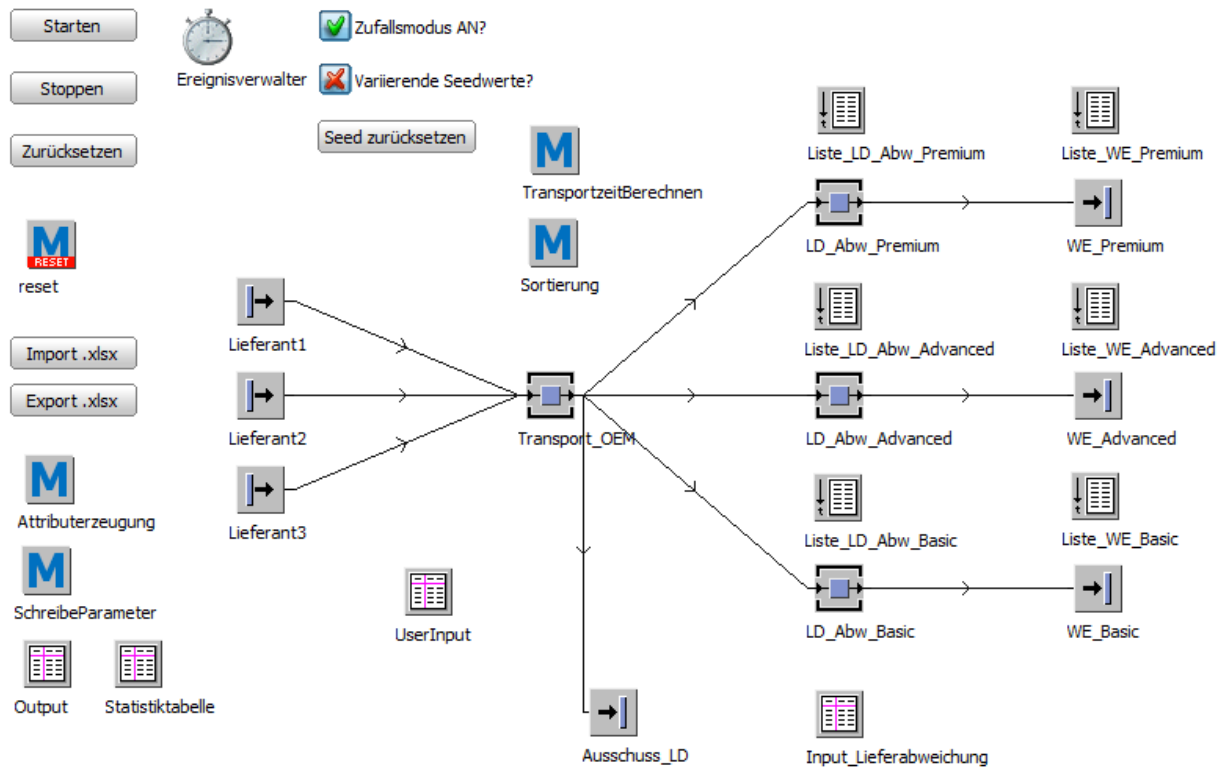


Abbildung 3.5: Übersicht des Simulationsmodells in Plant Simulation

Es wird ein Simulationsmodell erstellt, welches sich am klassischen Aufbau einer SC orientiert und den Liefervorgang von Waren und Produkten an den OEM nachstellt. Das Modell enthält temporäre aber keine permanenten Elemente (vgl. Abschnitt 2.1). Die für die Modellierung zwingend notwendigen temporären Elemente stellen die Bestellungen dar, welche in einer der drei Quellen mit den Namen *Lieferant1*, *Lieferant2* und *Lieferant3* erzeugt werden und in einer der Senken mit den Namen *WE_OEM_Premium*, *WE_OEM_Advanced*, *WE_OEM_Basic* oder *Ausschuss_LD* vernichtet werden. Permanente Elemente, wie z.B. Gabelstapler oder Förderstrecken, werden nicht benötigt und sind somit nicht Bestandteil des Modells.

Die Anzahl der Quellen beträgt entsprechend den Ausarbeitungen aus der Simulationsstudie (vgl. Abschnitt 3.1.2) 3, kann aber beim späteren Einsatz des Simulationsmodells durch *ableiten* erhöht werden (vgl. Abschnitt 2.3). Die abgeleiteten neuen Lieferanten *erben* die Parameter und Attribute der Klasse, von der abgeleitet worden ist. In dem Ausgangsszenario besitzen die drei Lieferanten jeweils unterschiedliche Attribute *Abstand* und *Start*, damit die nachfolgende Einzelstation mit dem Namen *Transport_OEM* nicht direkt zu Beginn des Modells gleichzeitig Bestellungen von den Quellen erhält.

Die Werte für die Abstandszeiten der Quellen wird Tabelle 3.4 entnommen. In das Attributfeld *Start* wurden, je nach Lieferant, exakt die gleichen Werte für eine Normalverteilung eingetragen (vgl. Tabelle 3.4), damit die erste Bestellung und somit das erste BE nicht direkt zum Start der Simulation erzeugt wird. Da in dem Modell drei verschiedene Lieferan-

ten vorhanden sind, würde ohne dieses Attribut jeder Lieferant direkt zum Simulationsstart (zum Simulationszeitpunkt $t = 0$) eine Bestellung erzeugen und an die nachfolgende Einzelstation transportieren, was nicht möglich ist, da Einzelstationen immer nur ein BE bearbeiten können (vgl. Abschnitt 2.3.1). Um Staus zu vermeiden, wurden aus diesem Grund Startwerte vergeben.

Das Attribut *BE-Auswahl* besitzt die Ausprägung *konstant* und sorgt somit für eine konstante Erzeugung von BEs der Klasse *Bestellung*. Hierbei handelt es sich um Kühlschränke, die an einen OEM geliefert werden und unterschiedliche Ausprägungen in ihren Attributen aufweisen.

Jeder Lieferant besitzt eine Eingangssteuerung (vgl. Abschnitt 2.1) mit dem Namen *AttributErzeugung*. Die mit der Steuerung verbundene Methode wird ausgeführt, sobald der jeweilige Lieferant ein neues BE erzeugt. Der Code der Methode ist im Anhang unter Quellcode A.1 abgebildet.

In den Zeilen 2 bis 9 werden insgesamt 8 Variablen definiert, die im späteren Quellcode dazu dienen, die verwendeten Parameter per Zufall zu bestimmen. Anhand der Namen der Variablen wird ersichtlich, für welchen Parameter sie verwendet werden. In Zeile 10 wird eine Variable mit dem Namen *Seedwert* definiert, welcher im Verlauf des Quellcodes als Platzhalter für den aktuellen Seedwert des Simulationsdurchlaufs dient (vgl. Abschnitt 2.2.1). Alle Variablen sind vom Typ *integer* (vgl. Abschnitt 2.3).

Im Anschluss an die Definition der Variablen folgt in Zeile 12 ein Befehl, welcher dazu dient, dass der Zähler der Spalte „Erzeugte Einheiten“ in der Tabelle *Statistiktabelle* um eins erhöht wird. Auf die Eigenschaften und Bedeutung dieser Tabelle wird an späterer Stelle eingegangen. Anschließend wird in den Zeilen 14 und 15 für das erzeugte BE ein benutzerdefiniertes Attribut mit dem Namen *Produktionsdatum* vom Typ *datetime* erzeugt (vgl. Abschnitt 2.3.1). In diesem benutzerdefinierten Attribut wird der Wert `Ereignisverwalter.AbsZeit` gespeichert, welcher das aktuelle Datum (inklusive Zeitstempel) innerhalb des Modells darstellt. Somit ist es dem Anwender später möglich, zu erkennen, wann die jeweilige Bestellung erzeugt worden ist.

Anschließend folgt in den Zeilen 17 bis 21 eine *if*-Steuerung, welche das Attribut *Wert* des Buttons mit dem Namen *Random_Seed* (Beschriftung: „Variierende Seedwerte?“) prüft. Ist der Wert des Attributs vom *boolean* Typ *true* (vgl. Abschnitt 2.3, so wird der Befehl `Ereignisverwalter.InkrementiereZufallszahlenVarianteBeiReset := true;` ausgeführt, welcher dazu dient, dass bei jedem Neustart der Simulation der verwendete Seedwert zur Erzeugung der Zufallszahlen um eins erhöht wird und somit vom initialen Seed abweicht (vgl. Abschnitt 2.5). Der Wert des Attributs ist *true*, wenn der Button vom Anwender angeklickt wird und ein grünes Häkchen anzeigt. Der aktuelle Seedwert kann in dem Menü „Extras“ des Ereignisverwalters eingesehen und manipuliert werden (vgl. Abbildung 2.6). Ist das Ergebnis der *if*-Prüfung *false* (Button zeigt ein rotes Kreuz), wird der initiale Seedwert für jeden Simulationsdurchlauf beibehalten und es entstehen die exakt selben Zufallswerte.

Im nächsten Schritt folgt in Zeile 23 erneut eine *if*-Prüfung, welche die Ausprägung des Attributs *Wert* des Bausteins *Random_Check* erfasst. Ist die Ausprägung *true*, folgenden die Anweisungen in den Zeilen 24 bis 51. Ansonsten springt die Steuerung direkt zur Zeile 53 weiter.

Zunächst wird in Zeile 24 die Anweisung

```
seedwert := Ereignisverwalter.ZufallszahlenVariante;
```

ausgeführt, welche der Eingangs definierten Variable mit den Namen *seedwert* den aktuellen Seedwert aus dem Ereignisverwalter zuordnet.

In den Zeilen 25 bis 32 erhalten die Variablen *zufallHöhe*, *zufallBreite*, *zufallTiefe*, *zufallEnergieverbrauch*, *zufallNutzihalt*, *zufallGeräuschwert*, *zufallGewicht* und *zufallQualität* basierend auf einer Gleichverteilung zufällige Werte (vgl. Abschnitt 2.2.1). Die Gleichverteilung wird in Plant Simulation als „*z_gleich*(s,a,b)“ definiert, wobei *s* den Seedwert, *a* die untere Schranke und *b* die obere Schranke darstellt.

Anhand des Quellcodes wird ersichtlich, dass für den Parameter *s* der Gleichverteilung die Variable *seedwert* verwendet wird, welche wiederum den im Ereignisverwalter festgelegten Seedwert darstellt. Somit ist es dem Anwender möglich, anhand des Seedwerts im Ereignisverwalter sämtliche Zufallsvariablen mit nur einer Manipulation zu beeinflussen.

Diese Option erfüllt gleich zwei der „3 F des Data Farmings“ (vgl. Abschnitt 2.5). Zum einen werden die *factors* des Simulationsmodells beeinflusst, welche derartig auf das Modell einwirken, dass der Anwender neues Wissen über das Systemverhalten erhält. Zum anderen wird dem Wunsch der *flexibility* nachgekommen und ein restriktives Modelldesign vermieden.

In den Zeilen 25 bis 32 werden anschließend die zufälligen (*integer*-)Werte für die benutzerdefinierten Attribute erzeugt. Die Ermittlung der Intervallgrenzen zur Erzeugung der zufälligen Attribute wird auf Grundlage von Tabelle 3.2 festgelegt.

Nachdem den Variablen Zufallswerte zugewiesen worden sind, werden in den Zeilen 34 bis 51 die jeweiligen Parameter für den aktuellen BE erzeugt. Jedes BE erhält acht benutzerdefinierte Attribute (vgl. Tabelle 3.2). Die Parameter sind exakt wie die Zufallsvariablen vom Typ *integer* und erhalten den Wert dieser Variablen.

Ist das Ergebnis der *if*-Prüfung aus Zeile 23 *false*, treten die Zeilen 53 bis 80 in Kraft. Hierbei werden erneut die selben Attribute aus Tabelle 3.2 erzeugt, mit dem Unterschied, dass die Werte für die erzeugten Parameter aus der Tabelle *UserInput* übernommen werden. Dies ermöglicht dem Anwender den Import von Echtdateien aus der laufenden Produktion oder der Vergangenheit. Das Importieren der Echtdateien kann auf zwei unterschiedliche Arten erfolgen. Zum einen kann der Button mit dem Namen „Import.xlsx“ verwendet werden. Der Code dieses Buttons ist in Quellcode A.2 abgebildet. Dieser Button ermöglicht den Import einer Microsoft Excel Datei im Format „.xlsx“ und ist somit nur ab Microsoft Excel 2004 verfügbar. Entgegen den Empfehlungen aus Abschnitt 2.5, wurde bewusst nicht das „.xml“-Format gewählt, da dieses zu Komplikationen beim Import und Export geführt hat. Zudem ist der Export der Ergebnisse im „.xlsx“-Format strukturierter und

verständlicher. Der Unterschied beider Datenformate liegt in der Strukturierung der Daten. Wird der Button „Import.xlsx“ benutzt, so ist darauf hinzuweisen, dass sich die zu importierende Datei in dem gleichen Ordner wie die verwendete Datei für das Simulationsmodell (Endung „.spp“) befinden und den Namen „Input.xlsx“ tragen muss. Das Gegenstück dieser Methode ist in A.3 abgebildet, welche den Inhalt der Tabelle *Output* in eine Excel-Datei exportiert, welche den Namen „Output.xlsx“ trägt. Die entsprechende Datei wird ebenfalls an dem Speicherort der Datei für das Simulationsmodell erzeugt.

Die zweite Methode zum Datenimport erfolgt, indem die Tabelle *UserImport* im Netzwerkfenster (vgl. Abschnitt 2.3) geöffnet wird und der Button *Importieren* in dem Menü *Liste* ausgewählt wird (vgl. Abbildung 3.6). Dieser Button erlaubt die Auswahl einer Datei in dem Format „.obj“, „.txt“, „.xml“ oder den Excel-Formaten „.xls“, „.xlsx“, „.xlsm“ und „.xlsb“.

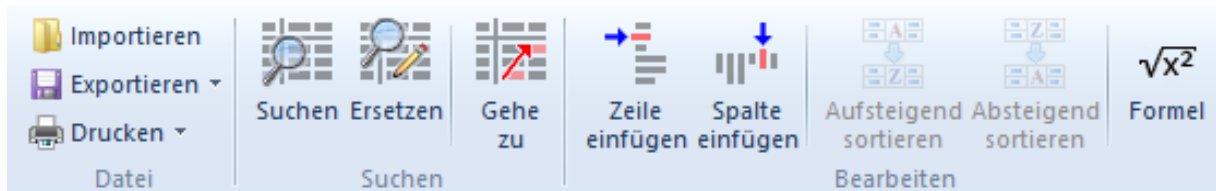


Abbildung 3.6: Import von Echtdateien in das Modell

Auch das manuelle Eintippen von Parameterwerten wird unterstützt, ist aber ohne Automatisierung des Inputs aufgrund der Natur des Data Farmings sehr zeitaufwändig und nicht empfehlenswert (vgl. Abschnitt 2.5).

Werden Parameter importiert, so ist darauf hinzuweisen, dass die Parameter *Höhe*, *Breite*, *Tiefe*, *Volumen*, *Gewicht*, *Nutzzinhalt*, *Geräuschwert*, *Energieverbrauch* und *Qualität* in der zu importierenden Datei vorhanden sein müssen. Außerdem müssen alle Werte den Datentyp *integer* aufweisen.

Wurden Parameter importiert, so werden diese den BEs zugeordnet. Hierbei werden den BEs nacheinander die entsprechenden Parameter übergeben. Basierend auf dem Attribut *id* des aktuellen BEs, werden die Parameterwerte aus der entsprechenden Zeile verwendet. Ein BE mit dem Attribut *id* = 5 ist das fünfte erzeugte BE für den Simulationsdurchlauf und erhält die Parameterwerte aus der fünften Zeile der Tabelle *UserInput*. Eine beispielhafte Datei mit dem Namen „Input.xlsx“ befindet sich im Ordner „Anhang“ auf dem dieser Arbeit beigelegten Datenträger und kann als Vorlage zum Import von Daten in die Tabelle *UserInput* genutzt werden.

Im Anschluss an die *if*-Prüfung folgt in den Zeilen 82 bis 83 die Erzeugung eines benutzerdefinierten Attributs mit dem Namen *Seedwert*. Dieses Attribut speichert den im dem Ereignisverwalter eingestellten Seedwert in das aktuelle BE ein, damit der Anwender bei der Ergebnisauswertung die Möglichkeit hat, unter Umständen das Modell zu manipulieren und mit dem selben Seedwert erneut auszuführen, um die Auswirkungen der

Anpassungen zu beobachten (vgl. Abschnitt 2.5). Der Befehl hierzu lautet `@.Seedwert := Ereignisverwalter.ZufallszahlenVariante;`

In den Zeilen 85 bis 91 wird ein weiteres Attribut mit dem Namen *Lieferant* vom Typ *string* (vgl. Abschnitt 2.3.1) erzeugt. Dieses Attribut dient dazu, dem Anwender eine Information darüber bereitzustellen, welcher Lieferanten das BE geliefert hat. Diese Information kann Rückschlüsse auf die Durchlaufzeit und weitere Parameter geben, da jeder Lieferant individuelle Parameter aufweist und somit von den anderen abweicht. Auch bei einer Anpassung des Modells kann die Herkunft des BEs eine entscheidende Rolle spielen, wenn es darum geht, die gewonnenen Informationen auszuwerten.

Nachdem die Methode *Attributerzeugung* ausgeführt worden ist, verlässt das BE den Warenausgang des First-Tier-Lieferanten (vgl. Abschnitt 2.4) und wird an die Einzelstation (vgl. Abschnitte 2.3 und 2.3.1) mit dem Namen *Transport_OEM* geliefert. Diese Einzelstation stellt einen Logistikdienstleister dar, welcher zwei Aufgaben hat. Die erste Aufgabe ist der klassische Transport der Waren vom First-Tier-Lieferanten an den OEM. Die zweite Aufgabe ist die Qualitätsprüfung der zu liefernden Teile. Erfüllt ein BE die Qualitätskriterien des OEM nicht, wird er vom Logistikdienstleister an die Senke mit dem Namen *Ausschuss_LD* transportiert, was einen gesperrten Bereich innerhalb des Lagers des Logistikdienstleisters darstellt. Hier kann der Logistikdienstleister etwaige Korrekturarbeiten vornehmen (falls möglich) oder aber das BE direkt vernichten. Eine Erweiterungsmöglichkeit für das Modell ist der Einbau einer Einzelstation zur Reparatur von BEs, die in den gesperrten Bereich gelangen.

Die Einzelstation *Transport_OEM* hat drei wichtige Eigenschaften. Zum einen ist das Attribut *Bearbeitungszeit* standardmäßig auf „2:00:00“ eingestellt, was zwei Stunden entspricht. Zum anderen besitzt diese Einzelstation jeweils eine Eingangs- und Ausgangssteuerung, die den Kurs des aktuellen BEs bestimmen.

Die Eingangssteuerung ist eine Methode mit dem Namen *TransportzeitBerechnen*. Der Code dieser Methode ist in Quellcode A.4 zu sehen. Diese Methode hat die Aufgabe, die Weiterleitung des BEs durch den Logistikdienstleister an den OEM nachzubilden. Hierzu wird zunächst in Zeile 3 eine *if*-Schleife aufgerufen, welche prüft, ob das Attribut *Qualität* des aktuellen BEs einen Wert größer gleich 10 besitzt. Ist dies nicht der Fall, so wird das BE direkt an die Senke mit dem Namen *Ausschuss_LD* weitergeleitet. Die Methode wird durch den *return*-Befehl in Zeile 5 beendet und die restlichen Anweisungen werden nicht ausgeführt.

Ist die Qualität des BEs größer gleich 10, wird in Zeile 8 eine *if*-Schleife ausgeführt, welche das Ergebnis der Multiplikation aus Gewicht und Volumen überprüft. Diese Berechnung besitzt die Überlegung, dass der Logistikdienstleister je nach Kombination aus Gewicht und Volumen des BEs mehr Zeit für den Transport an den OEM benötigt (vgl. Abschnitt 3.1.2). Aus diesem Grund kann einer von fünf Fällen eintreten, welche in Tabelle 3.5 aufgelistet sind. Dieses Verhalten des System erfüllt den Punkt *Features* der „3 F des Data Farmings“ (vgl. Abschnitt 2.5). Die verschiedenen Simulationsrückmeldungen und Ergebnisse der

Multiplikation aus Gewicht und Volumen stellen verschiedene *thresholds* dar, ab denen sich das Simulationsverhalten verändert.

In den Zeilen 20 bis 21 wird das letzte benutzerdefinierte Attribut mit dem Namen *Transportzeit* erstellt, welches den Datentyp *time* besitzt (vgl. Abschnitt 2.3). Diesem Attribut wird der Wert der Bearbeitungszeit (beziehungsweise der Transportzeit in der Realität) der Einzelstation *Transport_OEM* zugewiesen. Der Anwender hat später somit die Möglichkeit, die benötigte Transportzeit des Logistikdienstleisters an den OEM zu sehen und kann dementsprechend Rückschlüsse ziehen. Hierzu muss das Modell erweitert werden. Eine Möglichkeit wäre die Priorisierung von qualitativ hochwertigen Produkten in der Belieferung des OEM. Je nach Anwendungsfall kann das in dieser Arbeit vorgelegte Basisszenario erweitert und angepasst werden. Im Rahmen der „3 F des Data Farmings“ (vgl. Abschnitt 2.5) ermöglicht dieses offene Design die Erfüllung der Punkte *factors* und *flexibility*. Nachdem die Eingangssteuerung ausgeführt worden ist, vergeht in der Simulation diejenige Simulationszeit (vgl. Abschnitt 2.1.1), die der Einzelstation *Transport_OEM* zugeordnet worden ist. Anschließend wird die Ausgangssteuerung *Sortierung* ausgeführt, welche in Quellcode A.5 abgebildet ist.

Diese Methode leitet die BEs beim Verlassen der Einzelstation an einen der drei Wareneingänge des OEM weiter. Hierzu wurden, angelehnt an die Produktlinien von Bosch Home, drei fiktive Produktlinien erfunden, welche die Namen *Basic*, *Advanced* und *Premium* tragen. Die *if*-Schleife in Zeile 3 prüft den Wert des benutzerdefinierten Attributs *Qualität* des BEs und führt entsprechende Aktionen aus. Ist der Wert kleiner als 40, so wird das BE an die Einzelstation mit dem Namen *LD_Abw_Basic* weitergeleitet. Ist der Wert größer gleich 40 aber kleiner als 70, so wird das BE an die Einzelstation mit dem Namen *LD_Abw_Advanced* weitergegeben. Sobald die Qualität größer gleich 70 ist, wird das BE an die Einzelstation *LD_Abw_Premium* weitergeleitet. Tabelle 3.6 verdeutlicht diesen Vorgang unter der Berücksichtigung, dass Produkte mit einer Qualität kleiner 10 keiner Produktlinie zugeordnet werden und vom Logistikdienstleister als Ausschuss deklariert werden. Dieses Verhalten des System erfüllt den Punkt *features* der „3 F des Data Farmings“ (vgl. Abschnitt 2.5). Die verschiedenen Produktgruppen stellen mit ihren Qualitätsanforderungen verschiedene *thresholds* dar, ab denen sich das Simulationsverhalten verändert. Dieses Prinzip stellt einen Sortiervorgang beim Logistikdienstleister dar. Entsprechend der *Qualität* des BEs wird dieser an einen bestimmten Wareneingang beim OEM geliefert. Dieser Schritt des Modells kann auch dementsprechend interpretiert werden, dass der OEM drei verschiedene Werke besitzt, wovon jeder exakt einer Produktlinie zugeordnet wird. In einer möglichen Erweiterung, könnte die Distanz zu den verschiedenen Werken genauer modelliert werden. Entsprechend der Distanz zum vorhergesehenen Wareneingang beziehungsweise zum vorhergesehenen Werk, könnte die Transportzeit des Logistikdienstleisters variiert werden.

BEs mit einer Qualität kleiner als 10 sind in dieser Methode bewusst nicht vorhanden, weil es sich um eine Ausgangssteuerung handelt. Würden diese BEs erst an dieser Stelle auf

ihre Qualität geprüft werden, hätte in der Methode *TransportzeitBerechnen* bereits eine Berechnung der Transportzeit vom Logistikdienstleister zum OEM stattgefunden und die Ausgangssteuerung wäre erst *nach* Ablauf dieser Zeit ausgeführt worden. In der Realität würde dies bedeuten, dass der OEM auch BEs mit einer unzureichenden Qualität erhalten würde und diese erst am Wareneingang des OEM aussortiert werden würden. Dieses Vorgehen entspricht nicht der vorgenommenen Definition, wo der Logistikdienstleister die Qualitätsprüfung übernimmt und sorgt zudem für Kosten, da Teile transportiert werden, die zum Schluss aussortiert werden. Daher muss die Überprüfung der Qualität des aktuellen BEs mit der Eingangssteuerung erfolgen, um der Einzelstation die korrekten Transportzeiten zuweisen und die qualitativ minderwertigen BEs aussortieren zu können. Ist die Qualität eines BEs größer gleich 10, so wird dieser von der Einzelstation *Transport_OEM* an eine der folgenden Einzelstationen *LD_Abw_Premium*, *LD_Abw_Advanced* oder *LD_Abw_Basic* weitergeleitet. Das genaue Ziel des BEs ist abhängig von der Ausprägung des Qualitätsattributs und kann Tabelle 3.6 entnommen werden. Das folgende Prinzip des Wareneingangs bedarf einer Erklärung und stellt den in der Simulationsstudie festgelegten Lieferverzögerungsprozess dar.

Da die Einzelstation weder eine Eingangs- noch eine Ausgangssteuerung besitzt, beginnt mit dem Eintreffen des BE die *Bearbeitungszeit*. Diese ist vom Typ *dEmp*, was einer diskreten empirischen Verteilung entspricht. Für die Erläuterung dieses Funktionsprinzips bedienen wir uns dem Wareneingang für die BEs der Produktlinie *Premium*. Basierend auf den Werten in der Tabelle *Input_Lieferabweichung* (Werte aus Tabelle 3.7 entnommen) wird die Bearbeitungszeit der Einzelstation *LD_Abw_Premium* eingestellt. Zweck dieser Einzelstation ist die Umsetzung von Lieferabweichungen, welche in einer SC vorkommen können und bei der Analyse von Lieferengpässen von elementarer Bedeutung sind (vgl. Abschnitt 2.4). Die Bearbeitungszeit verzögert somit die Ankunft eines BEs an einer der Senken und somit dem Wareneingang des OEM.

Um die Lieferabweichungen protokollieren zu können, wurden für die Einzelstationen und die Senken aller Produktlinien jeweils Listen erstellt, welche automatisch die Ankunfts- und Abgangszeit eines BEs aufzeichnen. Diese Listen haben in unserem Beispiel die Namen *Liste_LD_Abw_Premium* und *Liste_WE_Premium*. Sie besitzen alle als Startwert „01.01.2015 00:00:00.0000“, was dem Startdatum des Ereignisverwalters entspricht und somit die Aufzeichnung der Ankünfte mit dem Start der Simulation beginnt. Im Reiter *Aufzeichnen* in den Einstellungen der jeweiligen Listen wird der Wert **.AnzahlBEs* eingetragen, wobei „*“ als Platzhalter für den Namen der korrespondierenden Einzelstation dient. Neben den Zeiten für die Ankünfte von BEs an den entsprechenden Einzelstationen werden auch die Zeiten für Abgänge von BEs aufgezeichnet. Die Liste mit dem Namen *Liste_WE_Premium* enthält somit die tatsächlichen Ankunfts- und Abgangszeiten der BEs am Wareneingang des OEM. Dieser Vorgang erfolgt für die beiden weiteren Produktlinien analog.

Das Ankommen eines BEs an einer der Senken *WE_Premium*, *WE_Advanced* oder

WE_Basic, markiert die Ankunft beim OEM. Alle drei Senken besitzen die selbe Eingangssteuerung mit dem Namen *SchreibeParameter*. Der Code dieser Ausgangssteuerung ist in Quellcode A.6 abgebildet.

Zunächst werden in den Zeilen 2 und 3 zwei Methodenspezifische Variablen „y“ vom Datentyp *integer* und „DLZ“ vom Typ *time* definiert, die im Laufe dieses Quellcodes benötigt werden.

In Zeile 5 wird der kürzlich erzeugten Variable „y“ der Wert `Output.YDim + 1`; zugeordnet. Dieser Wert wird aus den Attributen der Tabelle *Output* herangezogen und stellt die aktuell selektierte Zeile dar. Das Ziel dieses Befehls ist es, im späteren Verlauf des Quellcodes die benutzerdefinierten Attribute von ankommenden BEs in eine Tabelle mit dem Namen *Output* nacheinander einzutragen. Ohne diesen Befehl würde die erste Zeile dieser Tabelle mit jeder Ausführung dieser Methode überschrieben werden und Informationen würden verloren gehen.

In Zeile 6 folgt der Befehl `DLZ := Ereignisverwalter.AbsZeit - @.Produktionsdatum`; , welcher der Variable „DLZ“ das Ergebnis der Subtraktion der Werte `Ereignisverwalter.AbsZeit` und `@.Produktionsdatum` zuordnet. `@.Produktionsdatum` bezieht sich hierbei auf die benutzerdefinierte Variable aus der Methode *Attributerzeugung*. Das Ergebnis dieser Subtraktion ist die Zeit, die ein BE benötigt, um vom Warenausgang eines Lieferanten (Quellen *Lieferant1*, *Lieferant2* oder *Lieferant3*) zum Wareneingang des OEM (Senken *WE_Premium*, *WE_Advanced*, *WE_Basic* oder *Ausschuss_LD*) zu gelangen und wird *Durchlaufzeit* genannt.

In den Zeilen 8 bis 21 wird ein neues benutzerdefiniertes Attribut erzeugt, welches den Namen *Produktlinie* trägt und vom Datentyp *string* ist. Anschließend findet eine *if*-Prüfung statt, die das Attribut *Standort* des aktuellen BEs prüft und entsprechend des Ergebnisses dieser Prüfung dem Attribut *Produktlinie* einen Wert zuweist. Da diese Methode nach der Sortierung der BEs durch den Logistikdienstleister stattfindet, ist eine erneute Prüfung anhand des Attributs *Qualität* nicht erforderlich und es genügt, den Standortnamen des aktuellen BEs zu prüfen. Ist der aktuelle Standort die Senke *WE_Premium*, so wird dem Attribut *Produktlinie* der *string*-Wert „Premium“ zugewiesen. Ist der Standort eine der Senken *WE_Advanced* oder *WE_Basic*, so geht die Methode analog vor. Befindet sich das BE auf keiner der genannten Senken, so tritt die *else* Anweisung der *if*-Steuerung ein und es wird dem Attribut *Produktlinie* der *string*-Wert „Ausschuss“ zugeordnet, da die Senke *Ausschuss_LD* die einzige Senke ist, die übrigbleibt.

Abhängig von dem Attribut *Produktlinie* werden in den Zeilen 11, 14, 17 und 20 Anweisungen ausgeführt, die den Inhalt einer Tabelle mit dem Namen *Statistiktabelle* manipulieren. In dieser Tabelle wird die Anzahl aller BEs protokolliert, die über die verschiedenen Senken vernichtet worden sind und somit entweder beim OEM angekommen sind oder nicht den Qualitätsstandards des OEM entsprachen und aussortiert worden sind. Der Simulationsanwender erhält somit eine Übersicht über die Anzahl der erzeugten BEs und wieviele davon beim OEM bereits angekommen sind und welcher Produktlinie sie angehören.

Die Zeilen 23 bis 43 schreiben die Werte aller benutzerdefinierten Attribute in die Tabelle *Output*. Zusätzlich zu den Attributen wird eine Spalte mit dem Namen *AnkunftOEM* erzeugt, in welche die aktuelle Simulationszeit (vgl. Abschnitt 2.1.1) eingetragen wird (*Ereignisverwalter.Abszeit*). Die Tabelle *Output* enthält somit die Spalten: „AnkunftOEM“, „Produktionsdatum“, „Durchlaufzeit“, „Lieferant“, „Höhe“, „Breite“, „Tiefe“, „Volumen“, „Gewicht“, „Nutzinhalt“, „Geräuschwert“, „Energieverbrauch“, „Qualität“, „Transportzeit“, „Produktlinie“ und „Seedwert“.

In Zeile 36 findet eine *if*-Prüfung statt, welche den Wert des Attributs *Qualität* abfragt. Hierbei wird geprüft, ob das aktuelle BE eine Qualität größer gleich 10 besitzt oder nicht. Ist die Prüfung erfolgreich, so wird in die Spalte „Transportzeit“ in der Tabelle *Output* der Wert für das benutzerdefinierte Attribut *Transportzeit* eingetragen. Schlägt die Prüfung fehl und das BE weist eine Qualität kleiner 10 auf, wird mit dem Befehl `Output["Transportzeit",y] := str_to_time("0:00:00");` in die Spalte „Transportzeit“ der Wert „0:00:00“ eingetragen. Diese Prüfung ist notwendig, da jede Senke die Methode *SchreibeParameter* ausführt und somit zwischen akzeptierten und aussortierten BEs unterschieden werden muss. Aussortierte BEs dürfen keine *Transportzeit* ausweisen (da sie nicht zum OEM transportiert werden) und erhalten in der Tabelle *Output* daher den Wert „0:00:00“ als *Transportzeit*. Zusätzlich hierzu wird mit dem Befehl `Output.setzeHintergrundfarbeZeile(y,makeRGBValue(255,0,0));` die Hintergrundfarbe jeder Zeile, die sich auf ein aussortiertes BE bezieht Rot hinterlegt. Dies gibt dem Anwender eine optische Rückmeldung, dass es sich bei dem BE in der jeweiligen Zeile um ein BE mit einer nicht ausreichenden Qualität handelt, welches aussortiert worden ist. Die *if*-Prüfung in Zeile 45 stellt sicher, dass die Simulation gestoppt wird, sobald keine weiteren Parameter für neue BEs in der Tabelle *UserInput* vorhanden sind. Ist die Anzahl der Zeilen aus der Tabelle *UserInput* gleich der Iteration des aktuell an der Senke angekommenen BEs, so ist das Ergebnis der *if*-Prüfung aus Zeile 45 *true* und es werden die Anweisungen in den Zeilen 46 bis 47 ausgeführt, welche eine Nachricht in der Konsole (vgl. Abschnitt 2.3) ausgeben und die Simulation beendet.

Mit der Ausführung der Methode *SchreibeParameter* endet der Simulationslauf für exakt ein BE. Es können in der Zwischenzeit noch weitere BEs erzeugt worden sein, die sich auf den vorherigen Stationen des Modells befinden und noch nicht an einer der Senken angekommen sind. Das Simulationsmodell wird so oft wiederholt beziehungsweise so lange ausgeführt, bis es vom Anwender, über den Ereignisverwalter, gestoppt wird. Mit Hilfe des Ereignisverwalters kann das Modell zudem auf seinen Ausgangszustand überführt werden und erneut gestartet werden. Hierbei kann unter anderem auch ein abweichender Seedwert für variierende Zufallsergebnisse gewählt werden. Wird das Simulationsmodell auf seinen Ursprungszustand zurückgesetzt, wird automatisch die Methode *reset* ausgeführt. Der Code dieser Methode ist in Quellcode A.7 abgebildet.

Diese Methode dient dazu, den Ausgangszustand des Simulationsmodells wiederherzustellen, indem die Inhalte aller verwendeten Tabellen mit dem `*.löschen`-Befehl gelöscht

werden, wobei das „*“ als Platzhalter für den Namen der zu löschenden Tabelle dient. In Zeile 11 wird zudem mit dem Befehl `Output.Hintergrundfarbe := makeRGBValue(255,255,255)`; die Hintergrundfarbe der Tabelle *Output* auf Weiß zurückgesetzt. Die Anweisung in Zeile 14 dient dazu, die Bearbeitungszeit der Einzelstation *Transport_OEM* auf den Ausgangswert von 2 Stunden zu setzen.

Nachdem in diesem Abschnitt das Simulationsmodell erklärt worden ist, folgt in dem folgenden Abschnitt die Durchführung von funktionalen Tests, um das Data-Farming-Konzept zu verifizieren.

3.3 Verifikation des Simulationsmodells durch funktionalen Test

Nachdem ein mögliches Konzept für das Data Farming in Plant Simulation in Abschnitt 3.2 erläutert worden ist, muss dieses Modell, vor der Ergebnisanalyse, zunächst verifiziert werden. Schlägt die Verifikation des Modells fehl, sind keine gültigen Ergebnisse mit ihnen ermittelbar. Hierzu wurden Arten von Tests in Abschnitt 2.1.1 erklärt, welche im folgenden Anwendung finden.

Die Verifikation des Modells bezieht sich vor allem auf die semantische Korrektheit der verwendeten Methoden. Daher wird Schritt für Schritt jede Methode verifiziert.

Die erste Methode, die in dem Simulationsmodell ausgeführt wird, ist die Methode *Attributerzeugung* (vgl. Quellcode A.1). Diese Methode hat das Ziel, jedem erzeugten BE eine Reihe von benutzerdefinierten Attributen zuzuweisen und diesen zufällige Werte zuzuordnen. Zudem wird der Zähler für die Gesamtanzahl der erzeugten Einheiten in der Tabelle *Statistiktabelle* mit jedem BE erhöht, welches generiert wird. Dieses Verhalten kann mittels der Technik „Beobachten in der Animation“ (vgl. Abschnitt 2.1.1) nachgewiesen werden. Hierzu wird die Animation im Einzelschritt (vgl. Abschnitt 2.3) ausgeführt und mit jedem erzeugten BE der Wert für „Erzeugte Einheiten“ in der Tabelle *Statistiktabelle* betrachtet. Nach einigen erzeugten BEs wird ersichtlich, dass dieser Teil erfolgreich funktioniert.

Auch die Verifikation, dass der Zufallszahlenseed mit jedem Neustart der Simulation inkrementell erhöht wird, kann mithilfe von Einzelschritten überprüft werden. Hierzu wird ein Modell gestartet und es werden einige Simulationsschritte ausgeführt. In dem Ereignisverwalter kann unter dem Menü „Extras“ der aktuelle Zufallszahlenseed eingesehen werden. Setzt man nun die Simulation zurück und startet sie erneut, so wird deutlich, dass der Zufallszahlenseed in dem Ereignisverwalter um eins erhöht worden ist. Dieses Verhalten darf nur dann stattfinden, wenn die Checkbox mit der Beschriftung „Variierende Seedwerte?“ ausgewählt ist. Die Wiederholung des oben beschriebenen Ablaufs bestätigt die vorhergesehene Funktionsweise dieser Checkbox.

Anschließend wird die Funktion der Checkbox mit der Beschriftung „Zufallszahlen AN?“ geprüft. Hierzu wird die Box aktiviert und das Simulationsmodell gestartet. Bei vorher-

gesehenem Verhalten müssen in der Tabelle *Output* eine Reihe von Einträgen erzeugt werden, die sich innerhalb der Intervalle aus Tabelle 3.2 bewegen. Nachdem das Modell für eine längere Simulationszeit ausgeführt wird, wird sie angehalten und die Ergebnisse der Tabelle *Output* überprüft. Auch bei diesem Test wird deutlich, dass die Methode wie vorhergesehen funktioniert und sich die Attributwerte innerhalb der definierten Grenzen bewegen. Zusätzlich wurde an diesem Punkt überprüft, ob eine Veränderung des Seedwertes die zugewiesenen Zufallszahlen verändert. Hierfür wurde der Seedwert um eins erhöht und die Simulation neu gestartet. Die nun zugewiesenen Attributwerte unterscheiden sich alle derart von den vorherigen Ergebnissen (mit einem anderen Seedwert), dass auch dieses Verhalten als korrekt eingestuft werden kann. Die Anweisungen aus den Zeilen 82 und 83 funktionieren ebenfalls wie vorhergesehen und der im Ereignisverwalter eingestellte Seedwert wird korrekt als benutzerdefiniertes Attribut an jedes einzelne erzeugte BE übergeben.

Ist die Checkbox nicht ausgewählt, so werden die entsprechenden Werte für die benutzerdefinierten Attribute für jedes BE aus der Tabelle *UserInput* ausgelesen. Dem Anwender wird somit ein Import seiner eigenen Daten ermöglicht und das Simulationsmodell verhält sich basierend auf diesen. Eine beispielhafte Datei mit dem Namen „Input.xlsx“ befindet sich im Ordner „Anhang“ auf dem dieser Arbeit beigelegten Datenträger und kann als Vorlage zum Import von Daten in die Tabelle *UserInput* genutzt werden. Diese Datei wurde zudem für die Verifikation der Importfunktion verwendet. Die Datei „Input.xlsx“ basiert auf den erzeugten Werten für einen Seedwert von 1 und wurde entsprechend aufbereitet, um die Importfunktion des Simulationsmodells zu illustrieren. Nachdem die Datei importiert worden ist, wird die Simulation im Einzelschritt ausgeführt, um die den BEs zugewiesenen Attributwerte mit den Parametern aus der Tabelle *UserInput* zu vergleichen. Zudem soll somit überprüft werden, ob mit jedem neu erzeugten BE die nächste Zeile an Parameterwerten aus der Tabelle übernommen und als Attribute verwendet werden. Nach einigen Durchläufen wird ersichtlich, dass dieses Verhalten wie vorhergesehen ausgeführt wird und die BEs die korrekten Werte als Attribute erhalten.

Die Anweisungen aus den Zeilen 85 bis 92 können auch über die „Beobachtung in der Animation“ verifiziert werden. Hierzu wird das Modell erneut in Einzelschritten ausgeführt und überprüft, ob das benutzerdefinierte Attribut *Lieferant* eines jeden BEs mit dem Namen der erzeugenden Quelle übereinstimmt. Kurz nach dem Eintritt in das Netzwerk wird hierzu das Menü des BEs aufgerufen und der Wert des Attributs *Lieferant* überprüft. Abbildung 3.7 zeigt einen Auszug aus dem Menü des zu sehenden BEs und verdeutlicht diesen Schritt der Verifikation.

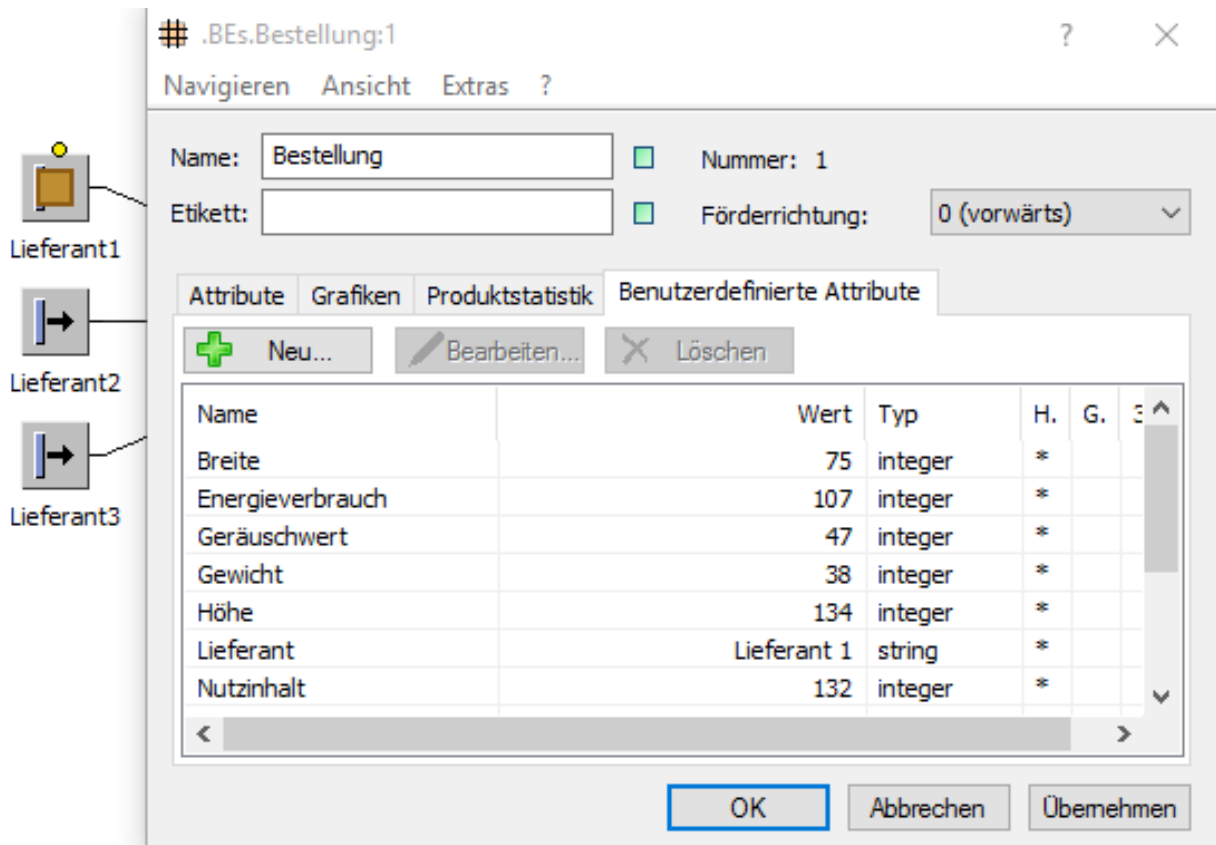


Abbildung 3.7: Verifikation des Attributs Lieferant

Anschließend wird das BE an die Einzelstation *Transport_OEM* weitergeleitet, welche je eine Eingangs- und Ausgangssteuerung besitzt, welche die Namen *TransportzeitBerechnen* und *Sortierung* besitzen.

Kurz vor dem Eintreffen des BEs (Option „Vor Aktionen“ im Menü der Einzelstation) wird die Eingangssteuerung ausgeführt, welche in Quellcode A.4 abgebildet ist. Diese Methode soll das Qualitätsattribut aller BEs prüfen. Ist der Parameter *Qualität* kleiner als 10, so ist dieses BE auszusortieren und wird zur Senke *Ausschuss_LD* weitergeleitet. Zum anderen soll die Bearbeitungszeit der Einzelstation basierend auf dem Gewicht und Volumen des ankommenden BEs verändert werden. Je schwerer und voluminöser ein BE ist, desto länger soll der nachgebildete Transport zum OEM dauern. Hierzu werden diese beiden Attribute miteinander multipliziert und ausgehend vom Ergebnis dieser Multiplikation die Bearbeitungszeit der Einzelstation modifiziert. Die einzustellenden Bearbeitungszeiten, sowie die jeweiligen Bedingungen sind in Tabelle 3.5 abgebildet. Auch hier wird die Technik „Beobachtung in der Animation“ angewendet und mittels Einzelschritt-Simulation die korrekte Einstellung der Bearbeitungszeit der Einzelstation geprüft. Abbildung 3.8 zeigt ein Beispiel mit den zugehörigen *Gewichts*-, *Höhen*- und *Breiten*attributen. Der Wert für das *Tiefen*attribut beträgt 75. Das Ergebnis der Rechnung $60\text{cm} * 74\text{cm} * 75\text{cm} * 75\text{kg}$ beträgt $24975000\text{cm}^3\text{kg}$ und entspricht somit einer Bearbeitungszeit von 3 Stunden. Dieses Vorgehen wird mehrere Male wiederholt, bis deutlich wird, dass die Zeitzuweisungen

korrekt durchgeführt werden und dieser Teil der Methode als verifiziert gilt.

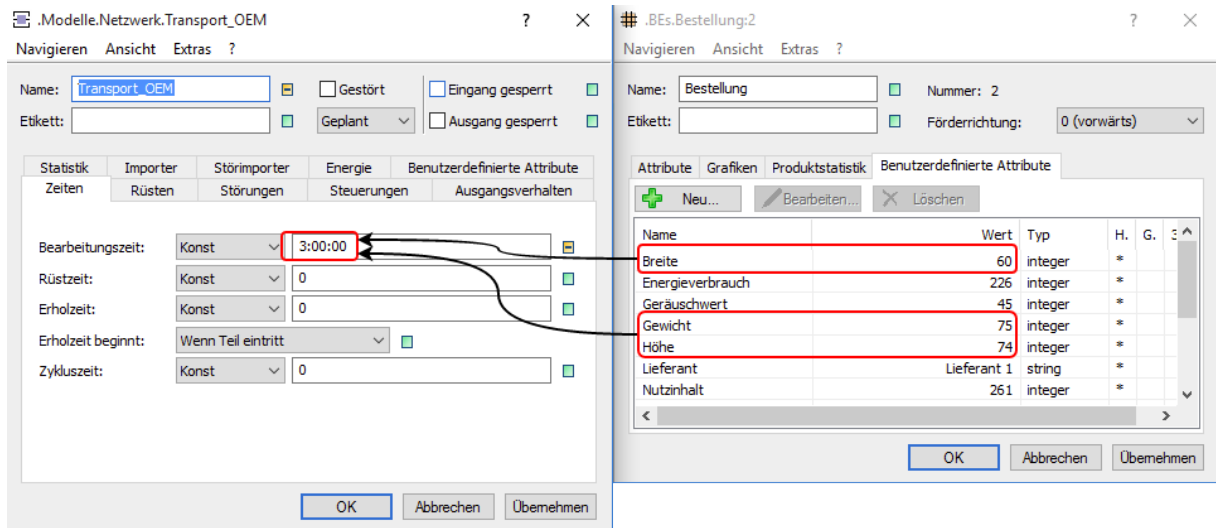


Abbildung 3.8: Verifikation der Bearbeitungszeit der Einzelstation *Transport_OEM*

Im letzten Abschnitt der Methode soll ein neues benutzerdefiniertes Attribut mit dem Namen *Transportzeit* für jedes BE erzeugt werden und den Wert der Bearbeitungszeit der Einzelstation *Transport_OEM* erhalten. Dies ist durch einen einfachen funktionalen Test beziehungsweise durch eine Überprüfung im Simulationsmodell verifiziert, womit die gesamte Methode als verifiziert gilt.

Nach dem Ablauf der Bearbeitungszeit wird die Ausgangssteuerung *Sortierung* ausgeführt, welche in Quellcode A.5 abgebildet ist. Hierbei sollen alle BEs ihrer Qualität entsprechend auf die folgenden Einzelstationen umgelagert werden. Die Intervalle für die Produktlinien sind in Tabelle 3.6 abgebildet. Mittels Einzelschritt-Simulation kann der Verlauf aller BEs langsam beobachtet und auf Korrektheit überprüft werden. Diese Beobachtung bestätigt die ordnungsgemäße Verteilung aller BEs auf die entsprechenden Einzelstationen der Produktlinien und verifiziert die Methode.

Nachdem die BEs an einer der Senken ankommen, wird die Eingangssteuerung *SchreibeParameter* ausgeführt, welche die benutzerdefinierten Attribute in die Tabelle *Output* überträgt und zudem das letzte benutzerdefinierte Attribut *Produktlinie* erzeugen soll. In Zeile 6 wird zudem zur Berechnung der Durchlaufzeit der Befehl `DLZ := Ereignisverwalter.AbsZeit - @.Produktionsdatum;` ausgeführt, welcher für die Berechnung der Durchlaufzeit eines BEs benötigt wird. Hierbei wird von der aktuellen Zeit im Simulationsmodell der Zeitwert subtrahiert, der in dem benutzerdefinierten Attribut *Produktionsdatum* hinterlegt ist. Das Ergebnis ist somit die Differenz aus Austritts- und Eintrittszeit eines BEs.

Im Anschluss wird der Standort des aktuellen BEs überprüft und basierend darauf ein Attribut mit dem Namen *Produktlinie* als *string*-Wert erzeugt. Für BEs, die aussortiert werden, wird in als *string*-Wert „Ausschuss“ eingetragen, da sie keiner Produktlinie angehören. Anschließend wird entsprechend der zugehörigen Produktlinie die Tabelle *Output*

aktualisiert. Diese enthält die Anzahl der erzeugten, der aussortierten und der zu verschiedenen Produktlinien zugehörigen BEs. Die Überprüfung dieses Abschnittes ist mittels „Beobachtung in der Animation“ trivial. Für die semantische Korrektheit ist es wichtig, dass die Aktualisierung der Tabelle *Output* erst mit der Ankunft der BEs bei einer der Senken stattfindet und nicht vorher. Erst mit dem Ankommen eines BEs an einer der Senken ist dieses tatsächlich im Besitz des OEM beziehungsweise aussortiert worden. So kann es vorkommen, dass die Addition der Werte „Premium“, „Advanced“, „Basic“ und Ausschuss aus der Tabelle *Output* in der Summe nicht dem Wert aus der Spalte „Erzeugte Einheiten“ entspricht. Dies deutet darauf hin, dass sich einige BEs noch auf dem Transportweg befinden und somit noch nicht beim OEM angekommen sind.

In den Zeilen 23 bis 43 werden die Werte der benutzerdefinierten Attribute in die Tabelle *Output* eingetragen, welche als Ergebnis des Data-Farming-Konzepts fungiert. Auch hier kann die Technik „Beobachtung in der Animation“ angewandt werden, um die in der Tabelle erzeugten Werte mit den Werten der BEs zu vergleichen. Dieses Verhalten funktioniert wie vorhergesehen und gilt nach Beobachtungen als verifiziert. Auch die Funktion der Anweisungen aus den Zeilen 37 bis 39 wurde in diesem Rahmen untersucht und verifiziert. In Zeile 40 wird der Befehl `Output.setzeHintergrundfarbeZeile(y, makeRGBValue(255,0,0));` ausgeführt, welcher den Zeilenhintergrund für alle BEs mit einer Qualität kleiner 10 rot färbt. Auch diese Funktion wurde mit der „Beobachtung in der Animation“ verifiziert.

Zum Schluss muss noch das Beenden des Simulationslaufs geprüft werden, sobald die letzte Zeile der Tabelle *UserInput* erreicht worden ist. Hierzu wird das Simulationsmodell mit der höchsten Geschwindigkeit ausgeführt und geprüft, ob es nach der Erzeugung des 37630. BEs beendet wird. Die Datei, die für den Import verwendet worden ist, weist exakt 37630 Zeilen auf, weshalb das BE mit der *id* 37631 keine weiteren Werte erhalten kann. Am 01.03.2101 um 09:30:21.4121 Uhr tritt dieser Fall ein und das Simulationsmodell wird mit einer entsprechenden Meldung beendet.

Die letzte Methode, die für den Ablauf des Simulationsmodells benötigt wird, ist die Methode *reset* (vgl. Quellcode A.7). Diese Methode dient hauptsächlich der Wiederherstellung des Ausgangszustandes des Modells. Hierzu werden alle Einträge in den Tabellen und Listen gelöscht. Dies geschieht mit dem Code in den Zeilen 3 bis 12. Die korrekte Funktion dieser Befehle ist durch eine Überprüfung der betroffenen Tabellen und Listen bestätigt. Die Anweisung aus Zeile 11 dient dazu, den gesamten Hintergrund der Tabelle *Output* weiß zu färben, damit die roten Zeilen aus dem vorherigen Simulationslauf entfernt werden. In Zeile 14 der Methode, wird die Bearbeitungszeit der Einzelstation *Transport_OEM* auf 2 Stunden gesetzt, welcher als Ausgangswert verwendet wird (vgl. Abschnitt 3.2).

Zum Schluss wird die Exportfunktion des Buttons mit der Aufschrift „Export .xlsx“ geprüft. Dieser Button ermöglicht dem Anwender eine schnelle Schaltfläche zum Export der erzeugten Daten. Die entsprechende Excel-Datei wird in dem gleichen Ordner auf dem verwendeten Rechner angelegt, indem sich die Datei, die das Simulationsmodell enthält,

befindet.

Im letzten Abschnitt dieses Kapitels werden die erzeugten Ergebnisse mit Hilfe des vorgestellten Simulationsmodells betrachtet.

3.4 Ergebnisse der Simulation

Nachdem die Verifikation des Simulationsmodells abgeschlossen ist, folgt eine Analyse der erzeugten Ergebnisse und Daten.

Da es sich bei der Simulationsstudie in dieser Arbeit um die Erstellung eines Konzepts handelt, ist die Erstellung von Experimentenplänen oder die Festlegung von Hypothesen (vgl. Abschnitt 2.1.1) nicht notwendig. Das Simulationsmodell bezieht sich auf kein reales System und stellt lediglich ein Konzept dar, welches eine Grundlage für weitere Erweiterungen und Anpassungen darstellen soll. In dieser Phase der Simulationsstudie werden lediglich die in dem jetzigen Zustand des Modells ermittelbaren Ergebnisse ausgewertet. Die unterschiedlich durchgeführten Experimente unterscheiden sich nur in der Wahl des Seedwertes, da dieser die Ausprägung aller benutzerdefinierten Attribute verändert und somit die Ergebnisse des Simulationslaufs beeinflusst. Um eine Vergleichbarkeit der Ergebnisse zu erhalten, wird in das Feld „Ende“ des Ereignisverwalters die Zeit „36500:00:00:00“ eingetragen. Diese Maßnahme stellt sicher, dass alle Simulationsdurchläufe den selben Simulationszeitraum (vgl. Abschnitt 2.1.1) aufweisen, was in diesem Fall etwa 86 Jahren entspricht.

Seedwert = 1 Das Simulationsmodell wird zunächst mit dem voreingestellten Seedwert von 1 gestartet und es wird kein eigener Input eingelesen. Nachdem der Simulationsdurchlauf beendet ist, zeigt der Ereignisverwalter das Datum „Di, 01.03.2101 03:30:21.4121“ an. Beim Öffnen der Tabelle *Statistiktabelle* wird deutlich, dass in dem Simulationszeitraum 37630 BEs erzeugt worden sind. 11387 BEs entfallen hierbei auf die Produktlinie *Premium*, 11192 auf die Produktlinie *Advanced* und 11338 auf die Produktlinie *Basic*. 3713 BEs haben die Qualitätskontrolle des Logistikdienstleisters nicht bestanden und wurden als *Ausschuss* deklariert.

Abbildung 3.9 zeigt einen Ausschnitt aus den ersten Zeilen der Tabelle Output, aus der qualitativ ungenügende BEs direkt hervorgehen. Die Durchlaufzeit (und auch die Transportzeit) beträgt bei BEs, die vom Logistikdienstleister als Ausschuss markiert werden stets „0“, da diese das Simulationsmodell mit der Ankunft beim Transportdienstleister verlassen. Hierbei wird angenommen, dass der Logistikdienstleister stets am Warenausgang der First-Tier-Lieferanten auf die BEs wartet. Die Anzahl der insgesamt erzeugten Zeilen kann aus der Anzahl der insgesamt im Simulationszeitraum erzeugten BEs (hier: 37630) abgeleitet werden, wobei diese Zahl leicht variieren kann, falls sich noch BEs zum Ende des gewählten Simulationszeitraumes innerhalb der SC befinden.

In Zeile 8 der abgebildeten Tabelle besitzt das entsprechende BE (mit der ID 8) eine

Durchlaufzeit von „4:39:45.2018“. Diese Durchlaufzeit ist von Interesse, da sie sich von den meisten anderen Durchlaufzeiten aus der Tabelle unterscheidet. Die meisten erzeugten BEs besitzen eine Durchlaufzeit, die einer exakten Stundenangabe entspricht, wie zum Beispiel in den Zeilen 2 bis 7. Das BE in Zeile 8 besitzt jedoch eine Durchlaufzeit mit Minuten- und Sekundenanteilen. Startet man die Simulation neu und betrachtet diese Schritt für Schritt, so wird deutlich, dass es hierbei zu einer Blockierung der SC kommt. Der Vorgänger mit der ID 7 kommt am „09.01.2015 15:14:18.1713“ an der Einzelstation *LD_Abw_Advanced* an und verlässt diese um „09.01.2015 22:14:18.1713“. Hierbei wird der Mechanismus aktiviert, der dafür sorgt, dass Lieferabweichungen abgebildet werden können (vgl. Abschnitt 3.1.6). Es kommt in dem betrachteten Fall zu einer Verzögerung von 7 Stunden für das BE mit der ID 7. Dadurch, dass das BE mit der ID 7 eine Qualität von 56 und das BE mit der ID 8 eine Qualität von 69 aufweist, entsteht beim Logistikdienstleister ein Stau, da eine Einzelstation nur ein BE gleichzeitig aufnehmen kann. Durch diesen Stau und durch die Normalverteilungen der unterschiedlichen Quellen entstehen die Durchlaufzeiten für die BEs mit den IDs 8 und 9, welche Minuten- und Sekundenanteile besitzen. Dieser Fall tritt in der Regel nur dann auf, wenn zwei BEs mit der gleichen Produktlinienzugehörigkeit hintereinander produziert werden und der erste von beiden einer langen Lieferabweichung unterworfen wird. Dies erklärt auch die nicht einheitliche Reihenfolge der Spalte „BE ID“ mit den Zeilennummern, da in der Zwischenzeit BEs anderer Produktlinien schneller als die betroffenen BEs zum OEM gelangen können (vgl. Zeile 8 und 9 in Abbildung 3.9). Da jedoch im Rahmen der Systemanalyse (vgl. Abschnitt 3.1.5) die Annahme getroffen wurde, dass der OEM je Produktlinie nur ein BE gleichzeitig annehmen kann, ist dieser Sachverhalt zu vernachlässigen.

Anhand der Ergebnisse kann der Simulationsanwender ermitteln, welche BEs eine zu niedrige Qualität aufweisen und wie hoch der Anteil dieser ist. Da in dem vorliegenden Szenario, der Verständlichkeit halber, bewusst günstige Intervalle (z.B. Qualitätsintervall von 0 bis 100) gewählt worden sind, sind die Ergebnisse trivial. Für den betrachteten Simulationsdurchlauf mit einem Seedwert von 1 wurden 3713 Ausschüsse bei insgesamt 37630 erzeugten BEs ermittelt, was etwa 10% der erzeugten BEs entspricht. Da zur zufälligen Zuweisung des Parameters *Qualität* eine Gleichverteilung genutzt worden ist und BEs mit einer *Qualität* zwischen 0 und 10 (von Maximal 100) als Ausschuss deklariert werden, ist dieses Ergebnis nicht verwunderlich. Es ist zu beachten, dass, wie bereits in Abschnitt 3.1.5 erwähnt worden ist, es dem Anwender überlassen ist, ein seinen Bedürfnissen entsprechendes Zuordnungssystem zu entwickeln. Die Ermittlung der Qualität eines Produkts ist zudem komplexer als in dieser Simulationsstudie vorgestellt worden ist und je nach Unternehmen oder Betrachtungsgegenstand unterschiedlich.

02.01.2015 05:57:35.7064								
string	datetime 1	datetime 2	time 3	string 4	integer 5	integer 6	integer 7	integer 8
	AnkunftOEM	Produktionsdatum	Durchlaufzeit	Lieferant	Höhe	Breite	Tiefe	Volumen
1	02.01.2015 05:57:35.7064	02.01.2015 05:57:35.7064	0.0000	Lieferant 1	134	78	81	412050
2	03.01.2015 03:55:31.2264	03.01.2015 00:55:31.2264	3:00:00.0000	Lieferant 1	74	60	75	333000
3	05.01.2015 14:20:56.6785	05.01.2015 12:20:56.6785	2:00:00.0000	Lieferant 1	130	46	50	299000
4	06.01.2015 14:32:06.5230	06.01.2015 09:32:06.5230	5:00:00.0000	Lieferant 2	165	65	67	718575
5	08.01.2015 13:44:09.1581	08.01.2015 05:44:09.1581	8:00:00.0000	Lieferant 3	119	61	57	413763
6	09.01.2015 03:28:42.1074	08.01.2015 23:28:42.1074	4:00:00.0000	Lieferant 1	196	57	46	513912
7	09.01.2015 22:14:18.1713	09.01.2015 12:14:18.1713	10:00:00.0000	Lieferant 2	83	53	61	268339
8	10.01.2015 00:14:18.1713	09.01.2015 19:34:32.9695	4:39:45.2018	Lieferant 2	123	67	43	354363
9	10.01.2015 04:14:18.1713	09.01.2015 17:17:31.2573	10:56:46.9140	Lieferant 1	178	60	53	566040
10	10.01.2015 09:41:57.2986	10.01.2015 02:41:57.2986	7:00:00.0000	Lieferant 1	190	54	42	430920
11	11.01.2015 19:32:01.1957	11.01.2015 15:32:01.1957	4:00:00.0000	Lieferant 2	107	56	71	425432
12	12.01.2015 02:42:04.5594	11.01.2015 22:42:04.5594	4:00:00.0000	Lieferant 3	116	71	50	411800
13	13.01.2015 03:13:42.8748	12.01.2015 23:13:42.8748	4:00:00.0000	Lieferant 1	167	63	71	746991
14	13.01.2015 11:46:27.9706	13.01.2015 04:46:27.9706	7:00:00.0000	Lieferant 2	191	45	47	403965
15	15.01.2015 05:05:40.6327	15.01.2015 03:05:40.6327	2:00:00.0000	Lieferant 2	82	43	57	200982
16	16.01.2015 01:07:00.3473	16.01.2015 01:07:00.3473	0.0000	Lieferant 1	134	78	82	543504
17	16.01.2015 20:49:28.1309	16.01.2015 17:49:28.1309	3:00:00.0000	Lieferant 2	159	63	48	480816
18	18.01.2015 07:02:50.8792	18.01.2015 04:02:50.8792	3:00:00.0000	Lieferant 1	92	74	60	408480
19	18.01.2015 14:49:05.8268	18.01.2015 08:49:05.8268	6:00:00.0000	Lieferant 3	150	43	62	399900
20	20.01.2015 10:23:49.0686	20.01.2015 08:23:49.0686	2:00:00.0000	Lieferant 1	191	50	40	382000
21	22.01.2015 13:03:26.7002	22.01.2015 09:03:26.7002	4:00:00.0000	Lieferant 2	182	79	76	1092728
22	22.01.2015 18:03:26.7002	22.01.2015 12:18:50.1648	5:44:36.5354	Lieferant 3	118	48	50	283200
23	23.01.2015 11:12:40.9545	23.01.2015 07:12:40.9545	4:00:00.0000	Lieferant 1	193	43	76	630724
24	23.01.2015 13:22:51.1107	23.01.2015 10:22:51.1107	3:00:00.0000	Lieferant 1	90	55	71	351450
25	24.01.2015 07:36:44.7359	24.01.2015 00:36:44.7359	7:00:00.0000	Lieferant 1	153	49	41	307377
26	24.01.2015 11:14:45.4246	24.01.2015 04:14:45.4246	7:00:00.0000	Lieferant 3	195	73	79	1124565
27	24.01.2015 11:14:45.4246	24.01.2015 06:09:28.4698	5:05:16.9548	Lieferant 3	146	49	42	300468
28	24.01.2015 22:58:08.5806	24.01.2015 19:58:08.5806	3:00:00.0000	Lieferant 3	126	67	54	455868
29	25.01.2015 06:26:26.5197	25.01.2015 03:26:26.5197	3:00:00.0000	Lieferant 2	164	46	58	437552
30	25.01.2015 22:12:07.3136	25.01.2015 19:12:07.3136	3:00:00.0000	Lieferant 1	191	53	58	587134
31	27.01.2015 09:11:03.2216	27.01.2015 09:11:03.2216	0.0000	Lieferant 3	197	86	79	871528
32	28.01.2015 09:18:09.1861	28.01.2015 07:18:09.1861	2:00:00.0000	Lieferant 3	142	51	76	550392
33	29.01.2015 03:00:26.2005	29.01.2015 00:00:26.2005	3:00:00.0000	Lieferant 1	105	65	75	511875

Abbildung 3.9: Ergebnisse der Tabelle Output mit einem Seedwert von 1

Im Rahmen einer Anpassung und Erweiterung des vorgestellten Konzepts wird dem Anwender geraten, dass die Zusammensetzung des Attributs *Qualität* aus unterschiedlichen (bekannten) Parametern erfolgen sollte und somit weniger vom Zufall abhängig sein sollte. Die Ermittlung dieser Parameter zur Zusammensetzung ist aus externer Sicht nur schwer bis gar nicht möglich, weshalb im Rahmen der Konzepterstellung auf eine Skala von 0 bis 100 ausgewichen worden ist (vgl. Abschnitt 3.1.4), was im Rahmen der Vorführung jedoch kein Problem darstellt.

Zusätzlich wird dem Anwender geraten, korrekte Transportzeiten in das Modell einzupflegen. In Abschnitt 3.1.6 wurden Annahmen getroffen, um auch diesen Aspekt des Simulationsmodells ohne das Vorhandensein von Echtdateien vorstellen zu können. Die Transportzeit wird basierend auf dem Ergebnis der Multiplikation der benutzerdefinierten Attribute *Gewicht*, *Höhe*, *Breite* und *Tiefe* festgelegt. In der Realität können diese Parameter einen Einfluss auf die Transportzeit von Waren haben, dieser ist jedoch auch von weiteren Faktoren abhängig. Während Echtdateien für die Parameter *Höhe*, *Breite* und *Tiefe* vorhanden sind, wurde die Festlegung der benötigten Lieferzeiten willkürlich (und so kurz wie möglich) gewählt. Dem Simulationsanwender liegen die Transportzeiten der Lieferanten vor, welche durch eine Anpassung der Methode *TransportzeitBerechnen* in das Simulationsmodell eingefügt werden können.

Die Anpassung der Normalverteilungen der First-Tier-Lieferanten ist ein weiterer Aspekt, der der Aufmerksamkeit des Anwenders bedarf. Um die Durchlaufzeiten realitätsgetreu simulieren zu können, muss der Anwender die statistische Verteilung an die Gegebenheiten

seiner SC anpassen. Auch diese Daten liegen dem Anwender vor.

Das Simulationsmodell ermöglicht dem Anwender die Betrachtung von vielen verschiedenen Parameterkombinationen und deren Auswirkung auf Attribute wie die Transportzeit, die zugehörige Produktlinie und die gesamte Durchlaufzeit. Zudem wird dem Anwender eine Übersicht der unterschiedlichen Ereignisse (vgl. Abschnitt 2.2), wie zum Beispiel das Produktionsdatum, das Ankunftsdatum und den ursprünglichen Lieferanten zur Verfügung gestellt, die bei der Verbesserung des realen SC von Nutzen sind.

Im nächsten Schritt wird der Seedwert willkürlich auf 21 erhöht und die Simulation für den gleichen Simulationszeitraum durchgeführt.

Seedwert = 21 Der zweite Simulationsdurchlauf wird über den gleichen Simulationszeitraum durchgeführt. Somit beträgt der Simulationszeitraum erneut „36500:00:00:00“. Da jegliche Zufallszahlen und statischen Verteilungen abhängig von dem im Ereignisverwalter eingestellten Seedwert sind (vgl. Abschnitt 2.2.1), werden in diesem Szenario unterschiedliche Ergebnisse als mit einem Seedwert von 1 ermittelt.

In Abbildung 3.10 ist erneut die Tabelle *Output* abgebildet, welche sich in diesem Fall von den Ergebnissen aus Abbildung 3.9 unterscheidet. In den abgebildeten Zeilen sind weniger qualitativ ungenügende BEs vorhanden. Während in Abbildung 3.9 3 BEs abgebildet sind, die rot markiert sind, ist es in Abbildung 3.10 nur ein BE. Betrachtet man hierzu die Tabelle *Statistiktabelle*, so werden die Unterschiede zum vorherigen Szenario deutlich. Es wird zudem deutlich, dass die generelle Verteilung der Parameterkombinationen der BEs unterschiedlich ist.

Bei einem Seedwert von 21 wurden 37631 Einheiten insgesamt erzeugt, während 11312 BEs der Produktlinie Premium, 11145 BEs der Produktlinie Advanced und 11458 BEs der Produktlinie Basic zugeordnet worden sind. Es wurden insgesamt 3715 BEs als Ausschuss markiert und nicht an den OEM weitergeleitet. Somit wird deutlich, dass auch in diesem Fall etwa 10% der Bestellungen qualitativ ungenügend sind. Außerdem wird deutlich, dass die Anzahl der erzeugten BEs konsistent geblieben ist. Dies rührt daher, dass der verwendete Seedwert keinen Einfluss auf die Erzeugungsdauer von BEs besitzt und dieses Attribut somit stets gleich bleibt (vgl. Abschnitt 3.1.6). Es steht dem Anwender jedoch frei, auch diesen Parameter des Simulationsmodells an seine Bedürfnisse anzupassen. Da die Attribute *Abstand* und *Start* der verschiedenen Quellen nach dem Eingabeschema „ μ , σ , untere Schranke, obere Schranke“ aufgebaut sind, ist es dem Anwender möglich, diese Parameter basierend auf einer statistischen Verteilungsfunktion festzulegen und diese Funktion abhängig vom Seedwert zu machen.

integer 1	integer 2	datetime 2	datetime 3	time 4	string 5	integer 6	integer 7	integer 8
string	BE ID	AnkunftOEM	Produktionsdatum	Durchlaufzeit	Lieferant	Höhe	Breite	Tiefe
1	1	02.01.2015 20:38:41.2893	02.01.2015 16:38:41.2893	4:00:00.0000	Lieferant 2	117	79	78
2	2	03.01.2015 21:37:10.3244	03.01.2015 19:37:10.3244	2:00:00.0000	Lieferant 1	86	41	64
3	3	04.01.2015 18:43:43.5803	04.01.2015 15:43:43.5803	3:00:00.0000	Lieferant 3	81	62	75
4	4	04.01.2015 22:34:55.5751	04.01.2015 19:34:55.5751	3:00:00.0000	Lieferant 2	78	54	69
5	5	06.01.2015 20:16:55.1565	06.01.2015 16:16:55.1565	4:00:00.0000	Lieferant 1	129	70	57
6	6	07.01.2015 19:25:28.0243	07.01.2015 13:25:28.0243	6:00:00.0000	Lieferant 2	74	46	72
7	7	10.01.2015 00:13:14.6042	09.01.2015 12:13:14.6042	12:00:00.0000	Lieferant 2	171	45	72
8	8	10.01.2015 01:44:31.8505	09.01.2015 19:44:31.8505	6:00:00.0000	Lieferant 2	131	79	59
9	9	10.01.2015 08:13:30.3386	10.01.2015 04:13:30.3386	4:00:00.0000	Lieferant 1	172	77	68
10	10	10.01.2015 09:07:42.2857	10.01.2015 09:07:42.2857	0:00:00.0000	Lieferant 3	170	41	52
11	11	10.01.2015 14:12:31.0086	10.01.2015 10:12:31.0086	4:00:00.0000	Lieferant 1	123	47	55
12	12	11.01.2015 13:37:41.2436	11.01.2015 11:37:41.2436	2:00:00.0000	Lieferant 1	144	45	59
13	13	12.01.2015 20:30:05.8655	12.01.2015 13:30:05.8655	7:00:00.0000	Lieferant 1	119	48	72
14	14	14.01.2015 01:05:55.1698	13.01.2015 23:05:55.1698	2:00:00.0000	Lieferant 1	181	48	44
15	15	14.01.2015 06:55:07.4312	14.01.2015 03:55:07.4312	3:00:00.0000	Lieferant 2	85	76	63
16	16	15.01.2015 13:02:20.1339	15.01.2015 07:02:20.1339	6:00:00.0000	Lieferant 3	118	64	64
17	17	16.01.2015 04:02:14.7125	16.01.2015 01:02:14.7125	3:00:00.0000	Lieferant 1	143	54	51
18	18	18.01.2015 08:23:14.8867	18.01.2015 03:23:14.8867	5:00:00.0000	Lieferant 1	193	58	55
19	19	18.01.2015 18:17:02.7983	18.01.2015 15:17:02.7983	3:00:00.0000	Lieferant 2	144	55	60
20	20	20.01.2015 09:13:35.0190	20.01.2015 06:13:35.0190	3:00:00.0000	Lieferant 1	159	66	67
21	21	22.01.2015 08:36:47.1233	22.01.2015 02:36:47.1233	6:00:00.0000	Lieferant 2	108	70	71
22	22	22.01.2015 09:48:41.4953	22.01.2015 06:48:41.4953	3:00:00.0000	Lieferant 1	164	55	43
23	23	23.01.2015 04:45:48.1572	22.01.2015 21:45:48.1572	7:00:00.0000	Lieferant 3	155	44	70
24	24	25.01.2015 03:02:11.2187	24.01.2015 23:02:11.2187	4:00:00.0000	Lieferant 1	92	58	64
25	25	25.01.2015 15:39:29.5691	25.01.2015 11:39:29.5691	4:00:00.0000	Lieferant 2	124	63	67
26	26	25.01.2015 18:39:29.5691	25.01.2015 12:18:14.4120	6:21:15.1571	Lieferant 1	137	57	42
27	27	26.01.2015 21:05:18.4544	26.01.2015 19:05:18.4544	2:00:00.0000	Lieferant 1	162	78	54
28	28	28.01.2015 08:58:57.4834	28.01.2015 05:58:57.4834	3:00:00.0000	Lieferant 3	154	59	63
29	29	29.01.2015 09:21:47.4019	29.01.2015 02:21:47.4019	7:00:00.0000	Lieferant 1	87	62	55
30	30	29.01.2015 10:21:47.4019	29.01.2015 03:00:53.4173	7:20:53.9846	Lieferant 1	194	67	44
31	31	31.01.2015 09:36:16.6662	31.01.2015 04:36:16.6662	5:00:00.0000	Lieferant 2	111	41	68
32	32	31.01.2015 17:07:29.6130	31.01.2015 11:07:29.6130	6:00:00.0000	Lieferant 1	162	51	59
33	33	02.02.2015 08:11:52.8789	02.02.2015 05:11:52.8789	3:00:00.0000	Lieferant 3	184	41	58

Abbildung 3.10: Ergebnisse der Tabelle Output mit einem Seedwert von 21

Weitere Seedwerte werden nicht betrachtet, da zwei Beispiele den Einfluss dieser auf das Simulationsmodell hinreichend genau wiedergeben. Die Veränderung des Seedwertes hat keinen Einfluss auf die insgesamt erzeugte Menge und betrifft lediglich die Parameterkombinationen und Verteilung der erzeugten BEs. Durch die weitere Verknüpfung der Parameter kann eine noch komplexere Datenlandschaft erzeugt und mit entsprechenden Techniken analysiert werden.

Somit kommt die Betrachtung der Ergebnisse zum Schluss, dass der Entwurf eines Konzepts für Data-Farming-Aktivitäten erfolgreich durchgeführt worden ist und das Simulationsmodell eine große Datenlandschaft erzeugt. Wird die Begrenzung für den zu betrachtenden Simulationszeitraum aufgehoben, kann eine größere Datenlandschaft erzeugt werden. Die Ergebnisse für beide Simulationsdurchläufe wurden als „.xlxs“-Datei exportiert und befinden sich auf dem Datenträger, der dieser Arbeit beiliegt.

4 Zusammenfassung und Fazit

Gegenstand der vorliegenden Arbeit ist es, ein Data-Farming-Konzept in „Tecnomatix Plant Simulation“ zu erstellen und dieses zu verifizieren. Da „Tecnomatix Plant Simulation“ hauptsächlich Anwendung in der Logistik findet, wird hierbei eine SC zwischen First-Tier-Lieferanten und einem OEM betrachtet. Der Transportvorgang der Waren wird durch einen Transportdienstleister durchgeführt. Für die Umsetzung dieses Untersuchungsgegenstandes müssen passende Elemente ausgesucht und miteinander in Relation gebracht werden. Zudem muss die Einpflegung von Daten aus der Realität ermöglicht werden, da es sich um eine fiktive SC handelt, welche nur zu Illustrationszwecken dient. Außerdem müssen die Grundsätze des Data Farmings beachtet und in das Modell eingebaut werden. Eines dieser Grundsätze sind die „3 F des Data Farmings“ nach Sanchez [2014].

Das vorrangige Ziel der Umsetzung ist die Erzeugung einer möglichst großen Datenlandschaft, welche in nachfolgenden Schritten mit Hilfe von Techniken auf neue Erkenntnisse und Informationen untersucht werden kann. Bisher fand die Umsetzung des Data Farmings in Spezialesimulatoren, wie z.B. „SANDIS“, „Mana“ oder „Pythagoras“ statt. Bei „Tecnomatix Plant Simulation“ handelt es sich hingegen um ein Simulationswerkzeug, welches einen Kompromiss zwischen Anpassungsmöglichkeit und Benutzerfreundlichkeit eingeht. Somit ermöglicht das vorgestellte Konzept erste Data-Farming-Experimente für Anwender, die nicht genügend Fachwissen besitzen um komplexe Simulatoren zu bedienen. Zudem erspart dieses Konzept den Einsatz eines externen Dienstleisters, der Data-Farming-Experimente für das eigene Unternehmen vornimmt.

Nachdem das Modell erstellt worden ist, wird eine Verifikation des erstellten Modells durchgeführt. Hierbei wird die Wichtigkeit der begleitende V & V nach Rabe et al. [2008] deutlich, da die Suche und Korrektur am Ende einer Simulationsstudie eine große Zeit in Anspruch nimmt. Der benötigte Zeitaufwand ist abhängig von der Komplexität des zu untersuchenden Simulationsmodells. Falsche Ergebnisse können zu falschen Entscheidungen führen, welche die Systeme aus der Realität negativ beeinflussen können.

Diese Arbeit ist ein weiterer Beweis für die Flexibilität und Bedeutung der Simulation in der heutigen Industrie. Im Rahmen der „Industrie 4.0“ und „Big Data“ wird die Bedeutung in den kommenden Jahren weiter steigen. Im Zuge des digitalen Zeitalters werden Informationen für Unternehmen immer wichtiger und die Erlangung dieser schwerer. Um die bestehenden Prozesse möglichst nicht zu gefährden, werden Simulationsmodelle erstellt und analysiert. Gerade um Strategien für Szenarien zu entwickeln, die bisher noch nicht eingetreten sind oder deren Eintreten als unwahrscheinlich gilt, kommen Simulations-

werkzeuge zum Einsatz. In Verbindung mit dem Data Farming und dem anschließenden Data Mining versuchen Unternehmen ihre Entscheidungen und Aktionen bestmöglich vorausszusehen und gegebenenfalls Notfallpläne zu entwickeln. Die Globalität, Komplexität und voranschreitende Entwicklung der heutigen SCs führt dazu, dass es oft zu Störungen innerhalb dieser kommt, weshalb es im Rahmen des SCM, der Supply Chain Resilience und der Supply Chain Flexibility von Bedeutung ist, diese Risiken zu identifizieren und aktiv zu bekämpfen.

Die Ergebnisse der vorliegenden Arbeit haben gezeigt, dass die Umsetzung eines Data-Farming-Konzepts in „Tecnomatix Plant Simulation“ möglich ist. Die Auswahl verschiedener Seedwerte sorgt dafür, dass bei jedem erneuten Simulationsdurchlauf stets neue Parameterkombinationen erzeugt werden und die maximal mögliche Datenlandschaft immer besser entdeckt wird. Parameterintervalle spannen in diesem Zusammenhang die maximale Datenlandschaft auf. Die Qualität der Ergebnisse dieses Konzepts ist stets abhängig von der Qualität der Eingangsdaten. Daher ist stets der Anwender Verantwortlich für die sachgerechte Implementierung seiner vorhandenen Daten. Das Konzept ermöglicht und erleichtert ihm diesen Prozess lediglich und verdeutlicht die Eigenschaften des Data Farmings anhand eines fiktiven Szenarios und stellt somit ein Grundgerüst für Data-Farming-Experimente dar.

A Quellcodes

```
1 is
2   zufallHöhe: integer;
3   zufallBreite: integer;
4   zufallTiefe: integer;
5   zufallEnergieverbrauch: integer;
6   zufallNutzinhalt: integer;
7   zufallGeräuschwert: integer;
8   zufallGewicht: integer;
9   zufallQualität: integer;
10  seedwert: integer;
11 do
12   Statistiktabelle["Erzeugte Einheiten",1] := Statistiktabelle["Erzeugte
13     Einheiten",1]+1;
14   @.erzeugeAttr("Produktionsdatum","datetime");
15   @.Produktionsdatum := Ereignisverwalter.AbsZeit;
16
17   if Random_Seed.Wert = true then
18     Ereignisverwalter.InkrementiereZufallszahlenVarianteBeiReset := true;
19   else
20     Ereignisverwalter.InkrementiereZufallszahlenVarianteBeiReset := false;
21   end;
22
23   if Random_Check.Wert = true then
24     seedwert := Ereignisverwalter.ZufallszahlenVariante;
25     zufallHöhe := z_gleich(seedwert,70,200);
26     zufallBreite := z_gleich(seedwert,40,80);
27     zufallTiefe := z_gleich(seedwert,40,80);
28     zufallEnergieverbrauch := z_gleich(seedwert,50,240);
29     zufallNutzinhalt := z_gleich(seedwert,110,360);
30     zufallGeräuschwert := z_gleich(seedwert,20,60);
31     zufallGewicht := z_gleich(seedwert,20,90);
32     zufallQualität := z_gleich(seedwert,0,100);
33
34     @.erzeugeAttr("Höhe","integer");
35     @.Höhe := zufallHöhe;
36     @.erzeugeAttr("Breite","integer");
37     @.Breite := zufallBreite;
38     @.erzeugeAttr("Tiefe","integer");
39     @.Tiefe := zufallTiefe;
40     @.erzeugeAttr("Volumen","integer");
41     @.Volumen := @.Höhe*@.Tiefe*@.Breite;
42     @.erzeugeAttr("Energieverbrauch","integer");
43     @.Energieverbrauch := zufallEnergieverbrauch;
44     @.erzeugeAttr("Nutzinhalt","integer");
45     @.Nutzinhalt := zufallNutzinhalt;
46     @.erzeugeAttr("Geräuschwert","integer");
47     @.Geräuschwert := zufallGeräuschwert;
48     @.erzeugeAttr("Gewicht","integer");
49     @.Gewicht := zufallGewicht;
50     @.erzeugeAttr("Qualität","integer");
```



```
51  @.Qualität := zufallQualität;
52
53  else
54    @.erzeugeAttr("Höhe","integer");
55    @.Höhe := userInput["Höhe",@.id];
56    @.erzeugeAttr("Breite","integer");
57    @.Breite := userInput["Breite",@.id];
58    @.erzeugeAttr("Tiefe","integer");
59    @.Tiefe := userInput["Tiefe",@.id];
60    @.erzeugeAttr("Volumen","integer");
61    @.Volumen := userInput["Volumen",@.id];
62    @.erzeugeAttr("Energieverbrauch","integer");
63    @.Energieverbrauch := userInput["Energieverbrauch",@.id];
64    @.erzeugeAttr("Nutzinhalt","integer");
65    @.Nutzinhalt := userInput["Nutzinhalt",@.id];
66    @.erzeugeAttr("Geräuschwert","integer");
67    @.Geräuschwert := userInput["Geräuschwert",@.id];
68    @.erzeugeAttr("Gewicht","integer");
69    @.Gewicht := userInput["Gewicht",@.id];
70    @.erzeugeAttr("Qualität","integer");
71    @.Qualität := userInput["Qualität",@.id];
72  end;
73
74  @.erzeugeAttr("Seedwert","integer");
75  @.Seedwert := Ereignisverwalter.ZufallszahlenVariante;
76
77  @.erzeugeAttr("Lieferant","string");
78  if @.Standort = Lieferant1 then
79    @.Lieferant := "Lieferant 1";
80  elseif @.Standort=Lieferant2 then
81    @.Lieferant := "Lieferant 2";
82  else
83    @.Lieferant := "Lieferant 3";
84  end;
85  end;
```

Quellcode A.1: Methode Attributerzeugung

```
1  is
2  do
3    userInput.leseExcelDatei("Input.xlsx");
4  end;
```

Quellcode A.2: Methode Import.xlsx

```
1  is
2  do
3    Output.schreibeExcelDatei("Output.xlsx");
4  end;
```

Quellcode A.3: Methode Export.xlsx

```

1 is
2 do
3   if @.Qualität < 10 then
4     @.umlagern(Ausschuss_LD);
5     return;
6   end;
7
8   if @.Gewicht*@.Volumen <= 20000000 then
9     Transport_OEM.Bearbeitungszeit := str_to_time("2:00:00");
10  elseif @.Gewicht*@.Volumen <= 40000000 then
11    Transport_OEM.Bearbeitungszeit := str_to_time("3:00:00");
12  elseif @.Gewicht*@.Volumen <= 60000000 then
13    Transport_OEM.Bearbeitungszeit := str_to_time("4:00:00");
14  elseif @.Gewicht*@.Volumen <= 80000000 then
15    Transport_OEM.Bearbeitungszeit := str_to_time("5:00:00");
16  else
17    Transport_OEM.Bearbeitungszeit := str_to_time("6:00:00");
18  end;
19
20  @.erzeugeAttr("Transportzeit","time");
21  @.Transportzeit := Transport_OEM.Bearbeitungszeit;
22
23 end;

```

Quellcode A.4: Methode TransportzeitBerechnen

```

1 is
2 do
3   if @.Qualität < 40 then
4     @.umlagern(LD_Abw__Basic);
5   elseif @.Qualität < 70 then
6     @.umlagern(LD_Abw__Advanced);
7   else
8     @.umlagern(LD_Abw__Premium);
9   end;
10 end;

```

Quellcode A.5: Methode Sortierung

```

1 is
2   y: integer;
3   DLZ : time;
4 do
5   y := Output.YDim + 1;
6   DLZ := Ereignisverwalter.AbsZeit - @.Produktionsdatum;
7
8   @.erzeugeAttr("Produktlinie","string");
9   if @.Standort = WE_Premium then
10    @.Produktlinie := "Premium";
11    Statistiktabelle["Premium",1] := Statistiktabelle["Premium",1] + 1;
12  elseif @.Standort = WE_Advanced then
13    @.Produktlinie := "Advanced";
14    Statistiktabelle["Advanced",1] := Statistiktabelle["Advanced",1] + 1;
15  elseif @.Standort = WE_Basic then
16    @.Produktlinie := "Basic";
17    Statistiktabelle["Basic",1] := Statistiktabelle["Basic",1] + 1;
18  else
19    @.Produktlinie := "Ausschuss";
20    Statistiktabelle["Ausschuss",1] := Statistiktabelle["Ausschuss",1] + 1;

```

```

21 end;
22
23 Output ["AnkunftOEM",y] := Ereignisverwalter.AbsZeit;
24 Output ["Produktionsdatum",y] := @.Produktionsdatum;
25 Output ["Durchlaufzeit",y] := DLZ;
26 Output ["Lieferant",y] := @.Lieferant;
27 Output ["Höhe",y] := @.Höhe;
28 Output ["Breite",y] := @.Breite;
29 Output ["Tiefe",y] := @.Tiefe;
30 Output ["Volumen",y] := @.Volumen;
31 Output ["Energieverbrauch",y] := @.Energieverbrauch;
32 Output ["Nutzinhalt",y] := @.Nutzinhalt;
33 Output ["Geräuschwert",y] := @.Geräuschwert;
34 Output ["Gewicht",y] := @.Gewicht;
35 Output ["Qualität",y] := @.Qualität;
36 if @.Qualität >= 10 then
37     Output ["Transportzeit",y] := @.Transportzeit;
38 else
39     Output ["Transportzeit",y] := str_to_time("0:00:00");
40     Output.setzeHintergrundfarbeZeile(y,makeRGBValue(255,0,0));
41 end;
42 Output ["Produktlinie",y] := @.Produktlinie;
43 Output ["Seedwert",y] := @.Seedwert;
44
45 if @.id = userInput.YDim then
46     print "Keine weiteren Eingabeparameter mehr vorhanden. Simulation
47         wird beendet.";
48     eventController.stop;
49 end;

```

Quellcode A.6: Methode SchreibeParameter

```

1 is
2 do
3     Liste_LD_Abw_Premium.löschen;
4     Liste_LD_Abw_Advanced.löschen;
5     Liste_LD_Abw_Basic.löschen;
6     Liste_WE_Premium.löschen;
7     Liste_WE_Advanced.löschen;
8     Liste_WE_Basic.löschen;
9
10    Output.löschen;
11    Output.Hintergrundfarbe := makeRGBValue(255,255,255);
12    Statistiktabelle.löschen;
13
14    Transport_OEM.Bearbeitungszeit := str_to_time("2:00:00");
15    löscheConsole;
16 end;

```

Quellcode A.7: Methode reset

B Sonstiges

Bachelorarbeit

aus dem Fachgebiet IT in Produktion und Logistik

Kandidat: Erkut Baydar
Matrikelnummer: 150553
Fakultät: 7 – Maschinenbau
Betreuender Assistent: Dipl.-Inf. Anne Antonia Scheidler

Thema: Data-Farming-Konzept in Tecnomatix Plant Simulation

Die Simulation hat in den letzten Jahren durch ihre Vorteile innerhalb von Produktion und Logistik an Bedeutung gewonnen, denn sie erlaubt das Erlangen von Erkenntnissen über ein reales System, ohne direkt in dieses eingreifen zu müssen. In der Planung und Verbesserung von Supply Chains (SCs) hat die ereignisdiskrete Simulation bereits erfolgreich Anwendung gefunden. Ein verbreitetes Problem in der Planungsphase von SCs sind fehlende Daten. Hier setzt die Technik des Data Farmings an, die automatisierte Simulationsdurchläufe mit Parametervariation zur Datengenerierung nutzt.

Das Ziel dieser Bachelorarbeit ist die Erstellung eines Simulationsmodells mithilfe des Tools „Tecnomatix Plant Simulation“, sowie die Entwicklung eines zugehörigen Data-Farming-Konzepts. Im Rahmen des Data Farmings sollen automatisierte Simulationsdurchläufe unter Parametervariation konzipiert und umgesetzt werden.

Im Einzelnen sind folgende Aufgaben zu bearbeiten:

- Aufbereitung des Standes der Technik zur ereignisdiskreten Simulation in SCs und Data Farming
- Planung eines Data-Farming-Konzepts mit Parametervariation
- Erstellen eines SC-Simulationsmodells, dessen bewegliche Einheiten benutzerdefinierte Attribute übermitteln
- Verifikation des Simulationsmodells mittels funktionalem Test
- Bereitstellung der Data-Farming-Ergebnisse in einem standardisierten Format

Die Arbeit ist in ständigem Kontakt mit dem benannten Mitarbeiter des Fachgebietes durchzuführen. Für die Bearbeitung sind 3 Monate ab Vergabe der Aufgabe vorgesehen.

Ort, Datum

Prof. Dr.-Ing. Markus Rabe

Empfang der Aufgabenstellung und Einverständnis mit der Überlassung der Ausarbeitung an das Fachgebiet, verbunden mit dem nicht exklusiven Recht der – auch elektronischen – Publikation sowie die Kenntnis der Regeln zur „Guten wissenschaftlichen Praxis für das Verfassen wissenschaftlicher Qualifikationsarbeiten“ bestätigt:

Ort, Datum

Ercut Baydar


Projektpartner		Checkliste / Phase				Projekt / Auftrag		
		C6 – Aufgabendefinition				Verantwortung		
lfd. Nr.	Relevanz	Status	Dokument	Bearbeiter	Priorität	Deadline	Aktivität	ergänzende Unterlagen / Anmerkungen
Organisatorisches								
1			eD				Meilensteine für die Datenbeschaffung festlegen	
2			eD				Glossar anlegen	
Fachinhaltliches								
3			D2.1				Ziele und Studienzweck gemäß Zielbeschreibung endgültig abstimmen und dokumentieren	
4			D2.3				Fehlende Unterlagen zum zu untersuchenden System einholen	
5			D2.2				Systemgrenzen gemäß Zielbeschreibung festlegen	
6			D2.6				Mindestanforderungen für den Detaillierungsgrad von Modellkomponenten festlegen	
7			D2.6				für Projektziel erforderlichen Detaillierungsgrad für Modellkomponenten festlegen	
8			D2.3				Daten- und Informationsbedarf gemäß Zielbeschreibung und Detaillierungsgrad festlegen	
9			D2.4				geplante Modellnutzung festlegen	
10			D2.6				Abnahmekriterien für das Modell endgültig festlegen	
11			eD				Abnahmekriterien für die gesamte Simulationsstudie endgültig festlegen	
Dokumentnummer		Änderungsvermerke				Abnahme		Seite
		Nr.: Datum: Zeichen:				Datum:		1 von 1
Version	Datum	Nr.: Datum: Zeichen:				Unterschrift:		
		Nr.: Datum: Zeichen:						

Abbildung B.2: Aufgabendefinition nach [Wenzel et al., 2008]


Projektpartner						Checkliste / Phase		Projekt / Auftrag			
						C8a – Systemanalyse					
								Verantwortung			
ifd. Nr.	Relevanz	Status	Dokument	Bearbeiter	Priorität	Deadline	Aktivität	ergänzende Unterlagen / Anmerkungen			
							Organisatorisches				
1			—				regelmäßig Abstimmungsgespräche der Projektpartner durchführen				
2			eD				Glossar erweitern				
3			eD				erfolgreiche Validierung des Konzeptmodells bestätigen				
							Fachinhaltliches				
4			D3.2				Struktur und Komponenten des Systems erfassen				
5			D3.3				Beziehungen und Wechselwirkungen zwischen Systemkomponenten bestimmen				
6			D3.2				Hierarchien von Systemkomponenten bzw. Teilsystemen bestimmen				
7			D3.4				Eingangsrößen, Ausgangsrößen und interne Systemzustände festlegen				
8			D3.4				Material- und Informationsflüsse erfassen				
9			D3.3				Detaillierungsgrad der Komponenten spezifizieren				
10			D3.3				nicht zu modellierende Systemkomponenten benennen				
11			eD				deskriptive bzw. semiformale Modellbeschreibung erstellen				
12			D3.5				Anforderungen an das Modell in Bezug auf die spätere Verwendung klären				
13			eD				geplante Variationen der Modellparameter festlegen				
14			D3.4				Daten- und Informationsbedarf präzisieren				
15			D3.4				Auswirkungen von Datenmangel abschätzen				
Dokumentnummer			Änderungsvermerke				Abnahme		Seite		
			Nr.: Datum: Zeichen:				Datum:		1 von 2		
Version		Datum		Nr.: Datum: Zeichen:				Unterschrift:		1 von 2	
				Nr.: Datum: Zeichen:						1 von 2	

Abbildung B.3: Systemanalyse nach [Wenzel et al., 2008]


Projektpartner		Checkliste / Phase				Projekt / Auftrag		
		C8b – Modellformalisierung				Projekt / Auftrag		
						Verantwortung		
Ifd. Nr.	Relevanz	Status	Dokument	Bearbeiter	Priorität	Deadline	Aktivität	ergänzende Unterlagen / Anmerkungen
Organisatorisches								
1			—				Regelmäßig Abstimmungsgespräche der Projektpartner durchführen	
2			eD				erfolgreiche Validierung des formalen Modells bestätigen	
Fachinhaltliches								
3			D4.3				gemäß Konzeptmodell erforderliche Systemkomponenten formal beschreiben	
4			D4.3				Steuerung des Systems formalisieren	
5			D4.3				stochastische Parameter der Modellkomponenten festlegen	
6			D4.3				Charakteristika der stochastische Parameter (z.B. Mittelwert der Verteilung) ermitteln	
7			D4.4				Daten- und Informationsbedarf (insb. für stochastische Parameter) weiter präzisieren	
8			D4.4				Auswirkungen von Datenmangel abschätzen	
9			D4.4				Sensitivitätsanalysen durchführen	
			eD				Validierung des formalen Modells durchführen	
Dokumentnummer		Änderungsvermerke				Abnahme		Seite
		Nr.:	Datum:	Fachinhaltliches		Datum:		1 von 1
Version	Datum	Nr.:	Datum:	Zeichen:		Unterschrift:		
		Nr.:	Datum:	Zeichen:				

Abbildung B.4: Modellformalisierung nach [Wenzel et al., 2008]


Projektpartner						Checkliste / Phase		Projekt / Auftrag	
						C8c – Implementierung			
								Verantwortung	
lfd. Nr.	Relevanz	Status	Dokument	Bearbeiter	Priorität	Deadline	Aktivität	ergänzende Unterlagen / Anmerkungen	
							Organisatorisches		
1			—				aktuelle zu verwendende Lizenz prüfen		
2			—				Nutzung wiederverwendbarer Bibliotheken rechtlich prüfen		
3			eD				erfolgreiche Validierung des ausführbaren Modells bestätigen		
							Fachinhaltliches		
4			—				Grundsätze des Software Engineering beachten		
5			—				Umsetzbarkeit des formalen Modells mit Simulationswerkzeug prüfen		
6			—				werkzeugspezifische Implementierungsrestriktionen prüfen und umsetzen		
7			—				Konsequenzen einer eingeschränkten Implementierbarkeit für Projektziel ermitteln		
8			D5.2				Daten und Modell getrennt implementieren und dokumentieren		
9			D5.2				physikalisches Modell und Steuerung getrennt implementieren		
10			eD				Erweiterungen am Simulationswerkzeug dokumentieren		
11			—				Notwendigkeit der Kopplung an weitere Software prüfen		
12			eD				Dokumentation der Schnittstellen und Protokolle prüfen		
13			eD				Softwarekopplung implementieren		
14			eD				Validierung des ausführbaren Modells durchführen		
Dokumentnummer			Änderungsvermerke				Abnahme		Seite
			Nr.: Datum: Zeichen:				Datum:		
Version		Datum	Nr.: Datum: Zeichen:				Unterschrift:		1 von 1
			Nr.: Datum: Zeichen:						

Abbildung B.5: Implementierung nach [Wenzel et al., 2008]

Literaturverzeichnis

- [Arndt 2015] ARNDT, Holger: *Logistikmanagement*. Wiesbaden : Springer Fachmedien Wiesbaden, 2015 (Studienwissen kompakt). – ISBN 978-3-658-07211-7
- [Arndt 2014] ARNDT, Viktoria: *Ereignisdiskrete Simulation einer Supply Chain zur Generierung von Transaktionsdaten*, Technische Universität Dortmund, Diplomarbeit, 2014
- [Arnold u. Furmans 2009] ARNOLD, Dieter ; FURMANS, Kai: *Materialfluss in Logistiksystemen*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2009 (VDI-Buch). – ISBN 978-3-642-01404-8
- [Bangsow 2008] BANGSOW, Steffen: *Fertigungssimulationen mit Plant-Simulation und SimTalk: Anwendung und Programmierung mit Beispielen und Lösungen*. München : Hanser, 2008 (Edition CAD.DE). – ISBN 978-3-446-41490-7
- [Bangsow 2011] BANGSOW, Steffen: *Praxishandbuch Plant Simulation und SimTalk: Anwendung und Programmierung in über 150 Beispiel-Modellen*. München : Hanser, 2011. – ISBN 978-3-446-42782-2
- [Barry u. Koehler 2004] BARRY, Philip ; KOEHLER, Matthew: Simulation In Context: Using Data Farming For Decision Support. In: *Proceedings of the 2004 Winter Simulation Conference*, 2004. – ISBN 0-7803-8786-4
- [Becker 2008] BECKER, Torsten: *Prozesse in Produktion und Supply Chain optimieren. 2*. Berlin, Heidelberg : Springer-Verlag, 2008. – ISBN 978-3-540-77555-3
- [DIN 19226 1968] DIN 19226: *Regelungstechnik und Steuerungstechnik - Begriffe und Benennungen*. Berlin, 1968
- [EDAG Production Solutions GmbH & Co. KG 2015] EDAG PRODUCTION SOLUTIONS GMBH & Co. KG: *Plant Simulation - Materialflusssimulation von Produktions- und Logistiksystemen*, 2015. <http://www.isilog.de/de/produkte/produkte/plant-simulation.html>. – Zuletzt geprüft am: 20.01.2016
- [Eley 2012] ELEY, Michael: *Simulation in der Logistik: Einführung in die Erstellung ereignisdiskreter Modelle unter Verwendung des Werkzeuges "Plant Simulation"*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012 (Springer-Lehrbuch). – ISBN 978-3-642-27372-8
- [Gehr u. Hellingrath 2007] GEHR, Frank ; HELLINGRATH, Bernd: *Logistik in der Automobilindustrie: Innovatives Supply Chain Management für wettbewerbsfähige Zulieferstrukturen*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2007 (VDI-Buch). – ISBN 3-540-14045-X
- [Hedtstück 2013] HEDTSTÜCK, Ulrich: *Simulation diskreter Prozesse: Methoden und Anwendungen*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013 (eXamen.press). – ISBN 978-3-642-34871-6

- [Heidtmann 2008] HEIDTMANN, Volker: *Organisation von Supply Chain Management: Theoretische Konzeption und empirische Untersuchung in der deutschen Automobilindustrie*. Wiesbaden : Gabler Verlag / GWV Fachverlage GmbH Wiesbaden, 2008. – ISBN 978-3-8349-0885-8
- [Heiserich et al. 2011] HEISERICH, Otto-Ernst ; HELBIG, Klaus ; ULLMANN, Werner: *Logistik: Eine praxisorientierte Einführung*. 4., vollständig überarbeitete und erweiterte Auflage. Wiesbaden : Gabler Verlag / Springer Fachmedien Wiesbaden GmbH Wiesbaden, 2011. – ISBN 978-3-8349-1852-9
- [Horne u. Brandstein 1998] HORNE, Gary E. ; BRANDSTEIN, Alfred G.: *Data Farming: A Meta-Technique For Research In The 21st Century*. Maneuver Warfare Science, 1998
- [Horne u. Meyer 2004] HORNE, Gary E. ; MEYER, Ted E.: *Data Farming: Discovering Surprise*. Proceedings of the 2004 Winter Simulation Conference, 2004
- [Horne u. Schwierz 2008] HORNE, Gary E. ; SCHWIERZ, Klaus-Peter: *Data Farming: Around The World Overview*. Proceedings of the 2008 Winter Simulation Conference, 2008
- [Jetzke 2007] JETZKE, Siegfried: *Grundlagen der modernen Logistik: Methoden und Lösungen*. München : Hanser, 2007. – ISBN 978-3-446-40459-5
- [Klug 2010] KLUG, Florian: *Logistikmanagement in der Automobilindustrie: Grundlagen der Logistik im Automobilbau*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2010 (VDI-Buch). – ISBN 978-3-642-05292-7
- [Košturiak u. Gregor 1995] KOŠTURIK, Ján ; GREGOR, Milan: *Simulation von Produktionssystemen*. Wien : Springer, 1995. – ISBN 9783211827017
- [Kuhn 1998] KUHN, Axel: *Simulation in Produktion und Logistik: Fallbeispielsammlung*. Berlin : Springer, 1998. – ISBN 9783540638544
- [Maryann Lawlor 2005] MARYANN LAWLOR: Data Farming Cultivates New Insights. In: *AFCEA SIGNAL Magazine* (2005). <http://www.afcea.org/content/?q=data-farming-cultivates-new-insights>. – Zuletzt geprüft am: 06.01.2016
- [März et al. 2011] MÄRZ, Lothar ; KRUG, Wilfried ; ROSE, Oliver ; WEIGERT, Gerald: *Simulation und Optimierung in Produktion und Logistik: Praxisorientierter Leitfaden mit Fallbeispielen*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2011 (VDI-Buch). – ISBN 978-3-642-14535-3
- [Müller 2005] MÜLLER, Martin: *nbF Neue Betriebswirtschaftliche Forschung*. Bd. 341: *Informationstransfer im Supply Chain Management: Analyse aus Sicht der Neuen Institutionenökonomie*. Wiesbaden : Deutscher Universitätsverlag, 2005. – ISBN 978-3-8350-0026-1
- [NATO Science and Technology Organization 2014] NATO SCIENCE AND TECHNOLOGY ORGANIZATION: *Data Farming in Support of NATO / North Atlantic Treaty Organization*. 2014 (TR-MSG-088)
- [Pawellek 2008] PAWELLEK, Günther: *Ganzheitliche Fabrikplanung: Grundlagen, Vorgehensweise, EDV-Unterstützung*. Berlin, Heidelberg : Springer-Verlag, 2008 (VDI-Buch). – ISBN 978-3-540-78402-9

- [Rabe 2001] RABE, Markus: *Handlungsanleitung Simulation in Produktion und Logistik: Ein Leitfaden mit Beispielen für kleinere und mittlere Unternehmen*. Erlangen : SCS European Publ. House, 2001. – ISBN 1–56555–226–1
- [Rabe 2003] RABE, Markus: *Modellierung von Layout und Steuerungsregeln für die Materialfluss-Simulation*. Stuttgart : Fraunhofer-IRB-Verl., 2003 (Berichte aus dem Produktionstechnischen Zentrum Berlin). – ISBN 3–8167–6262–X
- [Rabe u. Scheidler 2015] RABE, Markus ; SCHEIDLER, Anne A.: Farming for Mining - Entscheidungsunterstützung mittels Simulation im Supply Chain Management. In: *Simulation in Production and Logistics 2015*. Stuttgart : Fraunhofer IRB Verlag, 2015. – ISBN 0–7803–8786–4
- [Rabe et al. 2008] RABE, Markus ; SPIEKERMANN, Sven ; WENZEL, Sigrid: *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken*. Berlin, Heidelberg : Springer-Verlag, 2008 (VDI-Buch). – ISBN 978–3–540–35281–5
- [Sanchez 2014] SANCHEZ, Susan M.: Simulation Experiments: Better Data, Not Just Big Data. In: *Proceedings of the 2014 Winter Simulation Conference*, 2014. – ISBN 978–1–4799–7486–3
- [Sanchez u. Wan 2012] SANCHEZ, Susan M. ; WAN, Hong: Work Smarter, Not Harder: A Tutorial On Designing And Conducting Simulation Experiments. In: *Proceedings of the 2012 Winter Simulation Conference*, 2012. – ISBN 978–1–4673–4781–5
- [Schneider 2008] SCHNEIDER, Markus: *Logistikplanung in der Automobilindustrie: Konzeption eines Instruments zur Unterstützung der taktischen Logistikplanung vor eStart-of-Production^a im Rahmen der Digitalen Fabrik*. Wiesbaden : Gabler Verlag / GWV Fachverlage GmbH Wiesbaden, 2008. – ISBN 978–3–8349–1213–8
- [Siemens Industry Software GmbH 2015] SIEMENS INDUSTRY SOFTWARE GMBH: *Plant Simulation*, 2015. http://www.plm.automation.siemens.com/de_de/products/tecnomatix/plant_design/plant_simulation.shtml. – Zuletzt geprüft am: 20.01.2016
- [Siemens PLM Software 2015] SIEMENS PLM SOFTWARE: *Tecnomatix Plant Simulation Hilfe*, 2015. <http://www.plm.automation.siemens.com>
- [VDI-Richtlinie 3633 2013] VDI-RICHTLINIE 3633: *Simulation von Logistik-, Materialfluss- und Produktionssystemen - Begriffe*. Berlin, 2013
- [VDI-Richtlinie 3633, Blatt 3 2013] VDI-RICHTLINIE 3633, BLATT 3: *Simulation von Logistik-, Materialfluss- und Produktionssystemen - Experimentplanung und -auswertung*. Berlin, 2013
- [Waters 2009] WATERS, Donald: *Supply chain management: An introduction to logistics*. 2. ed. Basingstoke : Palgrave Macmillan, 2009. – ISBN 978–0–2302–0052–4
- [Wenzel et al. 2008] WENZEL, Sigrid ; COLLISI-BÖHMER, Simone ; PITSCH, Holger ; ROSE, Oliver ; WEISS, Matthias: *Qualitätskriterien für die Simulation in Produktion und Logistik: Planung und Durchführung von Simulationsstudien*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2008 (VDI-Buch). – ISBN 978–3–540–35272–3

-
- [Werner 2014] WERNER, Hartmut: *Kompakt Edition: Supply Chain Controlling: Grundlagen, Performance-Messung und Handlungsempfehlungen*. Wiesbaden : Springer Fachmedien Wiesbaden, 2014. – ISBN 978-3-658-05621-6
- [Werners 2013] WERNERS, Brigitte: *Grundlagen des Operations Research: Mit Aufgaben und Lösungen*. 3., überarb. Aufl. 2013. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013 (Springer-Lehrbuch). – ISBN 978-3-642-40101-5

Eidesstattliche Versicherung

Name, Vorname

Matr.-Nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit mit dem Titel „Data Farming Konzept in Tecnomatix Plant Simulation“ 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.

Ort, Datum

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:

Ort, Datum

Unterschrift