

# Fachwissenschaftliche Projektarbeit (BA)

## Modellierung und Simulation selbststeuernder miniaturisierter Logistiksysteme mit GD.findi und Arena

Dominik Patzelt  
Studiengang: Maschinenbau (B.Sc.)  
Matrikelnummer: 177451

Christian Fahrenholz  
Studiengang: Maschinenbau (B.Sc.)  
Matrikelnummer: 158611

Ausgegeben am: 28.11.2017  
Eingereicht am: 30.05.2018

Prüfer: Univ.-Prof. Dr.-Ing. Markus Rabe  
Betreuer: M.Sc. Joachim Hunker, Dipl.-Geinf. Maik Deininger

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>1</b>
<b>2</b>	<b>Selbststeuernde Logistiksysteme</b> .....	<b>3</b>
2.1	Logistiksysteme im Allgemeinen .....	3
2.2	Selbststeuerung .....	4
2.3	Fördermittel .....	5
<b>3</b>	<b>Simulation</b> .....	<b>6</b>
3.1	Simulation allgemein .....	6
3.2	Simulationsmodell .....	7
3.3	Simulationsstudie .....	9
3.4	Simulation in der Praxis .....	12
<b>4</b>	<b>GD.findi</b> .....	<b>15</b>
4.1	Allgemeine Informationen .....	15
4.2	Programmaufbau .....	16
4.3	Ablauf einer Modellierung .....	17
<b>5</b>	<b>Arena Simulation Software</b> .....	<b>21</b>
5.1	Allgemeine Informationen .....	21
5.2	Programmaufbau .....	22
5.3	Ablauf einer Modellierung .....	27
<b>6</b>	<b>Simulationsbeispiel 1: Laufradfertigung</b> .....	<b>29</b>
6.1	Erklärung .....	29
6.2	Abbildung in GD.findi .....	32
6.3	Abbildung in Arena .....	35
6.4	Vergleich der Simulationstools im Hinblick auf Simulationsbeispiel 1 .....	37
<b>7</b>	<b>Simulationsbeispiel 2: Fertigung eines Heckspoilers</b> .....	<b>39</b>
7.1	Erklärung .....	39
7.2	Abbildung in GD.findi .....	41
7.3	Abbildung in Arena .....	44
7.4	Vergleich der Simulationstools im Hinblick auf Simulationsbeispiel 2 .....	47
<b>8</b>	<b>Gegenüberstellung von GD.findi und Arena</b> .....	<b>48</b>
8.1	Festlegung der Kriterien zur Bewertung .....	48
8.2	Bewertung .....	49
<b>9</b>	<b>Zusammenfassung</b> .....	<b>53</b>
<b>10</b>	<b>Abbildungsverzeichnis/ Tabellenverzeichnis</b> .....	<b>I</b>
<b>11</b>	<b>Literaturverzeichnis</b> .....	<b>III</b>
<b>12</b>	<b>Anhang</b> .....	<b>V</b>

## 1 Einleitung

Die Industrie steht heutzutage vor der großen Herausforderung, immer komplexer werdende Logistik- und Produktionssysteme beherrschen zu müssen ([GRS<sup>+</sup>17], S.1). Um als Unternehmen wettbewerbsfähig zu bleiben, ist es deshalb extrem wichtig, sich den neuen Umständen schnellstmöglich anzupassen ([MKR<sup>+</sup>11], S.3). Allerdings ist dies bei einer sehr großen Anzahl von Einflussfaktoren auf das System ein schwieriges Unterfangen. Sobald jedoch das dynamische Verhalten eines Systems bekannt ist, kann daraus die Erkenntnis gewonnen werden, wie die Produktion auszulegen ist ([MKR<sup>+</sup>11], S.4). Um zu diesem Ergebnis zu kommen wird das Verfahren der Simulation benutzt. Es unterstützt die Planung und Prozessoptimierung der realen Fabrik und führt zu „abgesicherten und besser nachvollziehbaren Planungsergebnissen“ ([GRS<sup>+</sup>17], S.1). Eine Simulation wird durch ein Simulationswerkzeug umgesetzt, von denen der Markt eine sehr große Anzahl für unterschiedliche Anwendungsgebiete bereitstellt. Eine Möglichkeit zur Reduktion der Komplexität bietet die Entwicklung von selbststeuernden Logistiksystemen ([Bou17], S.7).

Im Rahmen dieser Projektarbeit ist herauszufinden, ob die beiden Simulationswerkzeuge GD.findi und Arena Simulation Software für die Anwendung auf selbststeuernde, miniaturisierte Logistiksysteme, wie etwa dem MINIS-Projekt der TU Dortmund, geeignet sind. Dazu sollen die Simulatoren auf zwei verschiedene Simulationsbeispiele angewendet werden. Aus den dort gewonnenen Erkenntnissen lassen sich die Programme vergleichen und es ergeben sich die Vorteile und Nachteile der Werkzeuge. In dem MINIS-Projekt wird ein selbststeuerndes, modulares Logistiksystem von einigen Lehrstühlen, Instituten und Fachschaften der TU-Dortmund entwickelt. Dabei ist das ITPL in dem Projekt für die Simulation der logistischen Vorgänge zuständig, mit welcher man die Systeme abbilden und testen kann.

Diese Projektarbeit wurde durch zwei Autoren verfasst und deshalb fällt es schwer, bei einigen Kapiteln den richtigen Autor zu nennen. Bei der Durchführung der Arbeit wurde allerdings darauf geachtet, dass der Aufwand gleichmäßig auf die beiden Verfasser aufgeteilt ist. Im Folgenden wird auf das Vorgehen eingegangen, das bei der Bearbeitung dieser Arbeit angewendet wurde. Dabei wird der Autor dem Teil der Arbeit zugeordnet, den er bearbeitet hat.

Zu Beginn wird dem Leser die Theorie, die zum Verständnis der Projektarbeit nötig ist, nahegebracht. Zuerst werden die selbststeuernden produktionslogistischen Systeme definiert (*Kapitel 2 bearbeitet von Patzelt*). Danach wird eine Einführung in die Simulation gegeben. Darin werden neben den allgemeinen Begrifflichkeiten, beispielsweise die Vorgehensweise bei einer Erstellung einer Simulationsstudie oder die Klassifizierung von Simulationswerkzeugen thematisiert (*Kapitel 3 bearbeitet von Fahrenholz*). Nachfolgend wird auf die beiden zu untersuchenden Simulationstools eingegangen. Es werden zunächst die allgemeinen Informationen zum jeweiligen Programm beschrieben, bevor der Aufbau des Programms und der Ablauf einer Modellierung erklärt wird. Als erstes wird sich mit GD.findi auseinandergesetzt (*Kapitel 4 bearbeitet von Fahrenholz*) und anschließend mit Arena Simulation Software (*Kapitel 5 bearbeitet von Patzelt*). In dem nächsten Teil der Arbeit wird die Eignung der beiden Werkzeuge mithilfe von zwei verschiedenen Anwendungsbeispielen überprüft. Das erste Beispiel wird aus einer alten Arbeit übernommen, um die Simulationstools aus dieser Projektarbeit mit denen aus der alten Arbeit zu vergleichen. Das zweite haben die Autoren selbst gewählt. Nach einer Erläuterung des Beispiels (*Kapitel 6.1 bearbeitet von Fahrenholz; Kapitel 7.1 bearbeitet von Patzelt*), wird seine Abbildung im jeweiligen Simulationswerkzeug beschrieben. Zuerst wurde die Umsetzung in GD.findi beschrieben (*Kapitel 6.2 & 7.2 bearbeitet von Fahrenholz*), gefolgt von der Beschreibung in Arena Simulation Software (*Kapitel 6.3 & 7.3 bearbeitet von Patzelt*). Um die beiden Tools zu vergleichen werden die Erkenntnisse aus der Modellierung gegenübergestellt und bewertet (*Kapitel 6.4 & 7.4 bearbeitet von Patzelt und Fahrenholz*). Nachdem die Modellierung der beiden Programme an jeweils zwei Simulationsbeispielen getestet wurde, wurden genügend Erkenntnisse gesammelt, um die Werkzeuge ausführlich zu vergleichen. Dazu werden Bewertungskriterien festgelegt und nach ihnen werden die beiden Programme bewertet (*Kapitel 8 behandelt von Patzelt und Fahrenholz*). Schlussendlich werden die Erfahrungen aus der Projektarbeit in einem Fazit gebündelt und es wird eine kurze abschließende Bewertung vorgenommen (*Kapitel 9 behandelt von Patzelt und Fahrenholz*).

## 2 Selbststeuernde Logistiksysteme

Um eine Simulation an einem, dem MINIS-Projekts ähnlichen, logistischen System durchführen zu können, muss der Begriff des „selbststeuernden, miniaturisierten Logistiksystems“ im Hinblick auf das Projekt definiert werden. Der Begriff „miniaturisiert“ ist hierbei selbsterklärend und bedeutet, dass es sich um eine verkleinerte Darstellung realer Fördermittel handelt. Die für dieses Projekt zur Verfügung stehenden Fördermittel werden zum Ende dieses Kapitels aufgezeigt. Da es sich bei dem MINIS-Projekt um ein selbststeuerndes Logistiksystem handelt, wird hier zuerst der Begriff Logistik im Allgemeinen erläutert. Anschließend werden Arten von Logistiksystemen benannt und unterschieden. Des Weiteren wird darauf eingegangen, was es bedeutet, dass ein Logistiksystem selbststeuernd ist. Außerdem wird geklärt, welchem logistischen System das MINIS-Projekt zuzuordnen ist.

### 2.1 Logistiksysteme im Allgemeinen

Die Logistik beschreibt alle praktischen Prozesse, welche mit dem Transport, Verkehr, Umschlag und Lagern von Gütern in direktem Zusammenhang stehen und greift auf eine lange Geschichte zurück. Dabei soll es das Ziel der Logistik sein ein „Effizientes Bereitstellen der geforderten Mengen benötigter Objekte in der richtigen Zusammensetzung zur rechten Zeit am richtigen Ort“ ([Gud10], S. 3) zu erreichen. Heutzutage versteht man unter dem Begriff vor allem die Bildung von einer Verbindung von Netzwerken, welche „Unternehmen, Haushalte und Konsumenten in aller Welt permanent mit den benötigten Gütern und Waren versorgen.“ ([Gud10], S. xix)

Solche Logistiksysteme können in Makro- und Mikrosysteme unterteilt werden. Die Makrologistik beschreibt dabei die Sicherung einer globalen, staatlichen oder regionalen Versorgung mit Gütern. Hierbei ist eine gut ausgebaute Infrastruktur aus Verkehrsnetzen und Logistikzentren unerlässlich. Die Mikrologistik beschäftigt sich dagegen mit dem Angebot und der Ausführung von Logistikleistungen, welche auf privaten Verträgen beruhen. Eine funktionierende Unternehmenslogistik ist hier eine wichtige Voraussetzung. Diese kann wiederum in innerbetriebliche Logistik oder auch Intralogistik genannt und außerbetriebliche Logistik oder auch Extralogistik genannt unterteilt werden. Unter Intralogistik versteht man dabei die logistischen Prozesse an einem Standort. Bei der Extralogistik werden dagegen die

Wareneingänge und Warenausgänge zwischen verschiedenen Standorten betrachtet. ([Gud10], S. 6 f.)

Für die in den folgenden Kapiteln verwendeten Simulationsbeispiele im Rahmen des MINIS-Projekts sollen intralogistische Prozesse aus der Mikrologistik betrachtet werden, da das Projekt eindeutig als ein solches zu bestimmen ist. Der angedachte Versuchsaufbau mit vielen Modulen, welche in verschiedensten Konstellationen kombiniert werden können ist dabei auf einen Standort ausgelegt und für Prozesse zwischen Wareneingang und Warenausgang konzipiert, dies trifft auf einen intralogistischen Prozess der Mikrologistik zu.

## 2.2 Selbststeuerung

Mit der voranschreitenden Technik im Bereich der „Industrie 4.0“ entstehen auch in der Logistik neue Möglichkeiten. Die Vision der Industrie 4.0 ist laut dem Bundesministerium für Bildung und Forschung (BMBF) unter anderem die digitale Fabrik in welcher „Maschinen, die miteinander kommunizieren, sich gegenseitig über Fehler im Fertigungsprozess informieren [...] [und] knappe Materialbestände identifizieren und nachbestellen“. ([Bmbf])

In diesem Zuge ist auch der Begriff der Selbststeuerung zu klären. Denn „Selbststeuerung [...] eines Systems bedeutet, dass dieses eigenständig und ohne direkte Anweisung von außen ein vorgegebenes Ziel verfolgt oder einen Zustand aufrechterhält“. ([Bou17], S.7) Bezogen auf logistische Prozesse kann man hier also von einer dezentralen Steuerung der Güter ohne menschliche Eingriffe sprechen. Denn durch Kommunikations- und Informationstechnologien wie „RFID“ (Radio Frequency Identification) können die Güter in gewissen Grenzen autonom transportiert und verteilt werden. ([Wyc09], S.112)

### 2.3 Fördermittel

Zur Realisierung eines selbststeuernden Logistiksystems stehen zurzeit im Rahmen des MINIS-Projekts verschiedene Fördermittel zu Verfügung. Diese können als Module auf unterschiedliche Weise kombiniert werden. Die Simulationsbeispiele im Rahmen dieser Projektarbeit werden einige dieser Module in ähnlicher Form verwenden und damit in direktem Bezug zum MINIS-Projekt stehen. Die Tabelle auf der folgenden Seite zeigt eine Auflistung der verschiedenen Module mit ihrer Definition.

*Tabelle 1: Module MINIS-Projekt*

Modulbezeichnung	Definition
Linienmodul	Gerades Transportband, welches mittels angetriebener Rollen Güter fördern kann.
90°-Kurve	90°-Kurve mit vergleichbarer Transportweise wie das Linienmodul
180°-Kurve	180°-Kurve mit vergleichbarer Transportweise wie das Linienmodul
Drehtisch mit RFID-Chip	RFID-Tags können ausgelesen werden. Der obere Teil des Tisches ist autonom drehbar, sodass Güter den Tisch in verschiedene Richtungen verlassen können. Der Tisch besitzt angetriebene Rollen zum Fördern der Güter.
Verteilerwagen	Wagen mit angetriebenen Rädern kann Güter transportieren. Kann nur auf gerader Strecke fahren.
Auswerfer	Transportmodul mit mechanischem Auswerfer.

Um ein solches selbststeuerndes Logistiksystem effizient und mit möglichst hoher Wirtschaftlichkeit betreiben zu können, müssen bei der Planung eines solchen Systems bestimmte Produktionsfaktoren berücksichtigt werden. Denn neben hoher Produktivität mit kurzen Durchlaufzeiten, sollen auch niedrige Lagerbestände und dem Kunden gegenüber eine hohe Termintreue realisiert werden. Des Weiteren sollen die Kosten niedrig gehalten werden. Da diese Größen allerdings gegenseitigen Wechselwirkungen unterliegen, werden Simulationstools genutzt um das System abzubilden und anschließend optimieren zu können. ([MKR<sup>+</sup>11], S.8ff)

## 3 Simulation

Dieses Kapitel enthält eine Einführung in die Simulation. Zuerst werden in Kapitel 3.1 allgemeine Begrifflichkeiten geklärt und der Bedarf an der Simulation wird erläutert. Danach wird auf das Simulationsmodell, deren Klassifizierung und Elemente in Kapitel 3.2 eingegangen. In Kapitel 3.3 wird die Simulationsstudie und das Vorgehen bei der Erstellung von dieser beschrieben. Abschließend beschäftigt sich Kapitel 3.4 mit der Simulation in der Praxis. Dazu zählen die Einsatzfelder, sowie die Betrachtung der Wirtschaftlichkeit einer Simulationsstudie. Außerdem wird das Simulationswerkzeug beschrieben.

### 3.1 Simulation allgemein

Die Simulation wird von dem VDI als ein „Verfahren zur Nachbildung eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind“ konkretisiert ([VDI18], S.28). Als System versteht man eine Anzahl von Elementen, die miteinander in Beziehung stehen und dessen vereinfachte Nachbildung wird als Modell bezeichnet ([Ele12], S.3).

Die heutige Wirtschaft stellt immer höhere Anforderungen, welche zu komplexeren technischen Systemen und kürzeren Entwicklungs- und Planungszyklen führen. Um diesen Anforderungen gerecht zu werden, wird die Simulation nicht nur in der Planung neuer Systeme angewendet, sondern findet auch in der Realisierung und im Betrieb von diesen ihren Nutzen. Somit deckt sie den kompletten Lebenszyklus eines Systems ab und besitzt damit zahlreiche Anwendungsfelder ([VDI18], S.4). Beispielsweise ermöglicht sie die Untersuchung an Systemen, die noch gar nicht bestehen und noch in Planung sind. Aber auch bereits bestehende Systeme können ohne einen direkten Betriebseingriff untersucht werden. Außerdem lässt sich das Systemverhalten über lange Zeiträume ermitteln, oder es können Anlaufvorgänge, Einschwingphasen und Übergänge zwischen definierten Betriebszuständen ausgewertet werden ([VDI18], S.5).

Die Ergebnisse einer Simulation liefern allerdings keine Lösung, sondern bewerten lediglich das Systemmodell ([MKR<sup>+</sup>11], S.18). Diese Bewertung muss zuerst interpretiert werden. Dafür werden die Ergebnisdaten des Simulationsdurchlaufes mit den Zielvorgaben an das System verglichen. Durch die Variation von den Parametern im Modell und den Einflussgrößen wird



versucht, die Differenz von Ergebnisdaten und Zielvorgaben zu minimieren ([MKR<sup>+</sup>11], S.10). Für diese „Suche nach dem Optimum unter gegebenen Voraus- und Zielsetzungen“ ([MKR<sup>+</sup>11], S.11) ist die Optimierung zuständig.

Für die Nutzung von Simulationen existieren einige wichtige Leitsätze, die zu beachten sind:

- „Simulation stets vor Investition.
- Simulation setzt vorherige Zieldefinition und Aufwandsabschätzung voraus.
- Vor der Simulation analytische Methoden ausschöpfen.
- Simulation ist kein Ersatz für Planung.
- Die Abbildungsgenauigkeit ist nicht so detailliert wie möglich, sondern so detailliert wie zur Zielerfüllung erforderlich.
- Die Ergebnisqualität von Simulationsexperimenten kann nicht besser sein als die dem Simulationsexperiment zugrunde liegenden Informationen, das heißt, Simulationsergebnisse sind wertlos oder irreführend, wenn die Datenbasis fehlerhaft ist oder aber die Ergebnisse falsch interpretiert werden.
- Simulationsergebnisse können nur so gut sein wie die Zusammenarbeit innerhalb des an der Simulationsstudie beteiligten Personenkreises“ ([VDI18], S.6)

### 3.2 Simulationsmodell

Das Simulationsmodell ist die „Abbildung eines realen oder geplanten Systems zur Zweck der Simulation“ ([VDI18], S.29). Es gibt verschiedene Arten von Simulationsmodellen. Die Einordnung in die jeweiligen Typen erfolgt mithilfe von drei Merkmalen: Das Verhältnis von Zeit, die Berücksichtigung von Wahrscheinlichkeiten und die Art der Änderung von Zustandsgrößen. Bei der Klassifizierung nach der Zeit existiert das statische Modell, bei dem Zeit keinen Einfluss hat oder das ein System zu einem bestimmten Zeitpunkt beschreibt. Das Modell, welches sich mit fortschreitender Zeit verändert, nennt man dynamisches Simulationsmodell ([Law07], S.5). Wird die Existenz vom Zufall betrachtet, steht das deterministische Modell dem stochastischen Modell gegenüber. Das deterministische Modell hat keine Bestandteile, die Wahrscheinlichkeiten berücksichtigen. Das heißt, dass das Ergebnis feststeht, sobald die Eingabewerte mitgeteilt wurden. Ein Beispiel für dieses Modell sind mathematische Formeln. Bei dem stochastischen Modell gibt es Elemente, die den Zufall

berücksichtigen. Deshalb ist der Output bei wiederholten Experiment unterschiedlich, obwohl der Input gleichbleibend ist ([Law07], S.6). Kategorisiert man nach dem letzten Merkmal, der Änderung von Zustandsgrößen im Modell, lassen sich Simulationsmodelle in stochastische und diskrete Modelle unterscheiden. Das stochastische Modell wird durch die kontinuierliche Änderung der Zustandsgrößen mit fortlaufender Zeit charakterisiert. In der Regel beinhaltet dieses Modell Differentialgleichungen, die die Änderungsrate der Zustandsgrößen mit der Zeit in Verbindung setzt ([Law07], S.70). Auf der anderen Seite steht das diskrete Modell, bei dem sich die Zustandsgrößen unverzüglich zu diskreten Zeitpunkten ändern ([Law07], S.3). Die Entscheidung, ob ein diskretes oder kontinuierliches Modell für ein System benutzt wird, hängt von den spezifischen Zielen einer Analyse ab ([Law07], S.5)

In der Produktion und Logistik werden die meisten Systeme mit Modellen simuliert, „die das dynamische Verhalten des Systems unter Verwendung stochastischer Komponenten mit Zustandsänderungen an diskreten Zeitpunkten abbilden“ ([MKR<sup>+</sup>11], S.14). Solche Modelle werden ereignisorientiert genannt, da beim Eintritt von einem Ereignis eine Zustandsänderung beschrieben wird und sie gehören zum Typ der ereignisorientierten Simulation (Discrete Event Simulation, DES) ([MKR<sup>+</sup>11], S.14).

Um ein Simulationsmodell zu erstellen, bedarf es einiger Grundelemente. Die hinreichend genaue Beschreibung des Systems zu einem bestimmten Zeitpunkt übernehmen die Zustandsgrößen (system state variables). Für die Abbildung der im System befindlichen Vorgänge sind Entitäten (entities) zuständig. Sie werden unterteilt in dynamische und statische Entitäten unterteilt. Die dynamischen Entitäten sind Objekte, die sich durch das System bewegen. Ein Beispiel für diese Objekte ist ein Auto, welches durch eine Lackiererei geführt wird. Die Farbe des Autos würde mit einem Attribut beschrieben werden. Attribute (attributes) beziehen sich auf Entitäten und lassen sich mit lokalen Variablen vergleichen. Dabei kann ein Objekt mehrere Attribute haben. Innerhalb des Systems durchläuft die dynamische Entität eine oder mehrere statische Entitäten – auch Ressourcen (resources) genannt. Eine Ressource kann sich nicht bewegen und wird von einem Objekt besetzt. Bezogen auf das Beispiel, wäre sie eine Lackiererei. Wenn die Ressource belegt ist, werden die nachfolgenden Objekte entweder in die Warteschlange (queue) der Ressource gesetzt, oder bekommen eine andere Aufgabe, wie zum Beispiel die Umleitung zu einer freien Ressource. Der Arbeitsinhalt an einer Ressource wird durch die Tätigkeit (activity) vereinfacht abgebildet. Sie beinhaltet lediglich die Dauer des Arbeitsschritts, beispielsweise die Dauer des

Lackiervorgangs. Der Anfang und das Ende von Tätigkeiten sind Ereignisse, welche für die Zustandsänderungen in einem Modell verantwortlich sind ([Ban98], S.7).

### 3.3 Simulationsstudie

Der VDI definiert die Simulationsstudie als ein „Projekt zur simulationsgestützten Untersuchung eines Systems“ ([VDI18], S.30). Dabei wird angemerkt, dass eine Studie mehrere Experimente beinhalten kann ([VDI18], S.30), welche das Modellverhalten „über einen bestimmten Zeithorizont durch wiederholte Simulationsläufe mit systematischen Parametervariationen“ ([VDI18], S.29) gezielt empirisch analysieren. Jeder dieser Simulationsläufe bildet das untersuchte System mithilfe eines Simulationsmodells über einen vorher bestimmten Zeitraum, der Simulationszeit, ab ([VDI18], S.29). Bei der Analyse von komplizierten Systemen müssen die Simulationsexperimente im Vorfeld statistisch geplant werden, um den „Anforderungen der Statistik an die Menge, Qualität und Aussagekraft der Daten“ auch dann gerecht zu werden, „wenn nicht alle denkbaren Parameterkonstellationen geprüft werden können“ ([VDI97.3], S.5). Die statistische Experimentplanung beschäftigt sich daher mit der Bestimmung der erforderlichen Simulationsläufe und Simulationsdauer, der geeigneten Wahl der Messzeitpunkt und Messintervalle, sowie der Einstellung der in den Experimenten zu untersuchenden Einflussgrößen ([VDI97.3], S.5).

Damit die Simulationsstudie einen größtmöglichen Erfolg hat, wird sich bei der Vorbereitung und Durchführung an einem Vorgehensmodell orientiert. Es beschreibt die einzelnen Schritte der Studie und setzt sie in Beziehung zueinander. Das vorgestellte Vorgehensmodell bezieht sich auf das von Rabe et al. ([RSW08], S.4) ausgearbeitete Modell (siehe Abbildung 1, S. 11). Die Besonderheit dieses Modells ist es, dass das Modell (Ellipsen) und seine Daten (Rechtecke) getrennt betrachtet werden. Jeder Phase des Modells wird als Ergebnis entweder ein Modell, ein Dokument oder eine Kombination aus beidem zugeordnet. Die Zielbeschreibung gilt als Anfangspunkt für die Simulationsstudie ([GRS<sup>+</sup>17], S.142) und beschreibt das durch die Simulation zu lösende Problem. Darauf baut die erste Phase der Simulationsstudie – die Aufgabendefinition – auf. Sie präzisiert die Zielbeschreibung und hat die Aufgabenspezifikation als Ergebnis. Dieses Dokument beschreibt die Aufgabe ausführlich und „erzeugt ein gemeinsames Grundverständnis hinsichtlich der gestellten Aufgabe und des Lösungsweges“ ([GRS<sup>+</sup>17], S.144) aller Projektbeteiligten. Die Systemanalyse trifft Annahmen

für das Simulationsmodell. Sie entscheidet, mit welcher Genauigkeit die Elemente des zu untersuchenden Systems modelliert werden und in welcher Beziehung sie stehen. Die Ergebnisse werden im Konzeptmodell beispielsweise durch ein Ablaufdiagramm, oder einer Systembeschreibung festgehalten. In der Phase der Modellformalisierung wird aus dem Konzeptmodell das formale Modell, welches das System detaillierter beschreibt. Das formale Modell lässt sich bestenfalls ohne zusätzliche Erklärung zum realen System implementieren, ist allerdings noch nicht an ein Simulationswerkzeug gebunden. Die letzte Phase der Modellbildung ist die Implementierung mit dem ausführbaren Modell als Ergebnis. In diesem Schritt wird ein Simulationswerkzeug ausgewählt und das formale Modell darin implementiert ([GRS<sup>+</sup>17], S.144). Die Schritte der Datenbeschaffung und der Datenaufbereitung mit ihren Ergebnissen – den Rohdaten und aufbereiteten Daten – werden gesondert zu den Modellierungsschritten betrachtet, „da sie inhaltlich, zeitlich sowie bezüglich der einzubindenden Personen unabhängig von der Modellierung erfolgen können“ ([GRS<sup>+</sup>17], S.142). Zuletzt werden die aufbereiteten Daten und das ausführbare Modell in der Phase „Experimente und Analyse“ zusammengeführt und es wird ein Experimentplan erstellt. Durch die Analyse der Ergebnisse aus den Experimenten werden Schlussfolgerungen für das reale System abgeleitet. Diese Schlussfolgerungen ergeben zusammen mit den für die Studie verwendeten Daten und deren Analyse die Simulationsergebnisse.

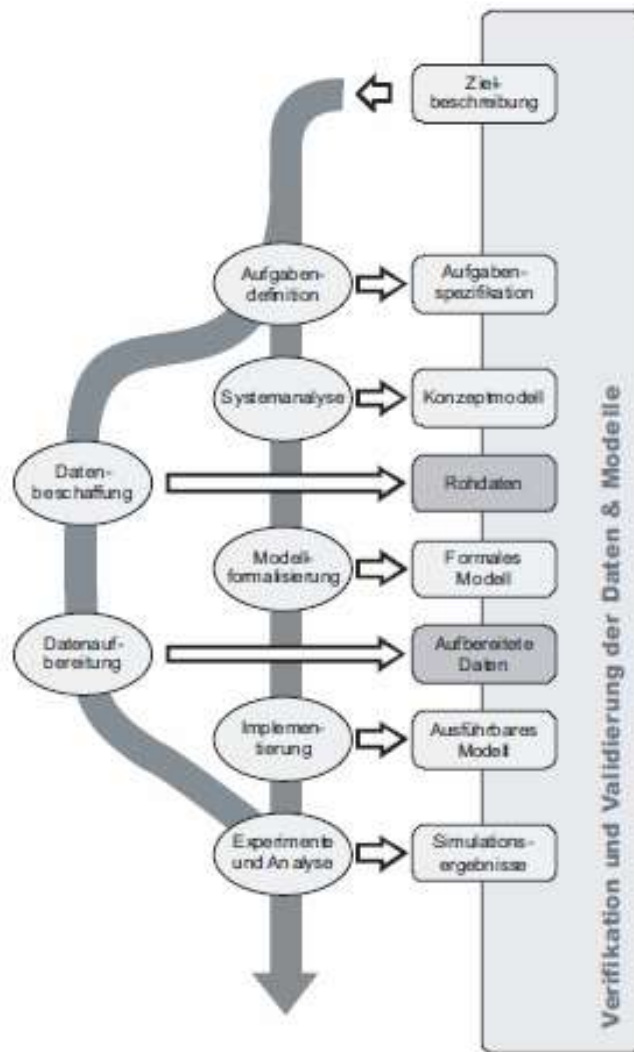


Abbildung 1: Vorgehensmodell bei der Simulation mit V&V

Ein stetiger Begleiter der Phasen einer Simulationsstudie ist die Verifikation & Validierung (V&V). Sie prüft die Phasenergebnisse auf ihre Richtigkeit und Brauchbarkeit ([VDI14], S.37). Ihr wesentliches Ziel ist es zu verhindern, dass aus einer Simulationsstudie die falschen Schlüsse gezogen werden, die zu Fehlentscheidungen führen ([RSW08], S.2). Des Weiteren soll die V&V die Glaubwürdigkeit eines Modells mithilfe der „Fragen „Ist das Modell richtig?“ (Verifikation) ([RSW08], S.14) und „Ist es das richtige Modell“ (Validierung)“ ([RSW08], S.15) prüfen. Dieses Unterfangen erweist sich als sehr schwierig, da die misslungene Modellierung eines Systems bereits durch ein Beispiel bewiesen werden kann. Demgegenüber kann die Korrektheit eines Modells selbst durch eine große Anzahl von Beispielen nicht belegt werden, sondern sie wird nur wahrscheinlich ([RSW08], S.1).

Abschließend werden die fünf grundlegenden Qualitätskriterien für die erfolgreiche Umsetzung von Simulationsprojekte erläutert:

- „Sorgfältige Projektvorbereitung
- Konsequente Dokumentation
- Durchgängige Verifikation und Validierung
- Kontinuierliche Integration des Auftraggebers
- Systematische Projektdurchführung“ ([WWC08], S.4)

### 3.4 Simulation in der Praxis

In der Produktion und Logistik werden Simulationsstudien in den unterschiedlichsten Einsatzfeldern durchgeführt. Hier werden die Einsatzfelder „entlang des Lebenszyklus von Produktionssystemen in Planungs-, Realisierungs-, und Betriebsphase“ ([GRS<sup>+</sup>17], S.45) unterschieden. Der größte Teil der Simulationsstudien ist der Planungsphase zuzuordnen. Das zweithäufigste Einsatzfeld ist die Realisierungsphase und der Rest fällt in den Bereich der Betriebsphase ([GRS<sup>+</sup>17], S.45).

Die Wirtschaftlichkeit der Durchführung einer Simulationsstudie ist zu Beginn unbekannt und lässt sich erst rückblickend bestimmen, da sich das Verhältnis von Kosten und Nutzen im Vorfeld nicht berechnen lässt. Die Kosten unterteilen sich grob in die Kosten für das Personal, das mit der Studie beauftragt ist und in Kosten für den Erwerb und die Wartung der eingesetzten Software. Allerdings ist der Nutzen anfangs nicht quantifizierbar. Mithilfe eines wirklichkeitsnahen Verlaufs der Kosten über die Zeit lässt sich der Vorteil einer Simulationsstudie darstellen (Abbildung 2, S.13). Nach diesem Diagramm muss man beim Einsatz einer Simulation in den Phasen der Planung und Realisierung zwar mehr Geld ausgeben, allerdings sind dann die Investitionen für Nachbesserungen in der Phase der Inbetriebnahme nicht so hoch. Außerdem kann das System dank des Einsatzes von Simulation schneller in Betrieb genommen werden ([GRS<sup>+</sup>17], S.47).

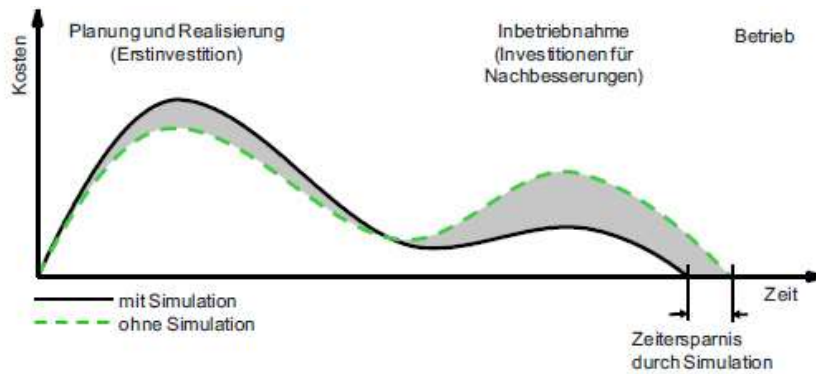


Abbildung 2: Schematischer Verlauf der Kosten von der Planung bis zur Inbetriebnahme ([GRS\*17], S.48)

Aus den Erfahrungen bereits durchgeführter Simulationsstudien lassen sich vier qualitative Nutzenaspekte identifizieren:

- Sicherheitsgewinn
- Kostengünstigere Lösungen
- Besseres Systemverständnis
- Günstigere Prozessführung

Auch wenn sich diese Punkte nicht quantifizieren lassen, sprechen sie für den Einsatz von Simulation ([VDI14], S.40).

Eine zwingende Voraussetzung für den Erfolg einer Simulation ist die Auswahl des richtigen Simulationswerkzeuges, welche sich anhand ihres Anwendungsbezugs klassifizieren lassen. In der Abbildung 3 ist eine Pyramide zu sehen, die in verschiedene Ebenen aufgeteilt ist. Diese Ebenen beschreiben die Klassen der Simulationswerkzeuge – unterteilt nach ihren Spezialisierungsgraden. Die obere Hälfte der Pyramide ist mit Werkzeugen belegt, die lediglich für einen begrenzten Anwendungsbereich eingesetzt werden können. Der Vorteil dieser Werkzeuge ist, dass sie vorgefertigte Bausteine und Methoden bereitstellen, die die Modellimplementierung erleichtern. Im unteren Teil sind Werkzeuge aufgelistet, die allgemeingültig sind. Links neben der Pyramide verdeutlichen zwei Pfeile die Beziehung von dem Anwendungsbezug zur Allgemeingültigkeit. Je höher der Anwendungsbezug eines Werkzeuges ist, desto geringer ist seine Allgemeingültigkeit, beziehungsweise Flexibilität. Auf der rechten Seite sind für jede Klasse eine Anzahl von Werkzeugen gegeben, die auf dem Markt erhältlich sind ([Ele12], S.10).

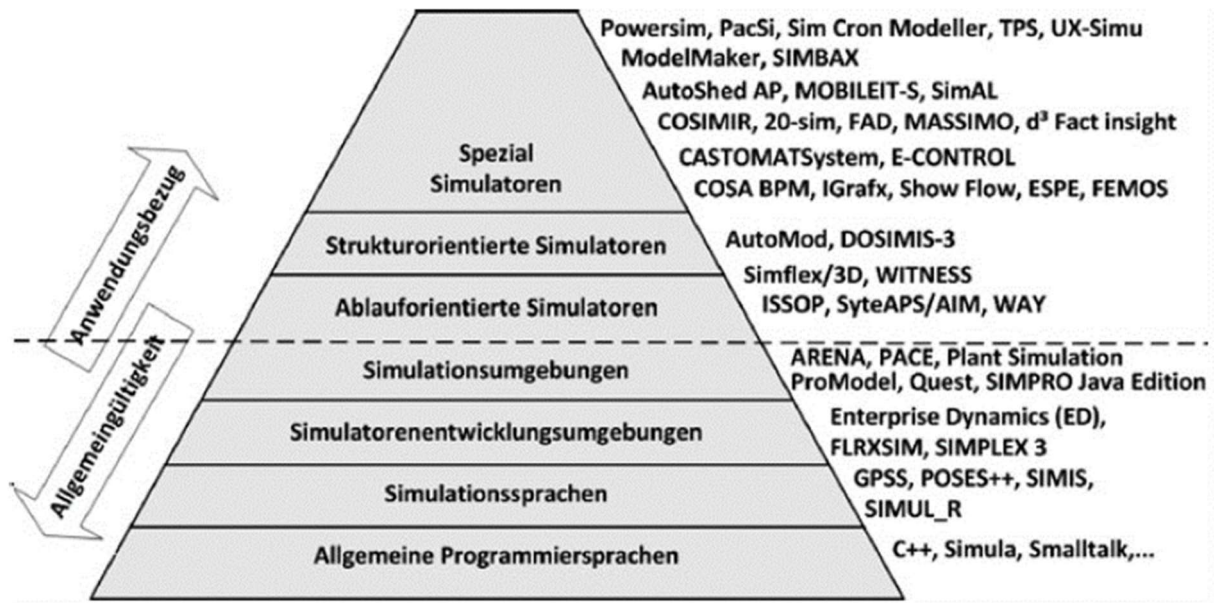


Abbildung 3: Klassifizierung von Simulationswerkzeugen ([Ele12], S.10)

Um das richtige Simulationswerkzeug für eine Anwendung zu finden, bedarf es einer Eingrenzung, da es nicht sinnvoll ist, jedes verfügbare Werkzeug auf die Eignung zu untersuchen. Diese Eingrenzung lässt sich in drei Stufen unterteilen. Die erste Stufe ist die Grobauswahl, bei der das Marktangebot nach einem geeigneten Werkzeug betrachtet wird. Es werden maximal vier bis sechs Werkzeuge ausgewählt, die für die weiteren beiden Stufen infrage kommen. Als nächstes findet die „engere Auswahl auf Basis von Testbeispielen, Demonstrationen und ggf. bereits einer Nutzwertanalyse, um so eine weitere Eingrenzung auf maximal zwei bis drei Simulationswerkzeuge vorzunehmen“ ([GRS<sup>+</sup>17], S.234). Im letzten Schritt wird die Endauswahl mithilfe von Testinstallationen und ggf. einer Nutzwertanalyse bestimmt ([GRS<sup>+</sup>17], S.234).



## 4 GD.findi

In dem folgenden Kapitel wird das Simulationswerkzeug GD.findi beschrieben und es wird die grundlegende Funktionsweise erläutert. Zuerst wird auf allgemeine Informationen des Programms und den Hersteller eingegangen. Danach wird die Benutzeroberfläche veranschaulicht, indem sie in verschiedene Bereiche eingeteilt wird. Dabei wird zusätzlich auf die Elemente eingegangen, die für das MINIS-Projekt relevant sind. Abschließend wird der Ablauf einer Simulation von der Modellerstellung bis zur Ergebnisauswertung herausgearbeitet.

### 4.1 Allgemeine Informationen

Das Programm GD.findi ist ein Simulator für Produktionssysteme. Es wurde von der japanischen Firma LEXER RESEARCH Inc. entwickelt und im Jahr 2012 auf den Markt gebracht ([Lex1]). Das Unternehmen befasst sich mit der Entwicklung von Programmen, die Entwicklungsprozesse unterstützen und hat mit GD.findi einen Simulator geschaffen, der sich ohne vorherige Programmierkenntnisse bedienen lässt ([Lex2]). Das Programm ist über eine Anfrage an den Hersteller auf dessen Webseite in zwei unterschiedlichen Versionen zu erwerben. Die kostenlose Version ist in ihrem Funktionsumfang im Gegensatz zur kostenpflichtigen Pro-Version eingeschränkt ([Lex3]). So kann man beispielsweise nur drei verschiedene Projekte erstellen oder sich die Ergebnisse einer Simulation nicht als Datei ausgeben lassen. In der Projektarbeit wurde mit der Version 3.1.6583.39038 gearbeitet, welche auch die aktuellste ist. Die Standardsprache des Programms ist Englisch, doch sie kann auch auf Japanisch umgestellt werden. Um dem Anwender den Einstieg in das Programm zu erleichtern, bietet der Hersteller unter anderem zwei halbtägige Kurse an. GD.findi wird über den Browser benutzt, wobei der Internet Explorer zur Nutzung empfohlen wird. Außerdem muss die Browser-Erweiterung Microsoft Silverlight installiert sein. Laut Bedienungsanleitung ist auch die Anwendung über macOS möglich, doch in dieser Arbeit wurde ausschließlich der Internet Explorer genutzt ([Lex4], S.4). Seitens des Herstellers gelten keine besonderen Systemanforderungen.

## 4.2 Programmaufbau

Das Programm hat einen höheren Anwendungsbezug als Arena und lässt sich somit im oberen Bereich der Pyramide in Abbildung 3 einordnen (s. Kapitel 3.4, Seite 14). Allerdings stellt GD.findi keine Bibliothek mit vorkonfigurierten Bausteinen bereit, was die genaue Eingrenzung des Simulationswerkzeuges erschwert. Eine Besonderheit von GD.findi ist, dass bei der Abbildung eines Systems in Raumplan (*floor plan*) und Produktionsprozess (*production process*) unterschieden wird. Der Raumplan beinhaltet das Layout der jeweiligen Anlage oder Halle, inklusive der darin befindlichen Arbeitsstationen. Im Produktionsprozess wird die Fertigungsmethode in einzelne Stationen unterteilt. Diese Eigenschaft macht es beispielsweise möglich, dass bei einer nachträglichen Änderung des Produktionsprozesses nur dieser neu erstellt werden muss und mit dem bestehenden Raumplan verknüpft werden kann.

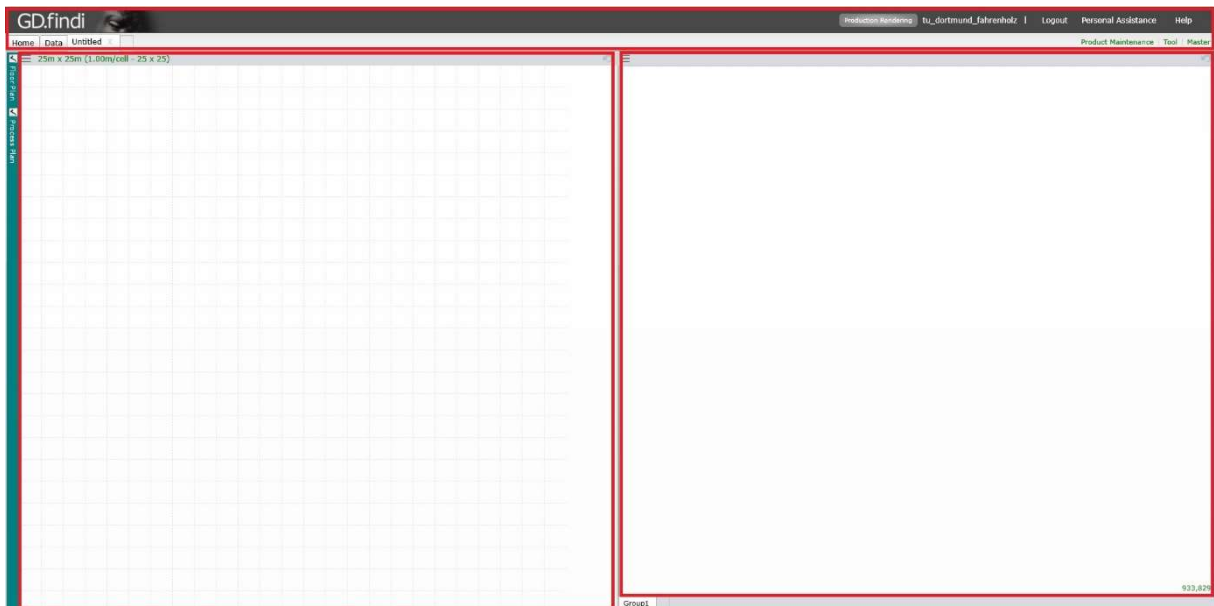


Abbildung 4: Grafische Benutzeroberfläche GD.findi

In Abbildung 4 ist die grafische Benutzeroberfläche zu erkennen. Es fällt auf, dass sie sehr simpel gestaltet ist. Aus diesem Grund ist ein neuer Benutzer nach Angaben des Herstellers nach wenigen Stunden Ausbildung dazu imstande, eigene Simulationen zu erstellen ([Lex5], S.3). Die Oberfläche kann in drei Bereiche aufgeteilt werden. Auf der linken Seite ist der Raumplan. Dort werden die Elemente in der Halle durch Stationen abgebildet und mithilfe von Pfeilen und Knoten miteinander verbunden. Die Stationen gehören in GD.findi zu den wichtigsten Komponenten. Sie werden verwendet, um einen Arbeitsbereich oder Apparaturen, die zu einer Anlage gehören, abzubilden ([Lex4], S.9) und ersetzen somit die

typischen Elemente aus vergleichbaren Simulationswerkzeugen, wie zum Beispiel Quellen, Senken, Prozesse oder Verteiler. Des Weiteren werden die Bausteine zur Beförderung von Entitäten durch eine Verbindung von Knoten und Pfeilen, die an die Stationen angeschlossen werden, abgebildet. Auf der rechten Seite befindet sich der Produktionsprozess. Dieser kann durch Hinzufügen und Beschreibung von verschiedenen Arbeitsschritten dargestellt werden. Die Leiste im oberen Bereich der Webseite beinhaltet grundlegende Funktionen zum Ausführen der Simulation. Beispielsweise kann sie darüber gestartet werden, oder es lässt sich die favorisierte Ausgabe der Simulationsergebnisse auswählen. Außerdem befindet sich dort die Verlinkung zu einer Hilfeseite, falls bei der Modellierung Probleme auftreten sollten.

### 4.3 Ablauf einer Modellierung

Der Hersteller hat zur Anwenderunterstützung ein Vorgehensmodell für neue Simulationen erstellt, dessen Schritte im Folgenden erläutert werden. Es gliedert sich in sieben Schritte ([Lex6]). Der erste Schritt beinhaltet die Einrichtung eines neuen Projektes. Bei der Einrichtung wird die Größe des Hallen-, bzw. Anlagenlayouts bestimmt. Als nächstes wird dieses Layout mit den verschiedenen Arbeitsbereichen und Apparaturen der Anlage gefüllt. Diese werden durch Stationen abgebildet und können sich per Drag and Drop auf dem Layout erstellen lassen. Eine Verbindung zwischen Stationen wird mithilfe von Anschlüssen, Knoten und Punkten abgebildet. Jede Station, die einen Input oder Output hat, muss mindestens einen Anschluss besitzen. Dabei gibt es keine Begrenzung, wie viele Ströme von Entitäten zugeführt oder abgeführt werden dürfen. Dadurch lassen sich nicht nur Arbeitsbereiche herstellen, sondern beispielsweise auch Elemente, die mehrere Entitäten aus unterschiedlichen Eingängen auf einen Ausgang zusammenführen oder umgekehrt. Die Wege zwischen den Anschlüssen werden mit Knoten angefertigt, welche durch Pfeile verbunden werden (siehe Abbildung 5). Diese Pfeile sollten den richtigen Maßstab haben, da sich aus deren Länge die Länge des Förderelements ergibt.

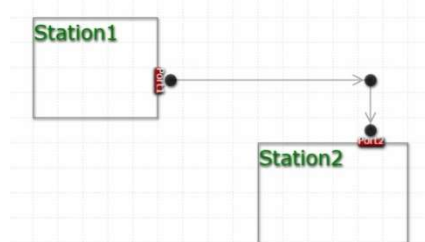


Abbildung 5: Verbindung von Stationen durch Anschlüsse, Knoten und Pfeile

Aufgrund der vereinfachten Darstellung von Fördererelementen lassen sich die im MINIS-Projekt verwendeten Elemente nicht genau abbilden. Das gerade Linienmodul lässt sich zwar durch den Pfeil beschreiben, doch wird dabei nicht auf die Anzahl der verwendeten Fördererelemente eingegangen. Da in GD.findi keine gebogenen Pfeile existieren, lassen sich die 90°-/180°-Kurvelemente nur durch ein, bzw. zwei aufeinanderfolgende rechte Winkel realisieren. Der Drehtisch ist durch eine Arbeitsstation darstellbar, die mehrere Ausgänge hat und der eine bestimmte Zeit für den Prozess des Drehens zugeschrieben wird. Der Auswerfer ähnelt dem Drehtisch mit dem Unterschied, dass bei ihm die Dauer des Drehens nicht berücksichtigt werden muss, da ein Objekt auf diesem Förderer nicht zum Stillstand gebracht wird. Nach dem Aufbau des Raumplans wird der Prozessplan ausgearbeitet. Es gilt dabei zu beachten, dass für jeden sich in der Anlage befindlichen Produkttypen ein eigener Prozessplan erstellt wird. Die einzelnen Arbeitsschritte und Apparaturen lassen sich dabei sehr simpel aneinanderreihen. Zu dieser Phase der Modellierung gehört ebenfalls die Zuweisung der Bearbeitungsdauer und der Vermerk von allen Eingängen und Ausgängen des Prozesses (siehe Abbildung 6).

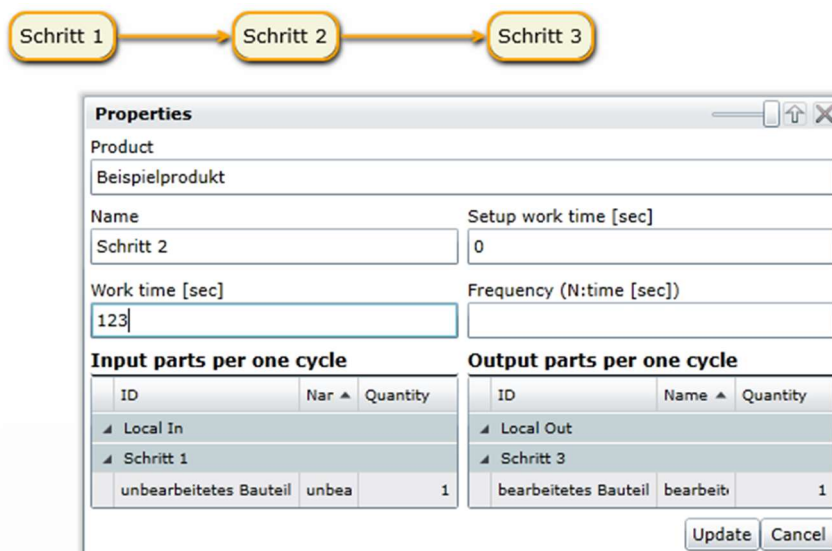


Abbildung 6: Beispiel Prozessplan

Im nächsten Schritt werden die Prozesselemente aus dem Produktionsprozess den jeweiligen Stationen im Raumplan zugeordnet. Dies passiert ebenfalls durch Drag and Drop. Darüber hinaus werden in dieser Phase die Transportrouten definiert. Dafür werden im Raumplan die Menüs Activity und Agent Designer geöffnet. Im Agent Designer lässt sich ein Fördererelement beschreiben. Danach werden im Menü Activity die Routen der Entitäten definiert, indem die Anschlüsse und Knoten verknüpft werden. Diesen Routen werden die Eigenschaften des zuvor

beschriebenen Förderelements zugewiesen. Zudem kann in diesem Menü die Wahrscheinlichkeitsverteilung der Bearbeitungszeit einer Station von konstant zu normalverteilt verändert werden. Im Anschluss daran wird die Produktionsplanung bestimmt. Es ist dazu das Produktionsziel anzugeben und die Fertigungsreihenfolge, falls mehrere Produkttypen die Anlage durchlaufen. Im Falle einer fehlerhaften Modellierung werden in diesem Fenster die verschiedenen Fehlermeldungen angezeigt. Diese sind auf der Hilfeseite von GD.findi erklärt und mit den dazugehörigen Lösungswegen beschrieben ([Lex7]). Danach kann die Simulation gestartet werden. Es gibt verschiedene Möglichkeiten für die Darstellung der Simulationsergebnisse. Das Standardinstrument ist das Production Cockpit und es bildet verschiedene Faktoren, wie zum Beispiel das Fertigungsvolumen oder die Auslastung der Stationen, über die Zeit ab. Dabei ist es möglich, mithilfe von Schaltflächen durch den Fertigungsablauf zu springen (siehe Abbildung 7).



Abbildung 7: Production Cockpit

Daneben gibt es das Material Flowchart, welches näher auf die einzelnen Entitäten eingeht sowie die Auslastung von den einzelnen Stationen beschreibt. Als weiteres Tool steht EasyGO zur Alternative, das ebenfalls verschiedene Diagramme zu dem Produktionsprozess anfertigt. Die Ergebnisse lassen sich auch quantifizieren und in einer Excel-Datei ausgeben, doch das ist bei der kostenlosen Version nicht möglich. Der letzte Schritt beinhaltet das Speichern des Projekts.

Zur Anwenderunterstützung bietet der Hersteller auf seiner Homepage dem Kunden eine Bedienungsanleitung und ein Funktionshandbuch zu dem Programm an. Allerdings sind diese Dokumente in der vierten, bzw. dritten Ausgabe von August 2014. Deshalb ist es nützlich, dass bei dem ersten Starten von GD.findi bereits einige Projekte vorhanden sind, die einen schnelleren Einstieg ins Programm bieten, wenn mit ihnen gespielt wird. Falls ein Problem auftaucht, für das keine Lösung gefunden wird, ist es möglich den Hersteller direkt zu kontaktieren. Es existieren ebenfalls einige Videos auf der Videoplattform YouTube zu dem Programm, von denen der größte Teil – wie auch die vorinstallierten Projekte – auf Japanisch sind.

## 5 Arena Simulation Software

In diesem Kapitel soll das Simulationsprogramm „Arena Simulation Software“ vorgestellt und die wichtigsten Funktionen erklärt werden. Dazu wird zuerst ein Überblick über das Programm und den Herausgeber gegeben und anschließend der Programmaufbau dargestellt und die, für die Beispiele in den folgenden Kapiteln verwendeten, Bausteine erläutert. Anschließend wird beispielhaft erklärt, wie man mit der Arena Simulation Software ein neues System modelliert und eine Simulation von diesem durchführt. Zuletzt werden die Möglichkeiten zum Selbststudium von Arena aufgezeigt.

### 5.1 Allgemeine Informationen

Herausgeber von ARENA Simulation Software (im Folgenden: Arena) ist der weltweit operierende US-amerikanische Konzern Rockwell Automation, Inc mit Sitz in Milwaukee, Wisconsin ([RA18]). Das Simulationstool ist im Internet als kostenlose Hochschulversion (*Academic*) oder als kostenpflichtige Business-Version erhältlich. In dieser Projektarbeit wurde die Version 15.00.00004 als Hochschulversion mit Studentenlizenz benutzt, welche im August 2017 herausgegeben wurde. Diese ist zeitlich unbegrenzt, aber gegenüber der Business-Version hinsichtlich der Modellgröße beschränkt und nicht für die kommerzielle Nutzung bestimmt. Neben der Studentenlizenz können auch Lizenzen für Professoren oder wissenschaftliche Mitarbeiter erworben werden. Die Programmiersprache ist ausschließlich Englisch ([ASS18]).

Arena arbeitet unter der Benutzeroberfläche mit ereignisdiskreter, objektorientierter Simulation auf Grundlage der Programmiersprache SIMAN. SIMAN besteht dabei nur aus den Objektklassen *block* und *element*. Die Objekte vom Typ *block* sind dabei logische Konstrukte, welche eine bestimmte Operation ausführen. Objekte vom Typ *element* stellen dagegen Strukturen, zum Beispiel Warteschlangen, dar. ([AM07], S.65)

## 5.2 Programmaufbau

Der Benutzer steht mit der zuvor genannten Programmiersprache über eine grafische Benutzeroberfläche in Verbindung. Diese ermöglicht es vordefinierte Bausteine per Maus in die Arbeitsoberfläche zu ziehen, zu verbinden und so eine, einem Flussdiagramm ähnliche, Struktur zu erstellen. Die Oberfläche (Abbildung 8) besteht dazu aus mehreren Fenstern, welche zum Teil personalisiert werden können. Die Fenster und ihre Funktionen sollen anhand der Abbildung erläutert werden.

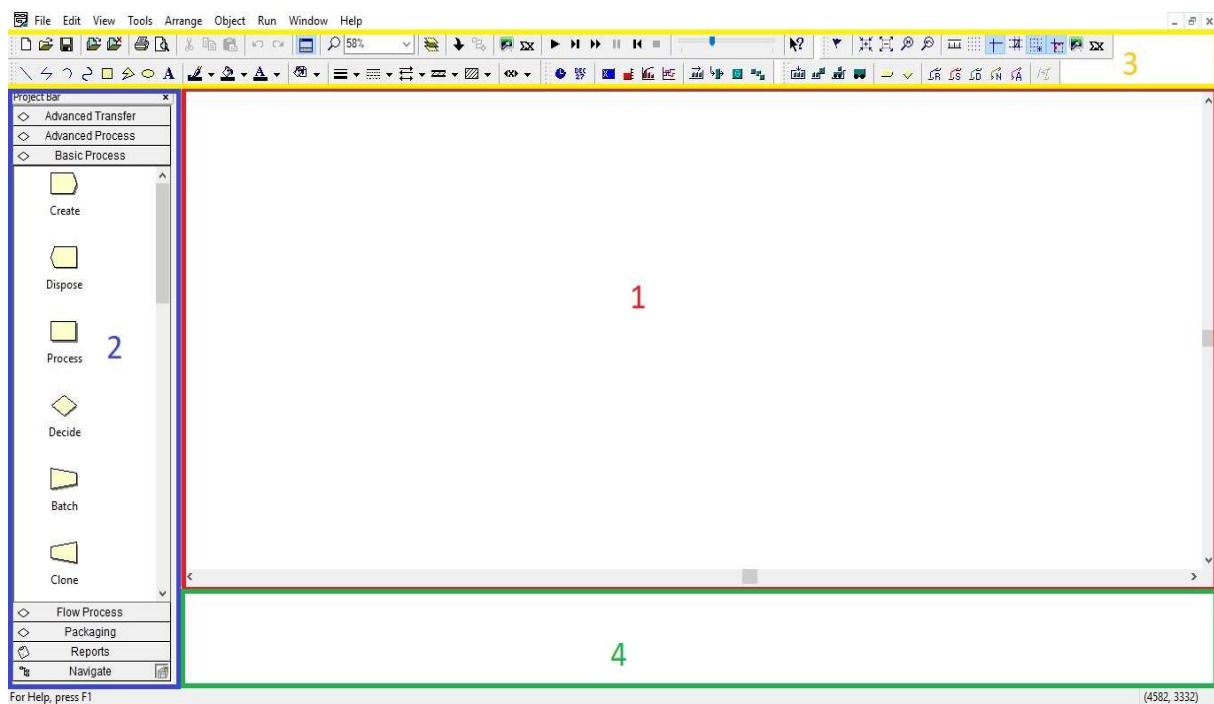


Abbildung 8: GUI Arena Simulation Software

Die Benutzeroberfläche kann, sofern man die obere Menüleiste unberücksichtigt lässt, im Wesentlichen in vier Bereiche unterteilt werden. Der mit „1“ gekennzeichnete Bereich stellt die Arbeitsoberfläche dar, auf welcher das Modell erstellt wird. Dazu werden die benötigten Bausteine aus Bereich „2“, der Projektleiste (*Project Bar*), mit der Maus in den Arbeitsbereich gezogen.

Die Projektleiste, welche die Bausteine zu Verfügung steht ist in mehrere Abschnitte unterteilt. Per Klick auf den gewünschten Abschnitt, zum Beispiel „*Basic Process*“, wird dieser ausgeklappt und die Bausteine, hier zum Beispiel *Create*, *Dispose*, *Process* und Weitere, angezeigt. Wie in 5.1 erklärt, wird hier zwischen Blöcken (*block*) und Elementen (*element*) unterschieden. Die Sortierung sieht dabei vor, dass alle Blöcke in dem Arbeitsbereich über den



Elementen angeordnet sind. Während Blöcke (siehe Abbildung 9) in den Arbeitsbereich gezogen werden können, sind die Elemente (siehe Abbildung 10) dort fest. Wählt man ein Element oder einen Block aus, so kann dieses, beziehungsweise dieser, in Bereich „4“ bearbeitet werden. Wird zum Beispiel eine Entität (*entity*), welche zu den Elementen zählt, ausgewählt, so werden alle Entitäten, welche im Modell bisher verwendet wurden, angezeigt. Neben den grundlegenden Bausteinen für Prozesse („*Basic Process*“) können auch fortgeschrittene Bausteine für Prozesse („*Advanced Process*“), sowie für den Transfer („*Advanced Transfer*“) ausgeklappt werden. Darunter sind personalisierte Bausteine für Fließprozesse („*Flow Process*“) und den Gütertransfer („*Packaging*“), ebenfalls ausklappbar, angeordnet. Zuletzt kommen in der Projektleiste die Auswahlmöglichkeiten für Berichte („*Reports*“) und für die Navigation („*Navigate*“). Unter dem Punkt für Berichte stehen nach einer durchlaufenen Simulation Berichte zu verschiedenen Objekten, wie zum Beispiel den Entitäten, oder auch ein kompletter Überblick über die Ergebnisse der letzten Simulation zur Verfügung. Diese werden bei Auswahl rechts neben der Projektleiste in Druckversion dargestellt. Wird der Bereich Navigation ausgeklappt, so wird ein verkleinerter Arbeitsbereich in der Projektleiste angezeigt und man kann dort, sofern das Modell größer als die Arbeitsoberfläche ist, über das Modell navigieren und so den in der Arbeitsoberfläche angezeigten Bereich anpassen.

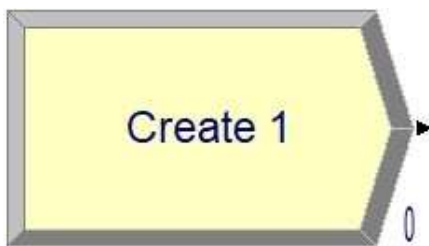


Abbildung 9: Beispiel block



Abbildung 10: Beispiel element

Bereich „3“ stellt in Arena die Werkzeugleiste dar. Durch einen Rechtsklick in diesen Bereich können Unterbereiche ausgewählt werden, diese werden dann in der Werkzeugleiste angezeigt und können in ihrer Position und Reihenfolge verändert werden. So lässt sich diese Leiste vollständig personalisieren. Die wichtigsten Unterbereiche sind *Standard* und *View*. Soll das Modell noch grafisch unterstützt werden, so sind die noch die Unterbereiche *Draw*, *Animate* und *Animate Transfer* von wichtiger Bedeutung. Der Unterbereich *Standard* (siehe Anhang A) gibt unter anderem die Möglichkeit zum Speichern und Öffnen eines Projektes,

stellt aber auch die Auswahlmöglichkeit zum Verbinden zweier Blöcke in der Arbeitsoberfläche (*Connect*) zur Verfügung. Zudem kann hier eine Simulation gestartet, gestoppt, sowie die Geschwindigkeit einer Simulation reguliert werden. Mit dem Bereich View (siehe Anhang B) lässt sich die Arbeitsoberfläche anpassen, so kann ein Koordinatennetz (*Grid*) oder Abmessungen (*Ruler*) angezeigt werden. Mit den Unterbereichen Draw und Animate Transfer ist es möglich das Modell 2-dimensional zu zeichnen und mit dem aus den Bausteinen bestehenden Modell zu verknüpfen. Der Bereich Animate hingegen beinhaltet Extras wie zum Beispiel eine Uhr oder ein Diagramm, welche in das Modell eingefügt werden können.

Der mit „4“ gekennzeichnete Bereich bietet direkten Zugriff auf die Eigenschaften der benutzten Blöcke und Elemente. Wählt man entweder in der Projektleiste oder in der Arbeitsoberfläche ein Element, beziehungsweise einen Block aus, so wird in diesem Bereich eine Tabelle mit allen Objekten des ausgewählten Typs angezeigt. Abbildung 11, welche nur zu Demonstrationszwecken dient und nicht die Realität widerspiegelt zeigt für einen fiktiven Wareneingang die Tabelle für alle Blöcke vom Typ „Create“.







Create - Basic Process								
	Name	Entity Type	Type	Value	Units	Entities per Arrival	Max Arrivals	First Creation
1	Zulieferer 1	Kleinteilpaletten	Constant	7	Seconds	1	Infinite	0.0
2	Zulieferer 2	Großpalette	Constant	5	Seconds	1	Infinite	0.0

Abbildung 11: Tabelle für Baustein vom Typ Create

Jede Zeile steht hier für einen verwendeten Baustein. In diesem Fall wurden also zwei Bausteine vom genannten Typ verwendet. Jede Spalte stellt wiederum eine Eigenschaft des jeweiligen Bausteins dar. Welche Spalten angezeigt werden hängt vom Typ des Bausteines ab. Im Folgenden werden die, für die in den Kapiteln 6 und 7 zur Realisierung der Beispiele genutzten, Bausteine aufgezeigt und erläutert. Dazu werden diese in nachfolgender Tabelle dargestellt. In der ersten Spalte wird dafür das jeweilige Symbol des Bausteins inklusive Namen dargestellt. In Spalte zwei folgt eine sinngemäße Übersetzung aus dem Englischen ins Deutsche. Die dritte Spalte gibt an unter welchem Unterpunkt in der Projektleiste der Baustein zu finden ist und in der vierten Spalte sind Definition und nennenswerte Eigenschaften von dem jeweiligen Baustein zu finden.

Tabelle 2: Verwendete Bausteine Arena

Symbol	Dt. Bezeichnung	Ort	Eigenschaft
 Create	Quelle	Basic Process	Erschafft eine neue Entität und führt diese in das Modell ein. Zeitliche Verteilung und Anzahl der Entitäten pro Zeitabstand, sowie die maximal erschaffenen Entitäten während einer Simulation können festgelegt werden. Außerdem sind verschiedene statistische Verteilungen möglich.
 Dispose	Senke	Basic Process	Ankommende Entitäten verlassen das Modell.
 Process	Prozess	Basic Process	Simuliert einen Arbeitsprozess. Es kann zwischen einer zeitlichen Verzögerung ( <i>Delay</i> ), der zeitlichen Verzögerung mit Aufnahme und Abgabe der Entität ( <i>Seize Delay Release</i> ) und Mischformen ( <i>Seize Delay</i> und <i>Delay Release</i> ) gewählt werden. Bei einer Variante mit Aufnahme und Abgabe einer Entität wird automatisch eine Warteschlange ( <i>Queue</i> ) erzeugt.
 Decide	Verteiler	Basic Process	Verteilt die Entität je nach ausgewählter Methode auf einen von 2 bis N Ausgängen Dabei kann ausgewählt werden, ob die Entscheidungsfindung für einen Ausgang prozentual ( <i>by Chance</i> ) oder nach einem Attribut ( <i>by Condition</i> ) der Entität stattfindet.
 Batch	Führe zusammen	Basic Process	Führt entweder alle eingehenden Entitäten oder alle eingehenden Entitäten, welche eine bestimmte Eigenschaft erfüllen zu einer neuen Entität zusammen. Wie viele Entitäten zusammengeführt werden sollen kann festgelegt werden ( <i>Batch Size</i> ).

 Assign	Zuordnen	Basic Process	Ermöglicht es, einer Entität neue Attribute zuzuordnen oder Attribute, wie zum Beispiel den Typ oder das angezeigte Bild, während der Simulation zu ändern.
 Match	Führe mehrere Eingänge zusammen	Advanced Process	Besitzt 1 bis N Eingänge und einen Ausgang. Führt jeweils eine Entität aus jedem Ausgang zu einer neuen Entität zusammen. Sind nicht an allen Eingängen Entitäten vorhanden, so warten die vorhandenen Entitäten in je einer Warteschlange pro Eingang.
 Station	Förderstation	Advanced Transfers	Steht am Anfang und am Ende einer Förderstrecke.
 Access	Zugang	Advanced Transfers	Steht am Anfang einer Förderstrecke. Ordnet der Entität einer Förderstrecke zu, beziehungsweise belegt einen Platz auf der dieser.
 Convey	Befördere	Advanced Transfers	Verknüpft den Baustein Access mit einer Station und stellt somit die Förderstrecke dar. Erzeugt einen Förderer ( <i>Conveyor</i> ), welcher wiederum ein Segment besitzt. Dadurch kann eine Förderstrecke über Einstellung der Parameter von Segment und Förderer simuliert werden.
 Exit	Ausgang	Advanced Transfers	Steht am Ende einer Förderstrecke. Verknüpft die Station am Ende einer Förderstrecke mit dem nächsten Baustein, zum Beispiel einem Prozess. Der vom Access-Baustein belegte Platz auf der Förderstrecke kann wieder frei gegeben werden.

### 5.3 Ablauf einer Modellierung

Zum Erstellen eines Modells wird in der Werkzeugliste der Punkt „Neu“ (*New*) ausgewählt. Dieser öffnet einen neuen Arbeitsbereich. Die Modellierung eines Systems mit Arena wird im Folgenden an Abbildung 12 erläutert.

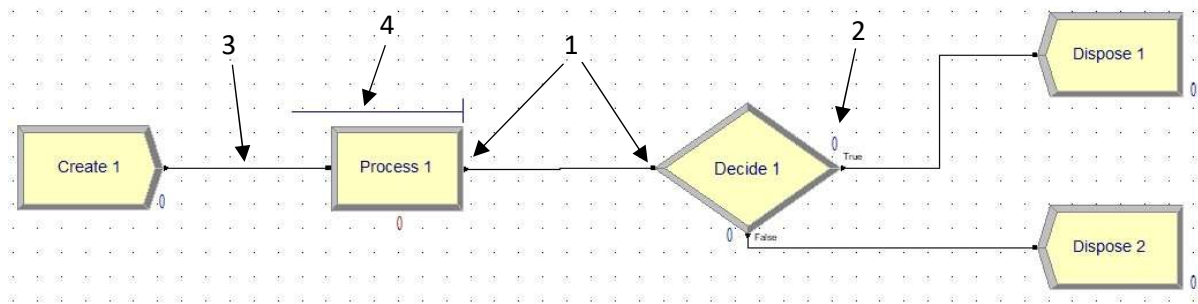


Abbildung 12: Beispiel eines Modells in Arena

Nach dem Öffnen einer neuen Arbeitsoberfläche können die Bausteine, wie bereits erwähnt, aus der Projektleiste in die Arbeitsoberfläche gezogen werden. Nun erscheinen bei einigen Bausteinen schwarze Quadrate und Dreiecke (1). Zudem wird neben jedem Baustein mindestens eine Zahl angezeigt (2). Die Quadrate am Rand eines Bausteins stellen Eingänge zu diesem und die Dreiecke Ausgänge aus diesem dar. Je nach Typ des Bausteins besitzt ein Baustein zwischen keinem und theoretisch unendlich Ein- beziehungsweise Ausgängen. Ein Beispiel für einen Baustein mit keinem Eingang wäre *Create*, welcher nur einen Ausgang besitzt, da dieser eine Entität erschafft. Ein Baustein mit vielen Eingängen wäre dagegen *Batch* mit zwei bis theoretisch unendlich vielen Eingängen. Wird in der Werkzeugleiste der Punkt „Verbinde“ (*Connect*) ausgewählt, so kann durch Anklicken der schwarzen Quadrate, beziehungsweise Dreiecke am Rand der Bausteine eine Verbindung von einem Ausgang zu einem Eingang erstellt werden. Diese Verbindung wird durch eine schwarze Linie (3) dargestellt. Dabei ist eine Verbindung zweier Ausgänge, beziehungsweise zweier Eingänge nicht möglich. Diese Linie stellt zudem keine Fördereinheit, sondern allein die logische Verknüpfung zweier Bausteine dar. Die zuvor erwähnte Zahl (2) ist eine lokale Zählvariable, welche während der Simulation die aktuelle Anzahl der durchlaufenen Entitäten, beziehungsweise bei Prozess-Bausteinen die Anzahl der aktuell in dem Prozess befindlichen Entitäten anzeigt. Dabei wird bei allen Bausteinen, außer den Prozess-Bausteinen, immer gezählt, wenn eine Entität den Baustein verlässt. Dabei wird bei jedem Baustein jeder Ausgang einzeln gezählt, so hat der Baustein „*Decide 1*“ im zuvor abgebildeten Beispiel aufgrund von

zwei Ausgängen auch zwei Zählvariablen. Ist ein Block auf der Arbeitsoberfläche platziert, so kann per Doppelklick auf einen Baustein ein Bearbeitungsfenster für diesen geöffnet werden.

Abbildung 13: Bearbeitungsfenster für "Process 1"

Die Abbildung 13 zeigt das Bearbeitungsfenster für den Prozess-Baustein „Process 1“. In der oberen Hälfte können Name, Typ (*Type*) und die Logik (*Logic*) des Bausteins festgelegt werden. Der Name ist in diesem Fall „Process 1“ und der Typ ist „Standard“. Neben diesem Typ gibt es noch den Typ *Submodel*, welcher benutzt werden kann, wenn der Baustein teil eines Untermodells ist. Ein Untermodell, im

Folgenden Submodel genannt, fasst

mehrere Bausteine zusammen und verringert dadurch die Größe des übergeordneten Modells. Als Logik ist in dem Beispiel aus der Abbildung *Delay*, also eine einfache Verzögerung ausgewählt. Dadurch wird jede einfließende Entität um den gewählten Wert verzögert, bevor diese den Baustein wieder verlässt. Wählt man hier *Seize Delay Release*, *Seize Delay* oder *Delay Release*, so erscheint über dem Baustein eine horizontale Linie, welche mit einer vertikalen Linie endet (Abbildung 12, (4)). Diese stellt eine Warteschlange dar. In der unteren Hälfte des Bearbeitungsfensters kann in diesem Fall der Typ der Verzögerung mit den entsprechenden Eigenschaften festgelegt werden. Arena bietet neben der Möglichkeit einer konstanten Verzögerung auch verschiedene statistische Verteilungen an. Im Beispiel ist eine Normalverteilung (*Normal*) mit einer Verzögerung von einer Stunde und einer Standardabweichung (*Std Dev*) von 0,2 Stunden gewählt. Nachdem alle Ein-, beziehungsweise Ausgänge mit jeweils einem Anderen verbunden sind, kann die Simulation mit dem Button „Start“ (*Go*) aus der Werkzeugleiste gestartet werden. Sollte ein Problem bei einem Objekt vorliegen, so weist Arena mit einer Fehlermeldung darauf hin und bricht die Simulation ab, beziehungsweise startet diese nicht. Ist kein Endpunkt der Simulation definiert, so kann diese mit einem Klick auf den Button „Ende“ (*End*) beendet werden. Daraufhin steht abschließend in der Projektleiste unter dem, bereits erwähnten, Punkt Bericht eine Auswahl an statistischen Auswertungen zur Verfügung.

## 6 Simulationsbeispiel 1: Laufradfertigung

Um die beiden Simulationswerkzeuge besser vergleichen zu können, werden sie auf zwei verschiedene Beispiele angewendet. Dazu wird das Beispiel zuerst erläutert, bevor auf seine Abbildung in GD.findi und Arena eingegangen wird. Danach werden die Vorgehensweise und die Erkenntnisse der Modellierung in den beiden Programmen gegenübergestellt.

Das Beispiel in diesem Kapitel behandelt die Fertigung von Laufrädern und wird aus der Projektarbeit von Frau Baller und Frau Nerke übernommen (vgl. [BN16]). Dies gewährleistet eine Vergleichbarkeit mit den dort untersuchten Simulationswerkzeugen. Im nächsten Kapitel wurde mit der Fertigung eines Heckspoilers ein neues Beispiel entwickelt. Durch die Modellierung der Systeme zeigt sich, ob die beiden Werkzeuge zur Simulation von selbststeuernden Logistiksystemen geeignet sind.

### 6.1 Erklärung

Das folgende Anwendungsbeispiel befasst sich mit den Fertigungs- und Montageprozessen eines Laufrades der Firma Wishbone Design Studio. In dem abzubildenden System werden verschiedene Komponenten in mehreren Arbeitsschritten zu dem Endprodukt montiert. Das Laufrad besteht aus sechs Einzelteilen, von denen die meisten aus Holz gefertigt sind. Dazu zählen der Lenker, die Gabel, der Rahmen und die Sattelstütze. Die übrigen Teile sind die beiden Räder, von denen sich eins aus einem Gummireifen und einer Metallfelge zusammensetzt.



Abbildung 14: Darstellung des Laufrades "Wishbone Bike 2in1 - original" ([WIS1])

Der Produktionsprozess des Laufrades kann durch die Abbildung 15 beschrieben werden und wird in drei Arbeitsschritte gegliedert, nämlich der Holzteilefertigung, der Radmontage und der Laufradmontage.

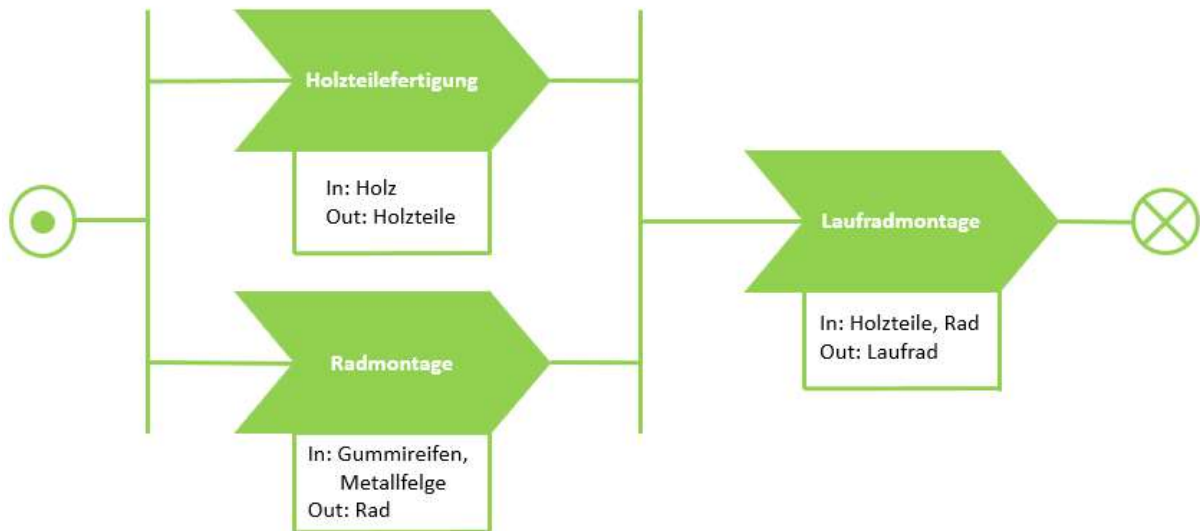


Abbildung 15: Modifizierte Prozessdarstellung inklusive Input/ Output nach [BN16]

Im ersten Arbeitsschritt, der Holzteilefertigung, werden der Lenker, die Gabel, der Rahmen und die Sattelstütze aus Holz gefertigt. Der Rohstoff Holz wird vom Lager bereitgestellt und trifft im Abstand von 180 Sekunden an der Bearbeitungsstation ein. Bei der Fertigung werden die Holzteile zugeschnitten, entgratet und mit den entsprechenden Bohrungen versehen, welche zusätzlich gesenkt werden. Diese Holzteile verlassen gesammelt diesen Arbeitsplatz. Zugleich wird an der zweiten Station, der Radmontage, ein Gummireifen auf eine Metallfelge gezogen. Dazu werden die benötigten Produkte vom Lager in einem Abstand von 120 Sekunden bereitgestellt. Das fertige Gummirad wird daraufhin zur nächsten Station transportiert. Der letzte Arbeitsschritt besteht aus der Endmontage des Laufrades. Hier werden die Einzelteile aus Holz und die Räder zum fertigen Produkt zusammengefügt. Die benötigten Teile werden von den jeweiligen Stationen geliefert. Nach der Montage wird das Laufrad zur Qualitätskontrolle transportiert, die in diesem System die Senke darstellt. Der Transport zwischen den Stationen geschieht durch die modularen Förderelemente des MINIS-Projekts, dessen Transportgeschwindigkeit 1 m/s beträgt. Es wird die vereinfachende Annahme getroffen, dass die Verfügbarkeit von benötigten Teilen, wie Lenkergriffe, Sitzpolster, Achsen, Schrauben oder Muttern, an den jeweiligen Bearbeitungsstationen



gesichert ist. Außerdem werden Abfallprodukte, die während der Fertigung anfallen, an den Stationen gesammelt und am Ende des Tages entsorgt ([BN16], S.23). Es wird davon ausgegangen, dass die Puffer an den jeweiligen Arbeitsstationen unendlich groß sind.

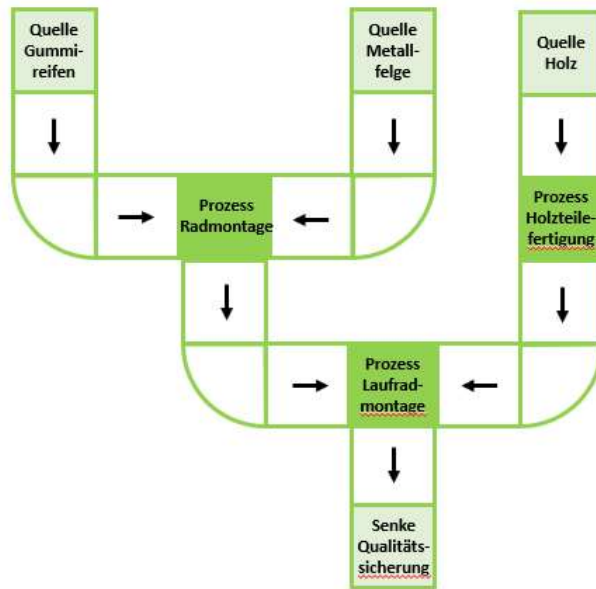


Abbildung 16: Beispielhaftes Layout der Produktionsanlage ([BN16]), S.24)

Die Bearbeitungszeiten für die einzelnen Arbeitsschritte ist Tabelle 3 zu entnehmen. Die Holzteilefertigung wird von einer Maschine übernommen. Aus diesem Grund kann die benötigte Zeit als konstant angenommen werden. Dahingegen findet die Montage der Reifen und des kompletten Laufrades durch Mitarbeiter statt, weshalb die Bearbeitungszeit variieren kann. Deshalb wird diese als normalverteilt angenommen ([BN16], S.25).

Tabelle 3: Bearbeitungszeiten für das Anwendungsbeispiel der Laufradfertigung

Arbeitsschritt	Bearbeitungszeit [s]	Wahrscheinlichkeitsverteilung	Mittelwert [s]	Standardabweichung [s]
Holzteilefertigung	180	konstant	-	-
Radmontage (je Rad)	-	Normalverteilung	72	30
Laufradmontage	-	Normalverteilung	375	60

## 6.2 Abbildung in GD.findi

Das Anwendungsbeispiel lässt sich problemlos in GD.findi modellieren. Wie in Kapitel 4.3 beschrieben, teilt sich ein Modell in einen Raumplan und einen Prozessplan. Um die Implementierung des in GD.findi zu verdeutlichen, werden auf diese Elemente eingegangen. Bei der Erstellung des Raumplans (siehe Abbildung 17) werden lediglich die verschiedenen Stationen erschaffen, die bei dem Produktionsprozess existieren. Eine Unterscheidung in Quelle, Prozess, Fördererelement oder Senke findet erst später statt. Es ist anzumerken, dass die in der Abbildung 17 befindlichen Stationen „Gummireifen“, „Metallfelgen“ und „Holz“ die Quellen der jeweiligen Materialien sind. Die „Radmontage“, „Laufradmontage“ und „Holzteilefertigung“ stellen die Prozesse dar und die „Qualitätskontrolle“ steht für die Senke. Im Raumplan ließe sich die Puffergröße einer Station einstellen, wenn diese nicht als unendlich hoch angenommen werden würde. Die Stationen werden mithilfe von Anschlüssen, Knoten und Pfeilen verbunden. Es ist bei GD.findi nicht möglich das Fördererelement auf eine andere Art abzubilden. In diesem Beispiel wurden nur Linienförderer verwendet.

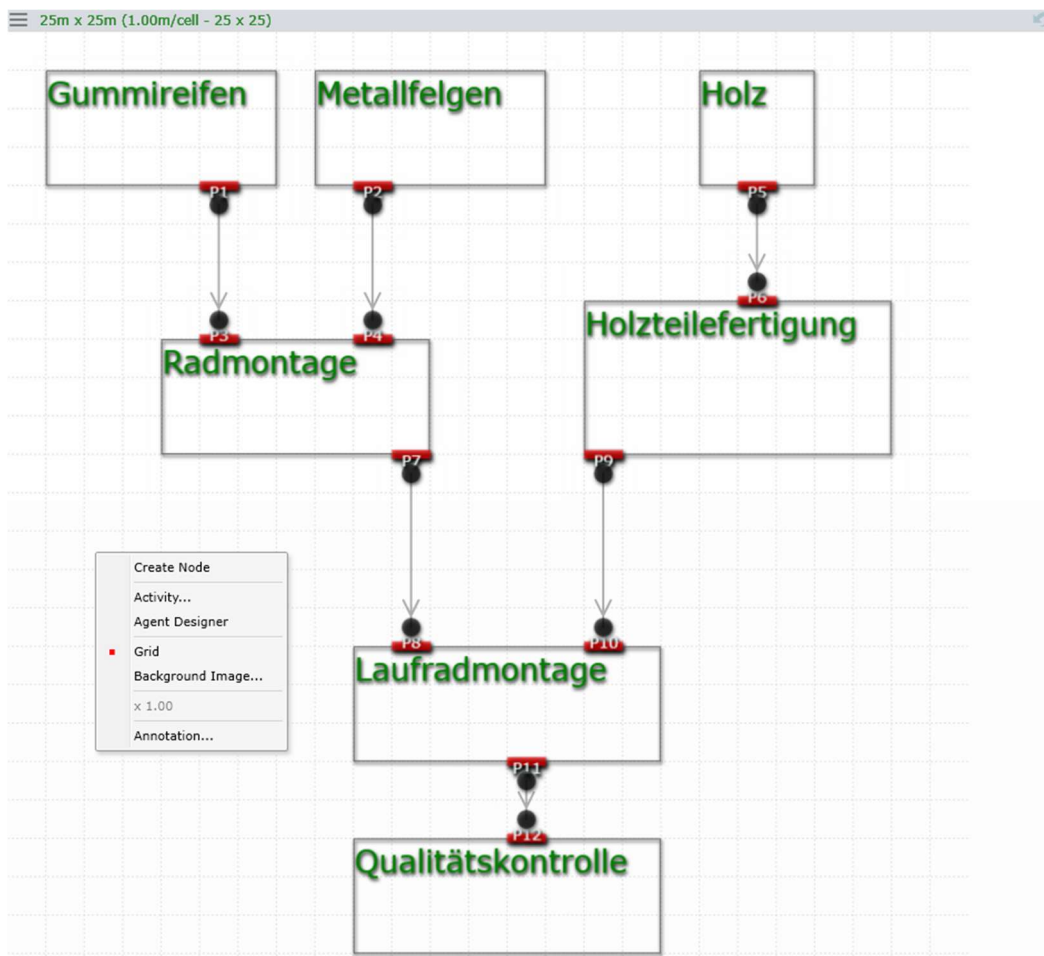


Abbildung 17: Raumplan für Anwendungsbeispiel Laufradfertigung

Nach der Konstruktion des Raumplans wird der Prozessplan angefertigt (siehe Abbildung 18). Dort wird für jede zuvor aufgestellte Station auch ein Prozess angelegt. Dabei steht ein Prozessplan für einen Produkttypen, in diesem Beispiel das Laufrad. Soll beispielsweise ein Laufrad mit unterschiedlichen Rädern die Stationen durchlaufen, muss für diesen Typ ein neuer Prozessplan angelegt werden. Bei der Beschreibung der Prozesse müssen die Teile vermerkt werden, die für den Prozess benötigt werden (Input parts) und diesen verlassen (Output parts). Die eingetragene Bearbeitungsdauer ist konstant und lässt sich erst später in eine Normalverteilung umändern. Um eine Quelle abzubilden, werden die entsprechenden Bauteile in dem Menü der jeweiligen Quelle der Zeile „Local In“ hinzugefügt, da einer Quelle kein Prozessschritt vorangeht. Das gleiche Bauteil wird auch dem nachfolgenden Prozess hinzugefügt, indem es in die Zeile von diesem eingetragen wird. Soll zwischen der Bereitstellung der Bauteile eine definierte Zeitspanne liegen, so lässt sich dies realisieren, indem man der Quelle eine Bearbeitungszeit hinzufügt. Eine Senke wird analog abgebildet.

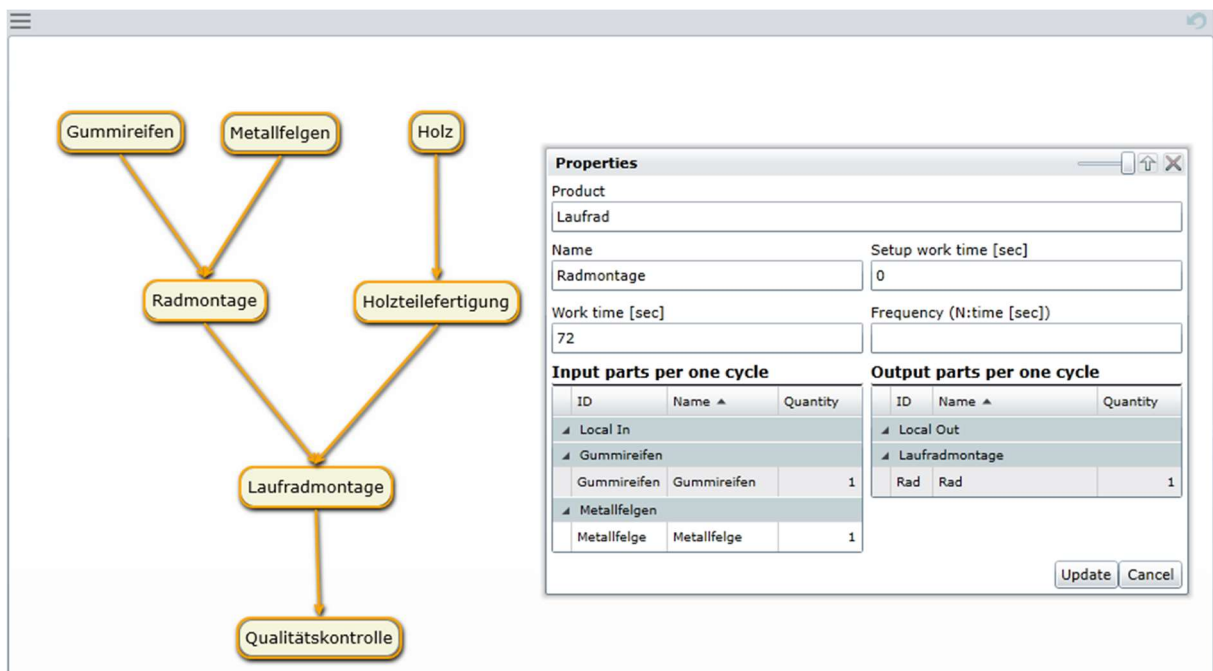


Abbildung 18: Prozessplan für Laufradfertigung mit Beschreibung eines Prozesses

Nach der Festlegung der beiden Pläne werden diese per Drag and Drop miteinander verbunden. Es folgt die Einrichtung der Transportelemente. Dafür wird auf der Seite des Raumplans der Agent Designer geöffnet und eine neue Klasse mit dem Model „Conveyor\_Async“ erstellt (siehe Abbildung 19, S.34). Des Weiteren werden alle im Raumplan bestehenden Wege hinzugefügt, um den Bereich zu definieren, in dem das Förderelement gilt.

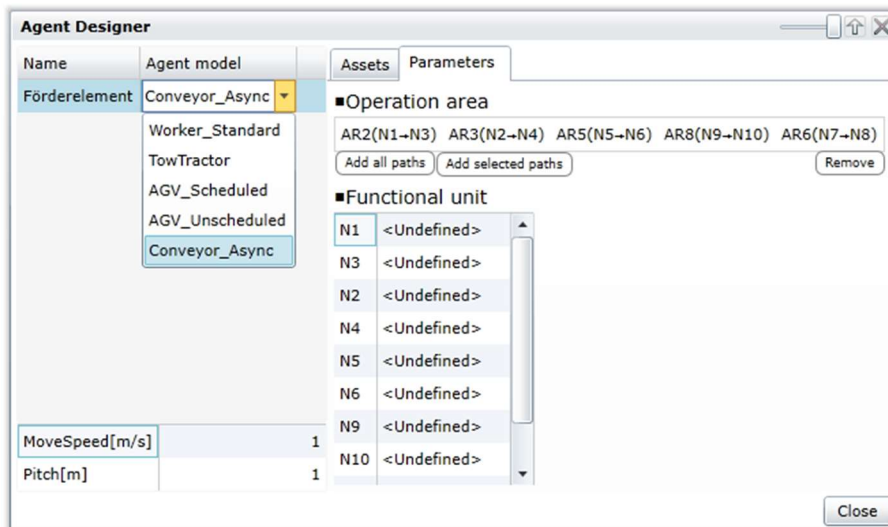


Abbildung 19: Agent Designer

Außerdem müssen die Routen definiert werden, auf denen sich die Entitäten bewegen. Dies geschieht im Menüunterpunkt „Activity“. Dort muss jeder Pfad zwischen zwei Stationen manuell durch das Verknüpfen von Anschlüssen und Punkten erstellt werden (siehe Abbildung 20). Außerdem kann unter dem Reiter „Station Activity“ die Wahrscheinlichkeitsverteilung einer Bearbeitungszeit verändert werden. Mit diesem Schritt wird die Modellerstellung abgeschlossen und die Simulation kann gestartet werden. Auf die Ergebnisse wird nicht weiter eingegangen, da dessen Untersuchung nicht zur Aufgabe gehörte.

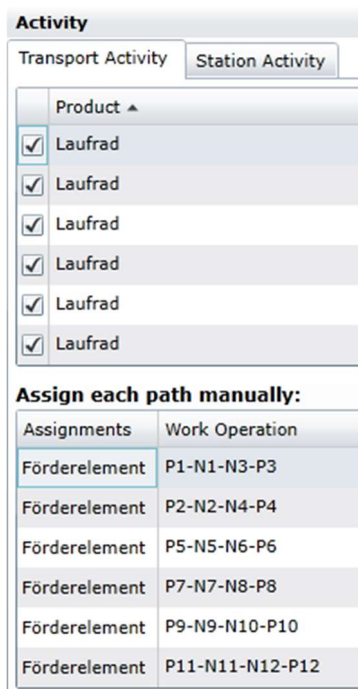


Abbildung 20: Verknüpfung der Anschlüsse mit Punkten zu Routen

### 6.3 Abbildung in Arena

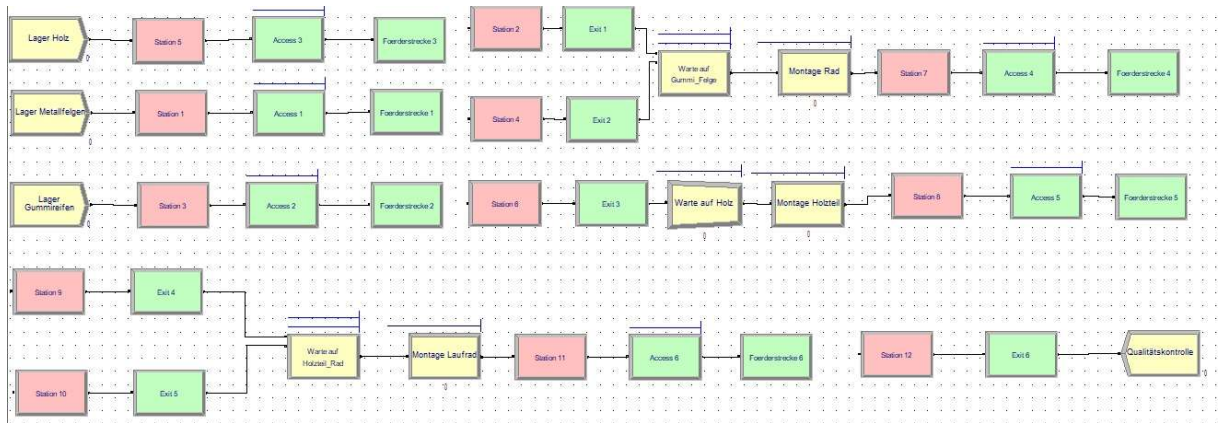


Abbildung 21: Modell der Laufradfertigung in Arena

Wie man ein Modell in Arena erstellt wurde bereits in Kapitel 5 erläutert, deshalb soll in diesem und dem folgenden Kapitel auf die Parameter der einzelnen Bausteine eingegangen werden. Die Abbildung 21 zeigt die Modellierung des in Kapitel 6.1 beschriebenen Systems mit Arena. Auffällig sind hier zum einen die bereits erwähnte, und einem Flussdiagramm ähnliche, Struktur und zum anderen die Besonderheit, dass nicht alle Bausteine von den Lagern zur Qualitätskontrolle mit einer Linie verbunden sind. Das Aussehen des Modells, welches eher einem Fluss als der tatsächlichen räumlichen Anordnung ähnelt ist der Tatsache geschuldet, dass in Arena nur die benötigte Zeit für einen Weg und nicht der Weg selbst für die eigentliche Simulation von Bedeutung ist. Wie bereits in Kapitel 5 erwähnt ist die Darstellung der räumlichen Anordnung und Wege zwar möglich, da die grafische Darstellung in Arena allerdings sehr aufwändig und für die Simulationsergebnisse irrelevant ist wurde auf diese bei den Simulationsbeispielen verzichtet. Denn Arena stellt zwar einige Bibliotheken zu Verfügung, allerdings muss die grafische Darstellung anschließend noch mit dem eigentlichen Modell verknüpft werden. Die zweite Auffälligkeit lässt sich ebenfalls damit erklären, dass der Weg in Arena irrelevant ist. Denn bei den Lücken zwischen einigen Bausteinen handelt es sich um Förderer, in diesem Fall Stetigförderer. Die Funktionsweise soll an dem Beispiel des Förderers vom Lager für Holz zur Montagestation der Holzteile erklärt werden. Diese lässt sich allerdings auf die vier anderen Förderer übertragen. Um einen Förderer, wie in Abbildung 22 auf der folgenden Seite gezeigt, darzustellen sind insgesamt fünf Bausteine nötig. Zu Beginn muss die Entität eine *Station* passieren. Danach kommt sie in einen Baustein vom Typ *Access*.

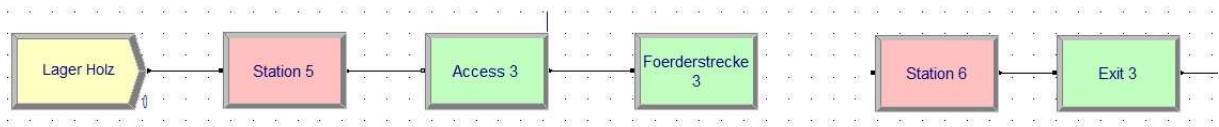


Abbildung 22: Modell einer Förderstrecke

Hier muss im Bearbeitungsfenster (siehe Anhang C) definiert werden welcher Förderer (*Conveyor Name*) der Entität zugeordnet wird und wie viele Plätze (*# of Cells*) diese dort belegt. In diesem Fall wurde im Bearbeitungsfenster „Foerderer 3“ ausgewählt und eine eingehende Entität wird einen Platz belegen. Des Weiteren wird hier eine Warteschlange erzeugt, in welcher die Entität wartet sofern der Förderer belegt sein sollte. Das geschieht allerdings in diesem Beispiel nicht. Im folgenden Baustein vom Typ *Convey* muss der Förderer, hier Foerderer 3, sowie die Zielstation (*Destination Type = Station*), hier Station 6 im Bearbeitungsfenster (siehe Anhang D) ausgewählt werden. Zuletzt muss die Zielstation, welche nicht über eine Linie mit dem Baustein *Convey* verbunden ist, mit einem Baustein vom Typ *Ausgang (Exit)* verbunden werden und im Bearbeitungsfenster (siehe Anhang E) die Anzahl der nun frei zu gebenden Plätze (*# of Cells*) angegeben werden. Es ist sinnvoll hier den gleichen Wert wie in dem *Access*-Baustein zu verwenden, da die Entität ihre Größe während des Transfers nicht verändert und keine Zellen aus dem Förderer entfernt. Als nächstes muss mit einem Klick auf das Element *Segment* in der Projektleiste die Länge des Förderers festgelegt werden. Mit einem Klick auf „0 rows“ öffnet sich das in Abbildung 23 gezeigte Fenster. Hier wurde als nächste Station (*Next Station*), Station 2 ausgewählt. Als Länge (*Length*) wurde 3 ausgewählt, da dieses Förderband eine Gesamtlänge von 3m haben soll. Wählt man in der Projektleiste *Convey* aus, so kann neben anderen Parametern auch die Geschwindigkeit (*Velocity*) in Längeneinheit pro Zeiteinheit des Förderers festgelegt werden.

Next Stations		
	Next Station	Length
1	Station 6	3

Double-click here to add a new row.

Abbildung 23: Bearbeitungsfenster *Conveyor.Segment*

Auch für die Gestaltung der Montagestationen sind in Arena meistens mehr als ein Baustein nötig. So wurde die Montagestation für die Räder, sowie die Endmontagestation für das Laufrad aus den Bausteinen *Match* und *Process* erstellt. Der Grund dafür ist, dass der Prozess-Baustein eine unbekannte Entität nur verzögert. Die eigentliche neue Entität wird bereits im *Match*-Baustein erzeugt. Dazu wird als „*Representative Entity Type*“ die neue Entität ausgewählt. Diese neue Entität verlässt den Baustein, sobald beide Warteschlangen, also zum Beispiel bei der Endmontagestation für das Laufrad, die Warteschlange für die Räder und die Warteschlange für die Holzteile mit mindestens einer Entität belegt sind. Bis zum Verlassen der neuen Entität verweilen die anderen Entitäten in der Warteschlange. Die Montagestation der einzelnen Holzstücke zu einem Holzteil wurde mittels einer Kombination von *Batch* und *Process* realisiert, da alle eingehenden Entitäten vom selben Typ sind. Es wird hier nur gewartet bis vier Entitäten vom Typ „Holz“ in der Warteschlange sind, bevor eine Entität vom Typ „Holzteil“ den Baustein verlässt und im Prozessbaustein verzögert wird. Alles in Allem konnte dieses Beispiel gut in Arena umgesetzt werden. Zwar war zu Beginn eine Eingewöhnungsphase nötig um die Funktionsweise des Programms zu verstehen, war dies jedoch geschafft so gestaltete sich die eigentliche Modellierung des Beispiels nicht zu schwierig. Das Beispiel kann durch die Verwendung von Submodels übersichtlicher gestaltet werden. Was das ist und wie der Umgang mit diesen funktioniert soll allerdings an Beispiel 2 erläutert werden um einen Vergleich zu schaffen.

#### 6.4 Vergleich der Simulationstools im Hinblick auf Simulationsbeispiel 1

Vergleicht man nun die Simulationsprogramme im Hinblick auf das erste Beispiel, so fallen direkt die unterschiedlichen Arten der Modellierung auf. Denn bei der Modellerstellung der Laufradfertigung mit GD.findi wird zwischen dem Raumplan und dem Produktionsprozess unterschieden. Es werden zuerst alle Arbeitsstationen im Raumplan erstellt und verbunden. Danach wird jeder Arbeitsschritt im Produktionsprozess definiert. Als nächstes werden diese Elemente verknüpft und den Verbindungen der Stationen werden die Eigenschaften eines Fördererelementes zugewiesen. Bei Arena werden dagegen erst die bereits vorgefertigten Bausteine aus der Projektleiste in den Arbeitsbereich gezogen. Anschließend werden diese verbunden und die einzelnen Prozesse, welche Arbeitsstationen darstellen definiert. Im Gegensatz zu GD.findi können die Verknüpfungen hier keine Fördererelemente darstellen. Dazu

sind weitere Bausteine nötig. Hierbei wurde direkt ersichtlich, dass Arena über wesentlich mehr Möglichkeiten zur Modellierung verfügt. Dies hat den Nachteil, dass sich der Benutzer vor dem ersten Modellieren mit dieser Vielfalt auseinandersetzen muss. Wohingegen mit GD.findi schnell ein erstes Modell erstellt werden kann. Ein großer Vorteil dieser Vielfalt ist allerdings, dass selbst sehr komplexe Systeme in Arena modelliert werden können. Das gewählte Beispiel konnte allerdings auch in GD.findi ohne Probleme oder weitere Vereinfachungen abgebildet werden. Wird die Darstellung des Beispiels in den jeweiligen Simulatoren betrachtet, so lassen sich auch hier große Unterschiede erkennen. Während in GD.findi durch die Modellierung bereits ein Raumplan besteht, so muss dieser nachträglich in Arena erstellt, beziehungsweise eingelesen werden. Arena ist im Aufbau der Modellierung eher mit dem Prozessmodell von GD.findi zu vergleichen, wobei jedoch eine Arbeitsstation bei Arena aus mehreren Bausteinen bestehen kann, was hier zum Beispiel bei den Montagestationen der Fall ist. Zur Beschreibung vom Prozess werden daher in Arena erheblich mehr Elemente benötigt, als es bei GD.findi der Fall ist. Dies kann schnell zur Unübersichtlichkeit führen, wie es zum Beispiel bei der Modellierung der Förderbänder der Fall ist. Bei gleichem Endergebnis ist für dieses Beispiel in Arena zudem ein erhöhter Arbeitsaufwand notwendig. Auffällig war, dass bei der Simulation mit Arena der Fluss der Entitäten durch das Modell animiert dargestellt wurde, welcher eine Fehlersuche vereinfacht. Des Weiteren gestaltet dieser Vorteil die Prozessoptimierung deutlich einfacher, da Stautellen und erste Auffälligkeiten schon während der Simulation erkannt werden können. Diese animierte Darstellung der Entitäten ist in GD.findi nicht vorhanden.



## 7 Simulationsbeispiel 2: Fertigung eines Heckspoilers

Nachdem im ersten Beispiel das Hauptaugenmerk auf dem Zusammenführen mehrerer Entitäten an verschiedenen Bearbeitungsstationen lag, soll sich das zweite Beispiel vornehmlich mit der Verteilung von gleichen Entitäten auf verschiedene Transportbänder befassen. Dazu wird im Folgenden das zweite Simulationsbeispiel vorgestellt und anschließend die Abbildung in GD.findi und Arena betrachtet. Abschließend sollen die Abbildungen miteinander verglichen und Unterschiede, sowie Stärken und Schwächen der Softwaretools herausgearbeitet werden.

### 7.1 Erklärung

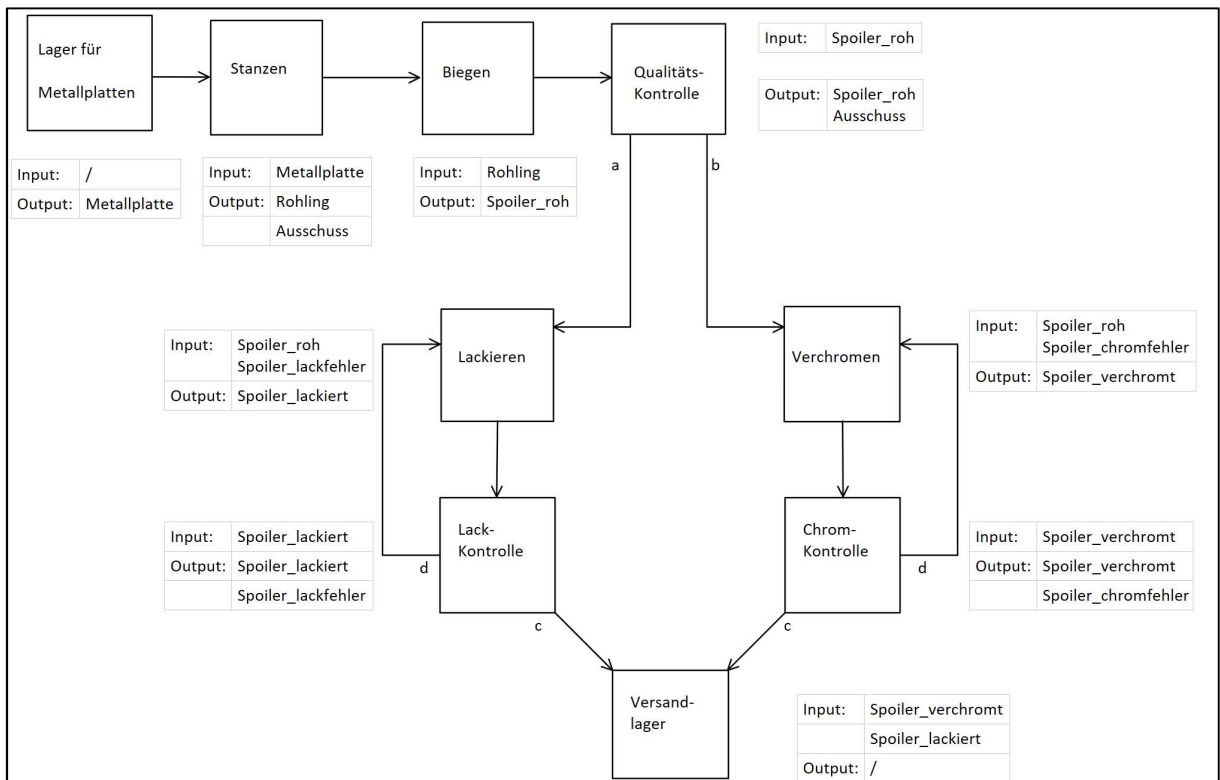


Abbildung 24: Fertigungslinie Heckspoiler

Das zweite Simulationsbeispiel stellt eine Fertigungslinie für Heckspoiler von funkferngesteuerten Modellautos der Firma ChriDo dar. Die Firma produziert einen Heckspoiler, welcher sowohl als lackierte, als auch als verchromte Variante erhältlich ist. Abbildung 24 zeigt eine komplett autonome Fertigungslinie für den Heckspoiler. Der Materialfluss startet in dem Lager für Metallplatten. Von diesem kommen die Metallplatten in eine Stanzmaschine, welche den Rohling für den Heckspoiler aus der Metallplatte stanz.

Der dabei entstehende Ausschuss wird an der Maschine gesammelt und nach Schichtende geleert. Von dort aus kommt der Rohling in eine Biegemaschine. Diese bringt den Rohling per Biegeverfahren in die Form des Spoilers. Eine anschließende Qualitätskontrolle sortiert fehlerhafte Spoiler aus. Nicht-fehlerhafte Spoiler kommen nach der Qualitätskontrolle entweder in die Lackiererei (a) oder in die Verchromung (b). Dabei ist jeder Spoiler ab der Qualitätskontrolle mit einem RFID-Tag versehen. Die Verteilung auf den Transportbändern erfolgt ab hier über mit RFID-Scannern ausgestatteten Drehtischen. Nach der Lackierstation erfolgt an einer Kontrollstation ob der Lack fehlerfrei aufgetragen wurde. Ist dies der Fall, so wird der Spoiler in ein Versandlager (c) transportiert. Ist die Lackschicht fehlerhaft, so wird der Spoiler wieder der Lackiererei (d) zugeführt und erneut lackiert. Das trifft bei der Lackierung auf 15% der bearbeiteten Spoiler zu. Bei der Verchromung sind 10% der verchromten Spoiler fehlerhaft. Der Aufbau der Fertigungslinie über die Verchromung ist identisch zu dem der Lackierung. Die einzelnen Bearbeitungsstationen sind dabei auf der gesamten Fertigungslinie über Stetigförderer miteinander verbunden. Für die Analyse der Simulationstools soll im Folgenden nur der Fertigungsprozess ab der Qualitätskontrolle modelliert werden, da die ersten Produktionsschritte trivial sind und in diesem Beispiel, wie bereits erwähnt, der Einsatz von Drehtischen, beziehungsweise der Verzweigung von Stetigförderern im Vordergrund stehen soll. Es erfolgt die Annahme, dass die Spoiler die Qualitätskontrolle konstant in einem Abstand von hundert Sekunden verlassen. Die angenommenen Bearbeitungszeiten für das Verchromen, dem Lackieren und der beiden Kontrollstationen sind frei erfunden und werden in folgender Tabelle 4 dargestellt.

Tabelle 4: Bearbeitungszeiten und Verteilungen der Arbeitsstationen bei der Heckspoilerfertigung

Station	Bearbeitungszeit	Verteilung	Standardabweichung
Verchromung	120s	normalverteilt	10s
Lackiererei	80s	normalverteilt	8s
Kontrollstation nach Verchromung	20s	konstant	/
Kontrollstation nach Lackierung	20s	konstant	/

Die Längen der Stetigförderer sind in Tabelle 5 auf folgender Seite festgehalten. Dabei wird davon ausgegangen, dass ein Heckspoiler in der Länge einen Platz von einem Meter auf dem Förderer benötigt. Die Transportgeschwindigkeit soll 1 m/s betragen. Es wird zudem davon ausgegangen, dass die fehlerhaft lackierten, beziehungsweise verchromten Teile mit einem

Auswerfer auf das Fördermittel für die Rückführung befördert werden. Die dafür benötigte Zeit ist als vernachlässigbar klein anzunehmen. Bei dem Drehtisch direkt hinter der Qualitätskontrolle ist eine durchschnittliche Zeit von zwei Sekunden hinreichend genau.

Tabelle 5: Längen der Transportförderer

Transportband [Anfangsstation – Endstation]	Länge [in m]
Qualitätskontrolle - Drehtisch	1
Drehtisch – Verchromen	1
Verchromen – Chromkontrolle	2
Drehtisch – Lackieren	6
Lackieren - Lackkontrolle	2
Chromkontrolle – Versandlager	5
Lackkontrolle – Versandlager	5

## 7.2 Abbildung in GD.findi

Im Vergleich zum ersten Anwendungsbeispiel war diese Implementierung durch die Abbildung des Drehtisches und der Rückführung komplexer. Im Folgenden bauen die Umsetzung des Beispiels und die dabei gewonnenen Erkenntnisse auf Kapitel 6.2 auf.

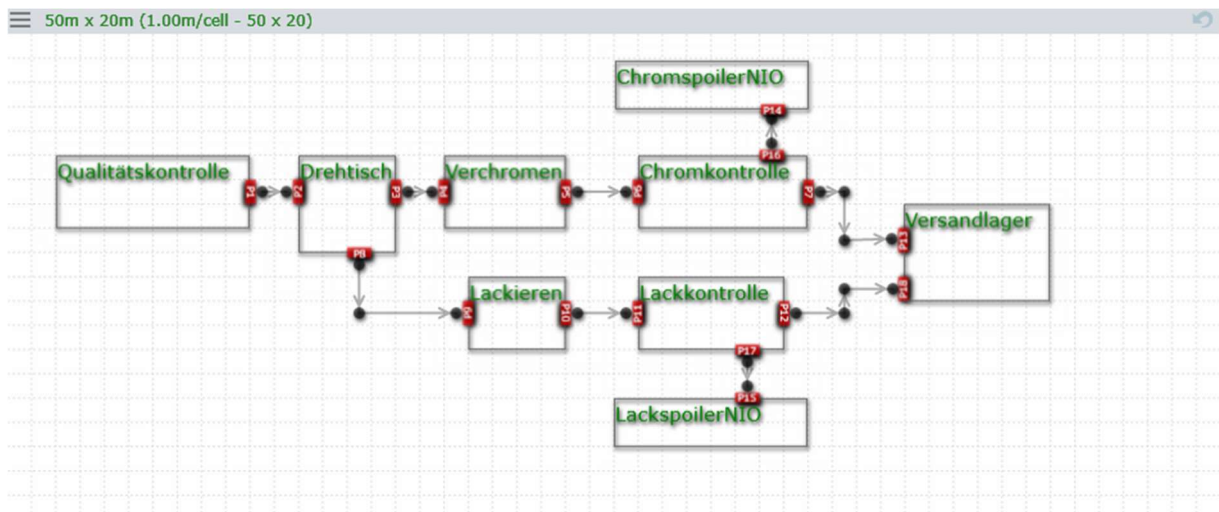


Abbildung 25: Raumplan für Anwendungsbeispiel Heckspoilerfertigung

Anhand des Raumplans ist erkennbar, dass eine Rückführung in GD.findi nicht realisiert werden kann (siehe Abbildung 25). Die Ursache liegt darin, dass für jedes unterschiedliche Produkt, das durch die Anlage läuft, ein eigener Prozessplan erstellen werden muss. Der

Wechsel einer Entität von einem Produkttypen zu einem anderen während sie das Modell durchläuft, ist ausgeschlossen. Aus diesem Grund lässt sich bei dem Prozess des Lackierens, bzw. des Verchromens nicht bestimmen, zu welcher Wahrscheinlichkeit die Entität in Ordnung ist und zu welcher sie fehlerbehaftet ist. In diesem Beispiel müssen also vier verschiedene Produkttypen definiert werden (siehe Abbildung 26).

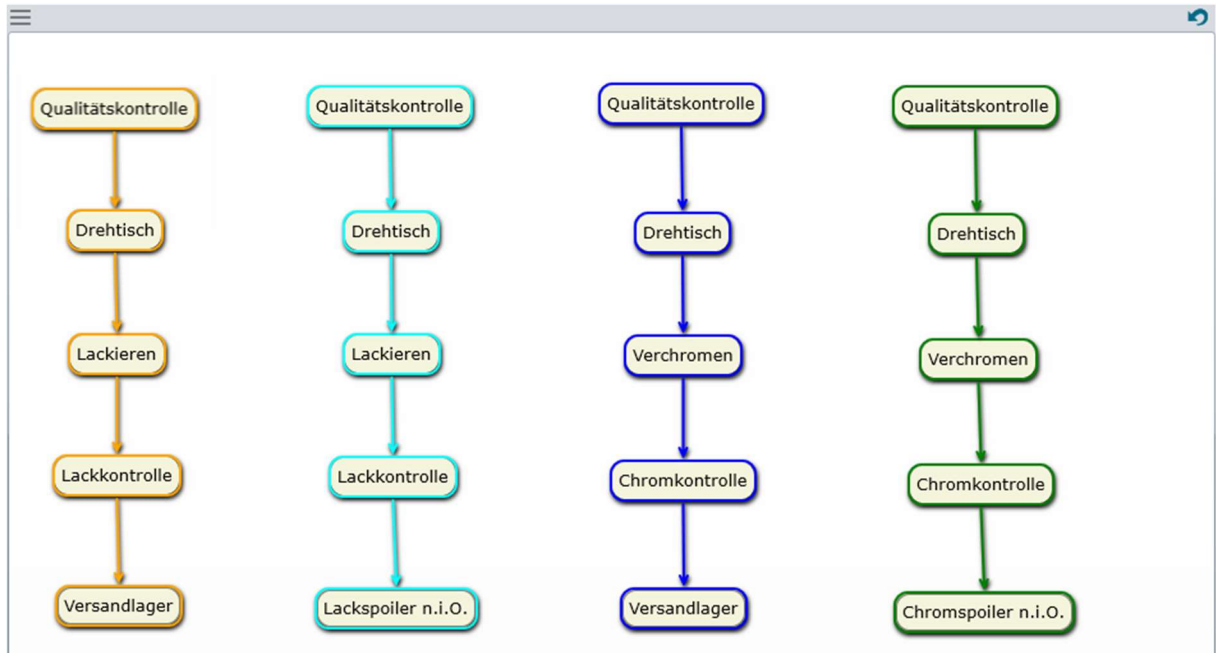


Abbildung 26: Prozessplan für Anwendungsbeispiel Heckspoilerfertigung

Wenn nun eine Rückführung definiert wird, bewegt sich die fehlerhafte Entität in einer Endlosschleife, da sie durch nochmaliges Durchlaufen des Lackierungs-/Verchromungsprozesses nicht zum Produkttyp der fehlerfreien Spoiler gehören kann und deshalb immer wieder zurückgewiesen wird. Als Vereinfachung wurde deshalb angenommen, dass fehlerhafte Bauteile in einen Behälter („LackspoilerNIO“ und „ChromspoilerNIO“) gesammelt werden. Die Anzahl der Produkttypen steigt mit der Anzahl der im System befindlichen Qualitätskontrollen.

Die Definition von vier verschiedenen Produkttypen bringt ein weiteres Problem mit sich. Bevor die Simulation gestartet wird, wird das Produktionsziel und die Fertigungsreihenfolge bestimmt, in der die verschiedenen Typen abgearbeitet werden sollen (siehe Abbildung 27, S.43). Bei einer gegebenen Fehlerrate eines Prozesses (z.B. Lackieren), müsste die Verteilung der einzelnen Typen im Vorfeld bestimmt werden und die Reihenfolge, in der sie produziert werden.

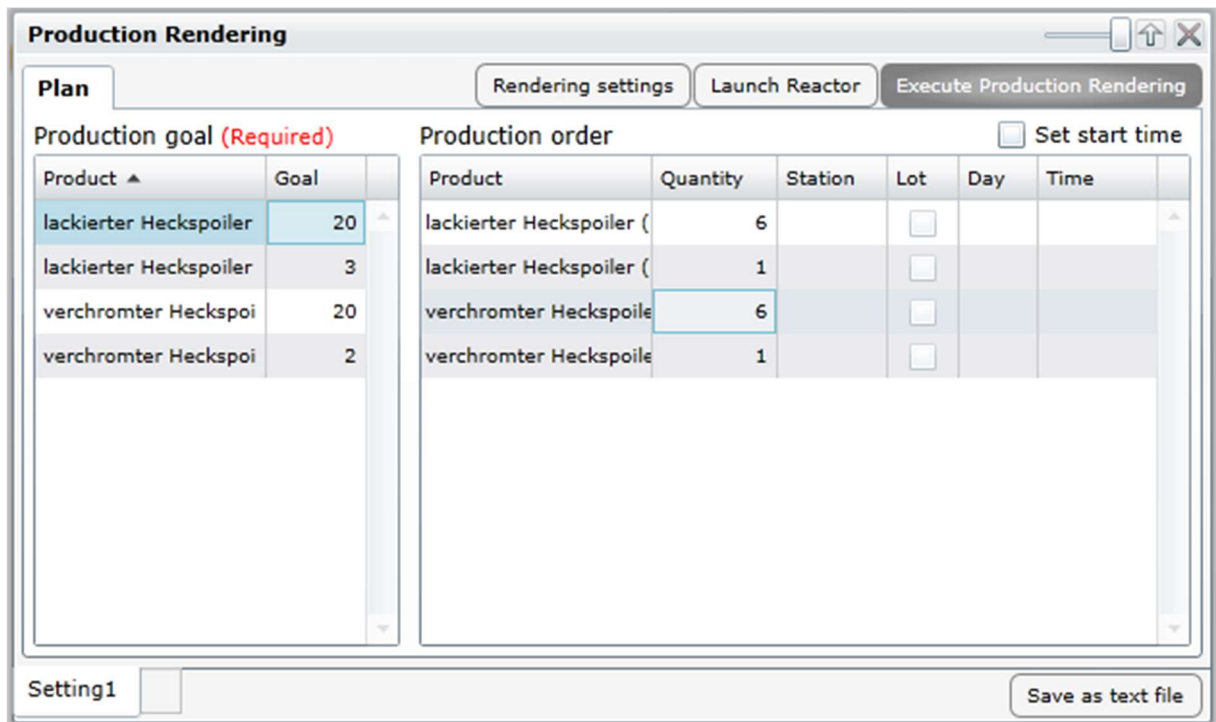


Abbildung 27: Produktionsplanung der Heckspoilerfertigung

In diesem Beispiel wurden weitere Fördererelemente des MINIS-Projekts verwendet. Ein Drehtisch lässt sich durch eine Station abbilden, die mehrere Ausgänge hat. Für den zugehörigen Prozess ist eine Bearbeitungszeit eingetragen, die der Drehzeit des Tisches entspricht (siehe Abbildung 28). Bei diesem Beispiel hat der Drehtisch nur einen möglichen Ausgang, da der Weg der Entität durch den Produkttyp vorgeschrieben ist. Ein Auswerfer ließe sich analog abbilden, mit dem Unterschied, dass man die Bearbeitungszeit vernachlässigen kann.

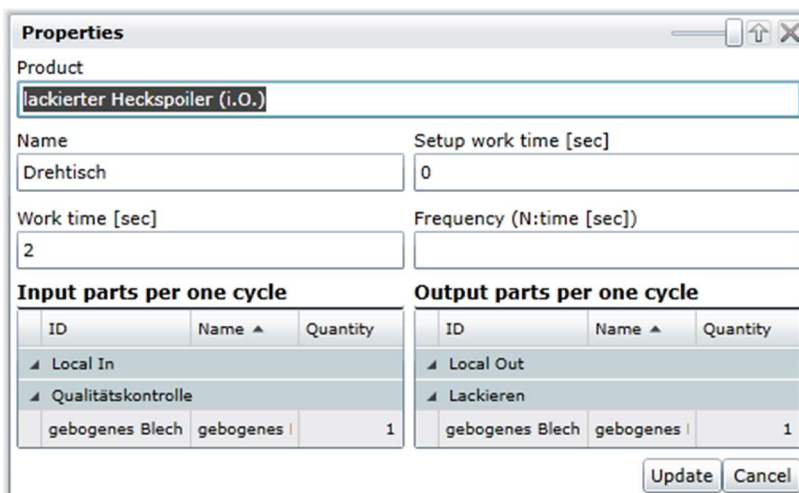


Abbildung 28: Prozessbeschreibung eines Drehtisches

Da sich die Förderelemente zwischen den Stationen aus Pfeilen zusammensetzen, die als Start- und Endpunkt jeweils einen Punkt haben, ist es die einfachste Lösung eine Kurve durch einen rechten Winkel abzubilden. Es besteht zwar die Möglichkeit, mithilfe von zahlreichen Punkten die Bahn einer Kurve zu formen, allerdings steht dieser Aufwand in keinem Verhältnis zu dem dadurch gewonnenen Nutzen.

### 7.3 Abbildung in Arena

Da das zweite Beispiel, im Gegensatz zu dem ersten Simulationsbeispiel aus Kapitel 6, durch den Einsatz des Drehtisches oder dem Auswerfer ein komplexeres Fördersystem besitzt, wurden bei der Modellierung von diesem Beispiel Submodels verwendet. Auf eine grafische Darstellung des Systems wurde erneut verzichtet, da diese für das Simulationsergebnis unerheblich ist. Abbildung 29 zeigt die in Arena modellierte Fertigungslinie des in 7.1 beschriebenen Beispiels.

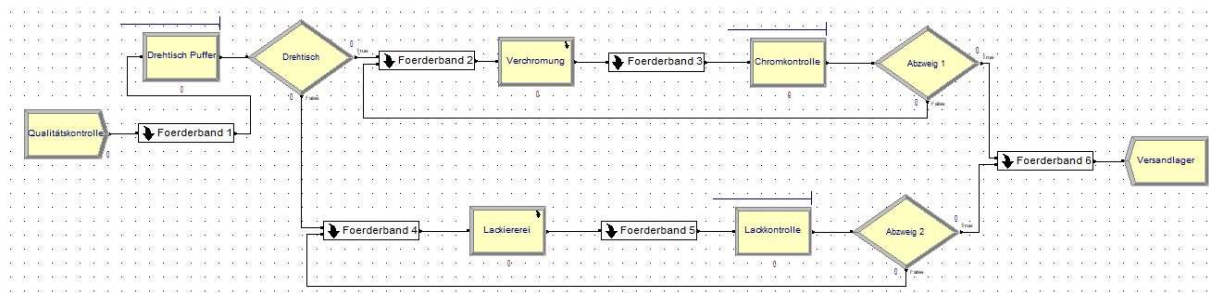


Abbildung 29: Fertigungslinie Heckspoiler in Arena

Auffällig ist hier besonders die bereits erwähnte Verwendung der Submodels. Diese schafft im Vergleich zu der Modellierung aus Kapitel 6.3 eine wesentlich höhere Übersicht, da mehrere Elemente in der darunter liegenden Ebene zu einem Baustein zusammengefasst werden. Dabei sind zwei Varianten von Submodels möglich. Zum einen gibt es den Submodel-Baustein (siehe Abbildung 30), welcher hier für die Darstellung der Förderbänder genutzt wurde und zum anderen kann ein Prozess-Baustein (siehe Abbildung 31) als Submodel genutzt werden. Letztere Möglichkeit wurde bei der Modellierung der Stationen für das Verchromen (Verchromung) und Lackieren (Lackiererei) verwendet und wird durch einen Pfeil in der rechten oberen Ecke des Bausteins angezeigt. Wie diese Submodels erschaffen werden soll an späterer Stelle erläutert werden.



Abbildung 30: Submodel-Baustein



Abbildung 31: Prozess-Baustein als Submodel

Den Anfang stellt eine Quelle (*Create*) dar. Diese schickt Entitäten vom Typ Spoiler\_roh in das System. Über das erste Förderband gelangen diese dann zu einem Drehtisch. Da das Förderband im Aufbau identisch zu den Förderern aus dem ersten Simulationsbeispiel (Kapitel 6) ist, soll hier noch nicht näher auf die Verwendung der Submodels eingegangen werden. Der Drehtisch besteht aus zwei Bausteinen. Zum einen aus einem Prozess (*Process*) und zum anderen aus einem Verteiler (*Decide*). Der Prozess stellt dabei einen Puffer dar und verzögert eine Entität um zwei Sekunden um die Dauer der Drehung des Drehtisches nach zu ahmen. Der Verteiler verteilt anschließend die Entitäten auf die Förderbänder, welche mit der Verchromung und der Lackiererei verbunden sind. Dabei verlassen die Entitäten den Drehtisch mit einer Chance von 10% in Richtung der Verchromung. Die restlichen Entitäten wandern über Förderband 4 zur Lackiererei. Im weiteren Verlauf wird allein auf die Modellierung des Materialflusses über die Verchromung eingegangen, da die Strecke über die Lackiererei im Aufbau sehr ähnlich ist und sich nur die Parameter an der Bearbeitungs-, beziehungsweise Kontrollstation unterscheiden. Abbildung 32 zeigt beispielhaft den Aufbau des Submodels „Förderband 2“, welcher identisch zu Förderband 4 ist.

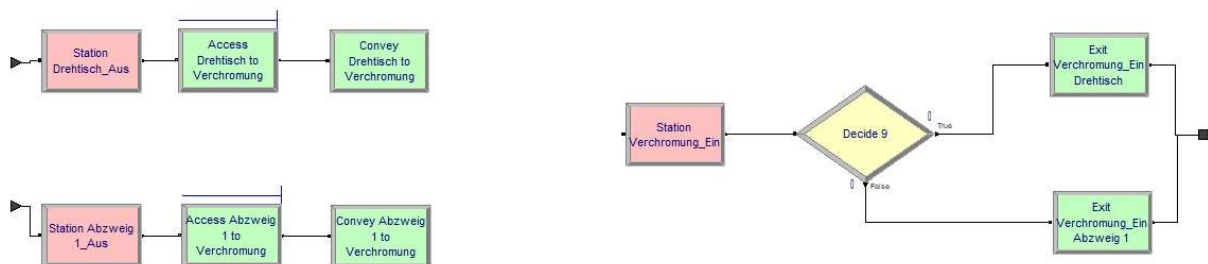


Abbildung 9: Submodel "Foerderband 2"

Platziert man den Submodel-Baustein, so kann man die Anzahl der Ein- und Ausgänge selbst wählen. In diesem Beispiel wurden zwei Eingänge und ein Ausgang gewählt. Der Aufbau der Transportstrecke ist dabei identisch zu den Strecken aus dem vorigen Kapitel. Hier bringen allerdings zwei unterschiedliche Förderer (*Conveyor*) von verschiedenen Förderstrecken die Entitäten zu einer Station (*Station Verchromung\_Ein*). Dies stellt die Zusammenführung zweier

Transportstrecken dar. Hier ist es wichtig zu beachten den zuvor belegten Platz einer Entität auf den unterschiedlichen Transportstrecken wieder frei zu geben. Dazu wird entschieden ob es sich um die Entität Spoiler\_roh handelt. Ist dies der Fall, so verlässt diese den Verteiler-Baustein (*Decide 9*) über den Exit-Baustein (*Exit Verchromung\_ Ein Drehtisch*), welcher den Platz auf dem Förderband aus Richtung des Drehtisches freigibt. Anschließend verlassen die Entitäten das Submodel über den selben Ausgang. Wird nicht der richtige Platz frei gegeben, so ist die Transportstrecke, sobald diese ihre maximale Kapazität erreicht hat, belegt und die Entitäten stauen sich bei dem Access-Baustein. In der Studentenversion ist so die maximale Anzahl an Entitäten in einem Projekt, welche 150 beträgt, nach einiger Zeit erreicht und die Simulation wird automatisch abgebrochen.

Die Verchromung ist ein Prozess vom Typ Submodel, da diese sich aus mehreren Bausteinen zusammensetzt. Wie auf Abbildung 33 zu sehen, gelangen Entitäten vom Eingang zu einem Prozess Baustein, welcher eine Aktion vom Typ Aufnahme (*Seize*), Verzögerung (*Delay*), Abgabe (*Release*) ausführt und somit eine Warteschlange bereitstellt und immer eine Entität nach der Anderen verzögert. Um anschließend fehlerhafte Teile zu simulieren wurde ein Verteiler-Baustein verwendet, welcher 90% der Entitäten auf einen und den Rest auf den anderen Ausgang verteilt des Bausteins verteilt. Anschließend folgen jeweils für jede Richtung ein Zuordnungs-Baustein (*Assign*), welcher die Entitäten zu Spoiler\_verchromt, beziehungsweise Spoiler\_chromfehler umbenennt. Anschließend verlassen die Entitäten wieder den Submodel-Baustein über den gleichen Ausgang.

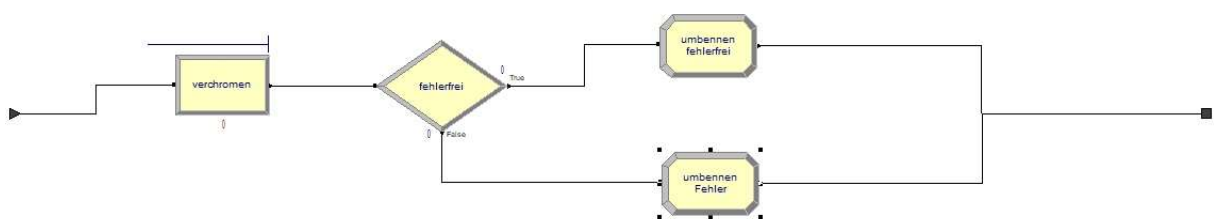


Abbildung 10: Prozess-Submodel "Verchromung"

Nach der Verchromung gelangen die Entitäten über ein weiteres Förderband zur Chromkontrolle, welche als Prozess-Baustein erneut mit Aufnahme, Verzögerung und Abgabe eine Warteschlange bereitstellt und die Entitäten nacheinander verzögert. Darauf folgt der Auswerfer, welcher in dieser Modellierung „Abzweig 1“ heißt und aus einem Verteiler-Baustein besteht. Dieser entscheidet nach dem Typ der Entität (*2-way by Condition*) ob diese den Baustein Richtung Versandlager (Spoiler\_verchromt) verlassen oder erneut der



Verchromung (Spoiler\_chromfehler) zugeführt werden. Das Förderband in Richtung Versandlager (Foerderband 6) ist im Aufbau identisch mit Förderband 2 und wird daher nicht näher erläutert. Das Versandlager ist ein Baustein vom Typ Senke (*Dispose*) und entfernt die Entität wieder aus dem System.

#### 7.4 Vergleich der Simulationstools im Hinblick auf Simulationsbeispiel 2

Werden die Abbildungen des zweiten Beispiels gegenübergestellt, so sollte zu Beginn erwähnt werden, dass dieses Beispiel bereits die Grenzen von GD.findi aufzeigt. Denn während das Modell mit Arena problemlos erstellt werden konnte, mussten bei der Modellierung mit GD.findi zusätzliche Annahmen zur Vereinfachung getroffen werden. So war es mit diesem Simulator nicht möglich eine Rückführung der Entitäten zu realisieren. Dies hat zwei Gründe, denn zum einen konnte kein Attribut, wie zum Beispiel der Typ, einer Entität verändert werden und zum anderen gibt es keine Verzweigung, welche einen Entitäten-Fluss prozentual auf zwei Richtungen aufteilt. Das bedeutet, dass bei einer Verzweigung für alle möglichen Richtungen die Entitätstypen bereits von Beginn an im System sein müssen. Dies bedeutet einen erheblichen Arbeitsaufwand, da für jede Entität ein eigener Prozessplan erstellt werden muss. In Arena gibt es hierfür die Bausteine *Assign* und *Decide*.

Der Drehtisch konnte mit beiden Tools abgebildet werden. In GD.findi definiert man dazu eine Station mit einem Eingang und mehreren Ausgängen. Die für die Drehung benötigte Zeit wird durch eine Bearbeitungszeit an dieser Station realisiert. In Arena geschieht dies ähnlich durch die Verwendung der Bausteine *Process* und *Decide*. Der Auswerfer kann analog modelliert werden. Sollte die durch den Auswerfer verursachte Verzögerung vernachlässigbar klein sein, so entfällt bei GD.findi die Bearbeitungszeit und bei Arena der *Process*-Baustein.

Durch die Verwendung von Submodels wurde zudem bei der Modellbildung mit Arena eine hohe Übersichtlichkeit geschaffen, sodass das Modell fast nahezu einem Flussdiagramm, beziehungsweise dem Prozessplan in GD.findi entspricht.

## 8 Gegenüberstellung von GD.findi und Arena

Nachdem GD.findi und Arena an zwei ausgewählten Beispielen getestet wurden, sollen die beiden Simulationstools in diesem Kapitel nach ausgewählten Kriterien bewertet und gegenübergestellt werden. Dazu erfolgt zuerst eine Festlegung der Kriterien und anschließend die Bewertung der beiden Tools mit darauffolgender Gegenüberstellung im Hinblick auf jedes Kriterium.

### 8.1 Festlegung der Kriterien zur Bewertung

Für die Bewertung und Gegenüberstellung der beiden Simulationstools wurden Kriterien aus verschiedenen Bereichen herangezogen um die Stärken und Schwächen der Programme herausarbeiten zu können. Die ausgewählten Kriterien richten sich dabei grob nach der VDI-Norm 3633, Blatt 4 ([VDI97.4]). So sollen zuerst die Anwendungsschwerpunkte, sowie besonderen Charakteristika der Tools gegenüber gestellt werden um einen ersten Eindruck über die Anwendungsbereiche beider Simulatoren zu bekommen. Anschließend werden die technischen Anforderungen, sowie die Qualifikationsanforderungen an den Benutzer dargestellt. Dies soll helfen die Anwendbarkeit im Rahmen der Anforderungen einzuordnen. Um die Funktionsweisen beider Werkzeuge vergleichen zu können, sollen danach die grafischen Oberflächen, die vorhandenen Modellelemente, die Animation während der Simulation und die Ergebnisauswertung aufgezeigt werden. Zuletzt sollen, um entscheiden zu können ob die Benutzung eines oder beider Tools langfristig sinnvoll ist, die Anwenderunterstützung und Softwarepflege durch den Herausgeber, sowie die Anwendbarkeit in der Lehre und auf das MINIS-Projekt vergleichend dargestellt werden. Zuletzt sollen die Zukunftsaussichten der Werkzeuge beurteilt werden.

## 8.2 Bewertung

### - *Anwendungsschwerpunkte*

Der Anwendungsschwerpunkt von GD.findi liegt in der Simulation von Produktionssystemen und der Planung von neuen Fabriken. Aus diesem Grund gehört es zu den anwendungsbezogenen Simulationswerkzeugen. Arena bietet durch die Art der Modellierung, welche auf SIMAN basiert ein weites Spektrum an Anwendungsmöglichkeit. Der Schwerpunkt liegt hier allerdings auf der Modellierung logistischer Prozesse, da hierfür bereits vorgefertigte Bausteine in der Projektleiste zu Verfügung stehen. Im Hinblick auf diesen Punkt stellt somit GD.findi einen näheren Anwendungsschwerpunkt zu den betrachteten Beispielen dar, stößt allerdings im Gegensatz zu Arena schnell an die Grenzen seiner Möglichkeiten.

### - *Qualifikationsanforderungen*

Während bei der Modellierung und Simulation mit GD.findi keine Programmierkenntnisse und nennenswerten Simulationserfahrungen nötig sind, muss der Anwender beim Umgang mit Arena eine Vorstellung von der Funktionsweise von SIMAN besitzen um den Aufbau logischer Konstrukte mit den vorhandenen Bausteinen realisieren zu können. Mit geeigneter Lektüre wäre eine Einarbeitung in GD.findi somit schneller durchzuführen als mit Arena.

### - *Schnittstelle zum Benutzer (GUI)*

Die Bedienoberfläche von GD.findi ist sehr simpel gehalten, was die Benutzung für Nutzer sehr bedienerfreundlich macht. Die wichtigsten Informationen sind schnell zu erlernen und nehmen neuen Nutzern die „Angst“, mit einem Simulationswerkzeug zu arbeiten. Bei Arena dagegen ist die Benutzeroberfläche zwar übersichtlich gestaltet und bietet viele Konfigurationsmöglichkeiten (vergleiche Kapitel 5), neue Nutzer müssen sich allerdings vor dem ersten Modellieren mit dem Programm vertraut machen.

- *Modellelemente*

Die vorgefertigten Elemente in GD.findi sind sehr begrenzt. Es lassen sich lediglich Stationen (egal ob Arbeitsstationen, Puffer, Quellen oder Senken) auswählen und verbinden. Soll ein Fließband, etc. abgebildet werden, ist dies nicht ohne weiteres möglich, da es in GD.findi keine vorgefertigten Bausteine gibt. Dies ist ein erheblicher Nachteil des Werkzeuges. Denn Arena bietet dagegen vorgefertigte Bausteine in einer Bibliothek (*Project Bar*) an, bei welchen bestimmte Parameter definiert werden können. Des Weiteren besteht die Möglichkeit direkt über die Programmiersprache SIMAN zu programmieren.

- *Animation während der Simulation*

Wird eine Animation mit GD.findi gestartet, so werden die Ereignisse im System durch eine Darstellung der Entitäten gegeben, welche sich gerade im System befinden. Da diese Simulation nicht angehalten werden kann ist eine Optimierung des Systems nur mit erhöhtem Aufwand durch die Aufbereitung der Ergebnisse möglich. Modelliert man dagegen in Arena mithilfe der bereitgestellten Bausteine, so werden die Entitäten angezeigt, während diese die Simulation durchlaufen. Hierbei lassen sich den Entitäten verschiedene Bilder zuordnen, sodass eine einfache grafische Darstellung gegeben ist. Arena bietet zudem Tools und Bibliotheken zum 2D-Darstellen an, auf welche allerdings in den Beispielen verzichtet wurde. Des Weiteren kann laut Hersteller mit zusätzlicher Software eine 3-dimensionale Animation realisiert werden.

- *Ergebnisdarstellung und Ergebnisauswertung*

Die Ergebnisse lassen sich in GD.findi in drei Umgebungen darstellen (vergleiche Kapitel 4). Des Weiteren können diese in der Pro-Version als Excel-Datei ausgegeben werden. In der verwendeten Studentenversion ist dies nicht möglich. Arena hingegen bietet detaillierte Auswertungen an. Unter dem Punkt *Reports* in der Projektleiste können Berichte zu einzelnen Elementen, wie zum Beispiel den Entitäten, den Warteschlangen oder den Prozessen herangezogen werden. Diese werden dann auf einem DIN A4 Bogen angezeigt und können sowohl gedruckt, als auch exportiert werden.

- *Anwenderunterstützung und Softwarepflege*

Betrachtet man zuerst die Anwenderunterstützung und Softwarepflege so lässt sich zum GD.findi feststellen, dass die Möglichkeit sich über das Programm zu informieren extrem begrenzt ist. Dies liegt unter anderem daran, dass das Programm vorwiegend auf dem asiatischen Markt präsent ist. Im Rahmen dieser Projektarbeit stellte sich zudem raus, dass die zu Verfügung stehenden Lernmittel auf englischer Sprache extrem begrenzt sind. Zwar wurde der Kontakt zum Vertreter des deutschen Marktes hergestellt, dieser musste allerdings bei den meisten Fragen Rücksprache mit den japanischen Mitarbeitern halten. Dies hatte einen sehr hohen Zeitaufwand zur Folge. Des Weiteren ist das letzte Update der kostenlosen Version im Dezember 2017 erschienen und die, auf der Webseite zu findende, Bedienungsanleitung aus dem Jahr 2014. Bei Arena dagegen wird nach der Registrierung, neben Link zu der Installationsdatei, ein Link zu einem Einführungsvideo per E-Mail zugestellt. Dieses Video gibt einen kurzen Überblick über die Modellierung mit Arena und erleichtert dadurch den Einstieg erheblich. Des Weiteren verfügt Arena, beziehungsweise Rockwell Automation über einen kostenfreien technischen Support, welcher per E-Mail kontaktiert werden kann und aus unseren Erfahrungen schnell und präzise antwortet. Zudem existieren einige Publikationen zur Modellierung und Simulation mit Arena. Ein Beispiel wäre „Simulation Modeling and Analysis with ARENA“ ([AM07]). Im Bereich Softwarepflege bietet Arena regelmäßige Updates an, so existiert bereits die Version 15.10.00001.

- *Anwendbarkeit in der Lehre/ im MINIS-Projekt*

Betrachtet man die Anwendbarkeit von GD.findi in der Lehre, so ist diese zu bezweifeln. Denn zum einen gibt die Studentenversion eine Begrenzung auf drei Projekte vorgibt und somit nicht viel getestet werden kann. Außerdem ist es nicht gestattet, die Ergebnisse abzuspeichern. Das würde einen Vergleich mehrerer Projekte über ein oder mehrere im Übungsbetrieb beinahe unmöglich machen. Arena bietet dagegen die Möglichkeit beliebig viele Projekte zu erstellen. Im Rahmen der Projektarbeit wurde als bisher einzige Einschränkung die Begrenzung von maximal 150, sich gleichzeitig im System befindenden Entitäten festgestellt. Diese Begrenzung sollte aber im Übungsbetrieb keine Probleme darstellen. Des Weiteren hilft das Tool durch die Nähe zur Programmiersprache SIMAN die Logik hinter einem Simulationsprogramm nachvollziehen zu können. Im Hinblick auf das

MINIS-Projekt ist GD.findi eher unbrauchbar, da einige Arbeitsschritte nicht, oder nur unter Vereinfachungen, dargestellt werden können. Mit Arena können alle Elemente abgebildet werden, auch wenn die einzelnen Module bei der Modellierung nur als Gesamtstrecke berücksichtigt werden.

- *Zukunftsansichten*

Abschließend lässt sich somit im Bezug auf die Zukunftsansichten von GD.findi sagen, dass diese nur schwer abzuschätzen sind, da die Ansätze zwar gut sind, das Werkzeug aber schnell an seine Grenzen stößt. Zudem fehlt das Angebot an Möglichkeiten zum Selbststudium, was neben der begrenzten Anzahl an Projekten in der Studentenversion, das Programm für die Benutzung in der Lehre unbrauchbar macht. Die Ansichten für Arena hingegen sind als sehr gut zu bewerten, da das Programm sehr mächtig ist und viele Anwendungsbereiche abdeckt. Es bietet sich außerdem für die Lehre an, da es hilft das logische Konstrukt, welches hinter einer Modellierung und Simulation steckt nachzuvollziehen. Zudem ist die Vielfalt an zu Verfügung stehender Lektüre sehr groß ist.

## 9 Zusammenfassung

Unter dem Aspekt, dass die Wichtigkeit von selbststeuernden Logistik- und Produktionssystemen im digitalen Zeitalter – wie dem MINIS-Projekt der TU Dortmund – immer mehr an Bedeutung dazugewinnen und dazu Optimierungsvorgänge während des laufenden Betriebs durchgeführt werden müssen, lässt sich aufgrund der Erkenntnisse aus dieser Projektarbeit, in welcher die Simulationswerkzeuge GD.findi und Arena Simulation Software untersucht wurden, ein Fazit zur Anwendbarkeit der Tools auf die Modellierung und Simulation von miniaturisierten selbststeuernden Logistiksystemen ziehen. Im Hinblick darauf wurden diese an zwei Anwendungsbeispielen getestet und im Anschluss miteinander verglichen. Grundlegend können beide Programme für die Modellierung von Logistik- und Produktionssystemen verwendet werden, allerdings stößt GD.findi schneller an die Grenzen seiner Möglichkeiten. Wird vergleichend dazu das Funktionsspektrum von Arena betrachtet, so ist dieses erheblich größer. Dies hat jedoch zur Folge, dass viel mehr Zeit benötigt wird, bis die Funktionen des Tools beherrscht werden. Im Gegensatz dazu ist die Modellierung mit GD.findi unkomplizierter, da für seine Nutzung keine Programmierkenntnisse und Simulationserfahrungen benötigt werden. Im Vergleich zu Arena besitzt GD.findi keine Bibliothek mit vorkonfigurierten Bausteinen, weshalb sich die Fördererelemente des MINIS-Projekts nur bedingt darstellen lassen. Arena bietet zwar keine modularen Fördermittel als Bausteine an, diese lassen sich jedoch wie in den Beispielen aus Kapitel 6 und 7 auf verschiedenste Weisen nachahmen. Insgesamt lässt sich das sagen, dass Arena für die Anwendung im MINIS-Projekt zwar in der Modellierung aufwendiger, jedoch durch das breitere Funktionsspektrum deutlich besser geeignet ist. GD.findi könnte insofern für ein Projekt wie dem MINIS-Projekt verwendet werden, indem mit diesem Tool ein vereinfachtes Modell erstellt wird um erste Einschätzungen zur Machbarkeit des Projekts treffen zu können. Für eine komplexere Modellierung müsste dann jedoch ein mächtigeres Tool, wie zum Beispiel Arena herangezogen werden.

## 10 Abbildungsverzeichnis/ Tabellenverzeichnis

Abbildung 1: Vorgehensmodell bei der Simulation mit V&V .....	11
Abbildung 2: Schematischer Verlauf der Kosten von der Planung bis zur Inbetriebnahme .....	13
Abbildung 3: Klassifizierung von Simulationswerkzeugen .....	14
Abbildung 4: Grafische Benutzeroberfläche GD.findi .....	16
Abbildung 5: Verbindung von Stationen durch Anschlüsse, Knoten und Pfeile .....	17
Abbildung 6: Beispiel Prozessplan.....	18
Abbildung 7: Production Cockpit.....	19
Abbildung 8: GUI Arena Simulation Software .....	22
Abbildung 9: Beispiel block .....	23
Abbildung 10: Beispiel element .....	23
Abbildung 11: Tabelle für Baustein vom Typ Create.....	24
Abbildung 12: Beispiel eines Modells in Arena .....	27
Abbildung 13: Bearbeitungsfenster für „Process 1“ .....	28
Abbildung 14: Darstellung des Laufrades „Wishbone Bike 2in1 – original“ .....	29
Abbildung 15: Modifizierte Prozessdarstellung inklusive Input/ Output nach [BN16] .....	30
Abbildung 16: Beispielhaftes Layout der Produktionsanlage .....	31
Abbildung 17: Raumplan für Anwendungsbeispiel Laufradfertigung .....	32
Abbildung 18: Prozessplan für Laufradfertigung mit Beschreibung eines Prozesses .....	33
Abbildung 19: Agent Designer.....	34
Abbildung 20: Verknüpfung der Anschlüsse mit Punkten zu Routen .....	34
Abbildung 21: Modell der Laufradfertigung in Arena .....	35
Abbildung 22: Modell einer Förderstrecke .....	36
Abbildung 23: Bearbeitungsfenster Coveyor.Segment.....	36
Abbildung 24: Fertigungslinie Heckspoiler.....	39
Abbildung 25: Raumplan für Anwendungsbeispiel Heckspoilerfertigung .....	41



Abbildung 26: Prozessplan für Anwendungsbeispiel Heckspoilerfertigung .....	42
Abbildung 27: Produktionsplanung der Heckspoilerfertigung.....	43
Abbildung 28: Prozessbeschreibung eines Drehtisches.....	43
Abbildung 29: Fertigungslinie Heckspoiler in Arena .....	44
Abbildung 30: Submodel-Baustein.....	45
Abbildung 31: Prozess-Baustein als Submodel .....	45
Abbildung 32: Submodel „Foerderband 2“ .....	45
Abbildung 33: Prozess Submodel „Verchromung“ .....	46
Tabelle 1: Module MINIS-Projekt.....	5
Tabelle 2: Verwendete Bausteine Arena.....	25
Tabelle 3: Bearbeitungszeiten für das Anwendungsbeispiel der Laufradfertigung .....	31
Tabelle 4: Bearbeitungszeiten und Verteilungen der Arbeitsstation bei der Heckspoilerfertigung .....	40
Tabelle 5: Längen der Transportförderer.....	41

## 11 Literaturverzeichnis

- [AM07] Altiook, T.; Melamed, B.: Simulation Modeling and Analysis with ARENA. Academic Press, 2007.
- [ASS18] ARENA SIMULATION SOFTWARE: Download Trial Version.  
<https://www.arenasimulation.com/simulation-software-download>, 18.12.2017.
- [Ban98] Banks, J.: Handbook of Simulation. Engineering & Management Press, 1998.
- [Bmbf] Bundesministerium für Bildung und Forschung: bmbf.de.  
<https://www.bmbf.de/de/zukunftsprojekt-industrie-4-0-848.html>, 12.01.2018.
- [BN16] Baller, N.; Nerke, C.: Untersuchung von Automod und Plant Simulation zur Abbildung und Analyse von Logistiksystemen. Projektarbeit, Dortmund, 2016.
- [Bou17] Bousonville, T.: Logistik 4.0. Die digitale Transformation der Wertschöpfungskette. Springer, Wiesbaden, 2017.
- [Ele12] Eley, M.: Simulation in der Logistik – Eine Einführung in die Erstellung ereignisdiskreter Modelle unter der Verwendung des Werkzeuges „Plant Simulation“. Springer-Verlag Berlin Heidelberg, Heidelberg, 2012.
- [GRS<sup>+</sup>17] Gutenschwager, K. et al.: Simulation in Produktion und Logistik – Grundlagen und Anwendungen. Springer-Verlag GmbH Deutschland, Berlin, 2017.
- [Gud10] Gudehus, T.: Logistik. Grundlagen – Strategien – Anwendungen. Springer, Berlin, 2010.
- [Law07] Law, A.M.: Simulation modeling and analysis. McGraw-Hill, Boston, 2007.
- [Lex1] LEXER RESEARCH Inc.: Chronology.  
<http://www.lexer.co.jp/en/company/history>, 03.01.2018
- [Lex2] LEXER RESEARCH Inc.: Vision.  
<http://www.lexer.co.jp/en/about/visio>, 03.01.2018
- [Lex3] [Lex3] LEXER RESEARCH Inc.: GD.findi Plans and Pricing.  
<https://www.gdfindi.com/en/entry/license.html>, 01.02.2018
- [Lex4] LEXER RESEARCH Inc.: Operations Manual.  
[https://www.gdfindi.com/en/support/help/pdf/Operation\\_manual.pdf](https://www.gdfindi.com/en/support/help/pdf/Operation_manual.pdf), 01.02.2018
- [Lex5] LEXER RESEARCH Inc.: GD.findi.  
[http://www.lexer.co.jp/static/pdf/lexer\\_GD.findi.pdf](http://www.lexer.co.jp/static/pdf/lexer_GD.findi.pdf), 03.01.2018
- [Lex6] LEXER RESEARCH Inc.: Diagramm auf Startbildschirm nach Anmeldung.  
<https://www.gdfindi.com/>, 01.02.2018
- [Lex7] LEXER RESEARCH Inc.: GD.findi modeling message.  
[https://www.gdfindi.com/en/support/help/err\\_msg\\_modeling.html](https://www.gdfindi.com/en/support/help/err_msg_modeling.html), 01.02.2018

- [Lex8] LEXER RESEARCH Inc.: Function Manual.  
[https://www.gdfindi.com/en/support/help/pdf/Function\\_manual.pdf](https://www.gdfindi.com/en/support/help/pdf/Function_manual.pdf), 01.02.2018
- [MKR<sup>+</sup>11] März, L. et al.: Simulation und Optimierung in Produktion und Logistik. Praxisorientierter Leitfaden mit Fallbeispielen. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [RA18] ROCKWELL AUTOMATION, Inc.: Our History.  
<https://www.rockwellautomation.com/global/about-us/overview.page?pagetitle=Our-History&docid=a162d41cd4310beab22a277ea3d4e2ac>, 18.12.2017
- [RSW08] Rabe, M.; Spieckermann, S.; Wenzel, S.: Verifikation und Validierung für die Simulation in Produktion und Logistik – Vorgehensmodelle und Techniken. Springer-Verlag Berlin Heidelberg, Berlin, 2008.
- [Sch15] Schenk, M.: Produktion und Logistik mit Zukunft. Springer, Berlin, 2015.
- [VDI14] VDI Verein Deutscher Ingenieure: VDI 3633 Blatt 1 – Simulation von Logistik-, Materialfluss- und Produktionssystemen – Grundlagen. Beuth Verlag GmbH, Berlin, 2014.
- [VDI18] VDI Verein Deutscher Ingenieure: VDI 3633 – Simulation von Logistik-, Materialfluss- und Produktionssystemen – Begriffe. Beuth Verlag GmbH, Berlin, 2018.
- [VDI97.3] VDI Verein Deutscher Ingenieure: VDI 3633 Blatt 3 – Simulation von Logistik-, Materialfluss- und Produktionssystemen – Experimentplanung und -auswertung. Beuth Verlag GmbH, Berlin, 1997.
- [VDI97.4] VDI-Gesellschaft Fördertechnik Materialfluß Logistik, Fachbereich Simulation: VDI 3633 Blatt 4. Auswahl von Simulationswerkzeugen. VDI, Düsseldorf, 1997.
- [WIS1] Wishbone Design Studio: Webshop wishbone bike 2in1 – original.  
<https://www.shopwishbonedesign.com/products/Details.aspx?p=854&c=83&g=all>, 18.01.2018
- [WWC08] Wenzel, S. et al.: Qualitätskriterien für die Simulation in Produktion und Logistik – Planung und Durchführung von Simulationsstudien. Springer-Verlag Berlin Heidelberg, 2008.
- [Wyc09] Wycisk, C.: Flexibilität durch Selbststeuerung in logistischen Systemen. Entwicklung eines realoptionsbasierten Bewertungsmodells. Gabler, Wiesbaden, 2009.

## 12 Anhang

A – Unterbereich „Werkzeuge/ Standard“



B – Unterbereich „Werkzeuge/ View“



C – Bearbeitungsfenster Access-Baustein

The 'Access' dialog box has a green title bar with a question mark and a close button. It contains the following fields:

- Name: Access 3 (dropdown)
- Conveyor Name: Foerderer 3 (dropdown)
- # of Cells: 1 (text input)
- Queue Type: Queue (dropdown)
- Queue Name: Access 3.Queue (dropdown)

Buttons: OK, Cancel, Help

D – Bearbeitungsfenster Convey-Baustein

The 'Convey' dialog box has a green title bar with a question mark and a close button. It contains the following fields:

- Name: Foerderstrecke 3 (text input)
- Conveyor Name: Foerderer 3 (dropdown)
- Destination Type: Station (dropdown)
- Station Name: Station 6 (dropdown)

Buttons: OK, Cancel, Help

## E – Bearbeitungsfenster Exit-Baustein

## Aufteilung der Kapitel auf die Autoren

Thema	Autor
Kapitel 1: Einleitung	Patzelt
Kapitel 2: Selbststeuernde Logistiksysteme	Patzelt
Kapitel 3: Simulation	Fahrenholz
Kapitel 4: GD.findi	Fahrenholz
Kapitel 5: Arena Simulation Software	Patzelt
Kapitel 6: Laufradfertigung	
Kapitel 6.1: Erklärung	Fahrenholz
Kapitel 6.2: Abbildung in GD.findi	Fahrenholz
Kapitel 6.3: Abbildung in Arena	Patzelt
Kapitel 6.4: Vergleich der Ergebnisse beider Simulationstools	Fahrenholz, Patzelt
Kapitel 7: Beispiel 2	
Kapitel 7.1: Erklärung	Patzelt
Kapitel 7.2: Abbildung in GD.findi	Fahrenholz
Kapitel 7.3: Abbildung in Arena	Patzelt
Kapitel 7.4: Vergleich der Ergebnisse beider Simulationstools	Fahrenholz, Patzelt
Kapitel 8: Gegenüberstellung von GD.findi und Arena	Fahrenholz, Patzelt
Kapitel 9: Zusammenfassung	Fahrenholz, Patzelt

## Eidesstattliche Versicherung

Wir versichern hiermit Eides statt, dass wir die vorliegende fachwissenschaftliche Projektarbeit mit dem Titel „Modellierung und Simulation selbststeuernder miniaturisierter Logistiksysteme mit GD.findi und Arena“ selbstständig und ohne unzulässige fremde Hilfe erbracht haben. Wir haben keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht.

---

Name, Vorname

---

Matrikelnr.

---

Ort, Datum

---

Unterschrift

---

Name, Vorname

---

Matrikelnr.

---

Ort, Datum

---

Unterschrift