

Diplomarbeit

Entwicklung eines Analysetools für Intralogistiksysteme mit autonomen Transportrobotern

von

David Forster

Technische Universität Graz

Fakultät Maschinenbau und Wirtschaftswissenschaften

Institut für Technische Logistik

Betreuer: Dipl.-Ing. Wolfgang Trummer
Beurteiler: Ass.Prof. Dipl.-Ing. Dr.techn. Norbert Hafner

Graz, Mai 2019

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Graz, am

.....
(Unterschrift)

Danksagung

An der Realisierung der vorliegenden Arbeit waren viele Personen beteiligt, denen ich an dieser Stelle meinen herzlichen Dank aussprechen möchte. Allen voran danke ich Michael Mätzler und den Mitarbeitern von Servus Intralogistics für die Möglichkeit, diese Arbeit durchzuführen sowie Dipl.-Ing. Wolfgang Trummer für die universitäre Betreuung.

Ebenso möchte ich meiner Freundin Lisa, meiner Familie, meinen Freunden und Studienkollegen für die Unterstützung während der Studienzeit und für die schöne Zeit in Graz danken.

Kurzfassung

Diese Diplomarbeit beschäftigt sich mit der Erstellung eines Analysetools für Intralogistiksysteme mit autonomen Transportrobotern und wurde in Kooperation mit der Firma Servus Intralogistics erstellt. Das Ziel der Arbeit ist es, die Analyse und Darstellung der Anlagenleistung, der Leistung der einzelnen Komponenten, der Anlagenverfügbarkeit und der Betriebsstörungen zu ermöglichen.

Um die Ausgangssituation besser zu verstehen, werden im ersten Schritt das Servus-System mit all seinen Komponenten und Diensten und der Ablauf des dazugehörigen Logistikprozesses vorgestellt. Bei der Abarbeitung der Fahraufträge durch die Transportroboter fällt eine Vielzahl an Daten an, die aber nicht direkt für die Auswertung der Anlage herangezogen werden kann. Deshalb werden im nächsten Schritt die Methoden zur Datenaufbereitung und Analyse erklärt, die notwendig sind, um aus den vorliegenden Rohdaten aussagekräftige Kennzahlen zu generieren.

Anschließend wird kurz auf den Entwurf der Software in C# eingegangen. Diese Software liest die Daten in regelmäßigen Abständen von einem Analyseserver ein, verarbeitet sie mit den ausgearbeiteten Methoden und speichert sie wieder auf dem Analyseserver ab, wo sie dann mit dem Visualisierungstool Kibana zu verschiedenen Dashboards zusammengefasst werden.

Abschließend werden die Ergebnisse anhand von Auswertungsbeispielen realer Anlagen präsentiert.

Abstract

This thesis addresses the implementation of an analysis tool for intralogistical systems by means of autonomous transport robots. It has been created in cooperation with Servus Intralogistics with the objective to determine and visualize the performance of the entire logistics system and different components associated as well as the malfunctions and availability of the system at that time.

In the first part, the Servus system with all its components and services are explained in detail. Whilst operating, the system creates a lot of data which cannot be used for direct analysis. In order to obtain the characteristic values needed for the evaluation of a logistics facility the raw data must be pre-processed and analysed with methods which are explained in the modelling chapter.

In the next part the implementation of the software created in C# is illustrated. This software frequently reads the raw data from an analysis server, processes them with the developed methods and writes them back to the database where the characteristic values are visualized with dashboards in a tool called Kibana. In the last chapter the results are presented with evaluations of real-world logistic facilities.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Ausgangssituation	1
1.2	Zielsetzung	1
2	Problemanalyse.....	2
2.1	Servus-System	2
2.1.1	Bestandteile des Servus-Systems	3
2.1.1.1	Fahrstrecke	4
2.1.1.2	Transportroboter (ARC).....	4
2.1.1.3	RFID-Knoten.....	5
2.1.1.4	Weiche/Einmündung.....	6
2.1.1.5	Heber	7
2.1.1.6	Paternoster.....	7
2.1.1.7	Arbeitsplätze	8
2.1.1.8	Pufferplätze.....	9
2.2	Datenbank.....	9
2.2.1	Elasticsearch.....	10
3	Modellbildung.....	12
3.1	Kennzahlen	12
3.1.1	Materialfluss.....	13
3.1.2	Komponenten.....	14
3.1.3	Verfügbarkeit.....	15
3.2	Ablauf der Modellbildung	16
3.3	Ausgangsdaten.....	17
3.3.1	Orderboard.....	17
3.3.2	Log-Files	18
3.3.3	Meldesystem	18
3.3.4	Pfadmanagement.....	18
3.4	Datenaufbereitung.....	19
3.4.1	Assistenten	19
3.4.2	ARC-Position	19
3.5	Methoden.....	20
3.5.1	ARC-Aufträge	20
3.5.1.1	Doppelspiele	20
3.5.1.2	Gefahrene Strecke.....	21
3.5.1.2.1	Direkte Ermittlung der gefahrenen Strecke	21
3.5.1.2.2	Indirekte Ermittlung der gefahrenen Strecke	21
3.5.1.2.3	Wahl der Methode	22
3.5.2	Assistenten	22
3.5.2.1	Telegrammtypen	24
3.5.2.2	Grenzdurchsatz und Assistentenauslastung.....	25
3.5.2.2.1	Verzweigungen	26
3.5.2.2.2	Beispiel.....	30
3.5.2.2.3	Auslastungsgrad mit Belegungszeiten	31
3.5.3	ARC-Position	33
3.5.4	Verfügbarkeit.....	33
3.5.4.1	Störungsbasierte Verfügbarkeitsrichtlinien.....	34

3.5.4.1.1	Beispiel für eine störungsbasierte Verfügbarkeitsrechnung	34
3.5.4.2	Leistungsverfügbarkeitsmodell.....	37
3.5.4.2.1	Beispiel für die Berechnung der Leistungsverfügbarkeit.....	38
3.5.4.3	Wahl der Verfügbarkeitsmethode	39
3.5.4.4	Anwendung der störungsbasierten Verfügbarkeitsrechnung....	39
3.5.5	Hilfsmethoden	40
3.5.5.1	Streckenmatrix	40
3.5.5.1.1	Floyd-Warshall Algorithmus.....	42
3.5.5.2	Zeitkorrektur der Log-Files.....	44
3.5.5.2.1	Zeitkorrektur mittels gewichtetem Mittelwert	44
3.5.5.2.2	Zeitkorrektur mittels Mindestwert.....	45
3.5.5.2.3	Vergleich der Methoden	46
3.5.5.3	ARC-Stillstand	46
3.6	Darstellung	47
3.6.1	Kibana.....	54
4	Implementierung	56
4.1	Softwareaufbau	56
4.1.1	Konfigurationsdaten.....	58
4.1.2	Inputdaten	60
4.1.3	Ausgewertete Daten	61
4.2	Verifikation/Validierung.....	63
4.2.1	Verifikation.....	63
4.2.2	Validierung	63
5	Ergebnisse	64
5.1	Fall 1: Doppelspielproblem	64
5.2	Fall 2: Systemupdate	66
5.3	Fall 3: Ermittlung der Leistungsreserve	68
5.4	Fall 4: Verfügbarkeit	70
6	Zusammenfassung und Ausblick.....	73
7	Verzeichnisse.....	74
7.1	Abkürzungsverzeichnis	74
7.2	Abbildungsverzeichnis.....	74
7.3	Tabellenverzeichnis	76
7.4	Literaturverzeichnis	76

1 Einleitung

1.1 Ausgangssituation

Die Firma Servus Intralogistics GmbH stellt Logistiksysteme mit autonomen, schienengebundenen Transportrobotern her. Beim Betrieb dieser Anlagen, die zum Teil aus 80 Robotern und etlichen Assistenten bestehen, fallen große Mengen an Daten an. Bislang lagen diese auf dem Server der Kunden und wurden aufgrund des begrenzten Speicherplatzes nach einigen Tagen gelöscht. Im Sommer 2018 erfolgte daher die Einrichtung eines Dienstes, welcher den Inhalt der Kundendatenbank an einen Analyseserver der Firma Servus übermittelt. Die gesammelten Rohdaten sind nicht sehr aussagekräftig und können auch nur vom Fachpersonal der Softwareentwicklungsabteilung unter hohem Aufwand analysiert werden. Aus diesem Grund entstand das Bedürfnis, ein Auswertungstool zu erstellen, welches auf alle bestehenden und in Zukunft installierten Anlagen angewendet werden kann.

1.2 Zielsetzung

Im Rahmen dieser Diplomarbeit soll ein Analysetool entwickelt werden, das sowohl den eigenen Mitarbeitern, als auch dem Kunden erlaubt sich einen schnellen Überblick über die Anlage zu verschaffen und zudem die Durchführung genauerer Auswertungen ermöglicht.

Ziel dieser Diplomarbeit ist:

- Die tabellarische und graphische Darstellung der Anlagenleistung sowie aller leistungsbeeinflussenden Faktoren (z.B. Bottlenecks, Doppelspiele etc.).
- Die tabellarische und graphische Darstellung der Anlagenverfügbarkeit. Außerdem soll das Tool den Anwender bei der Fehlererkennung unterstützen.

2 Problemanalyse

2.1 Servus-System

Die Firma Servus produziert Intralogistiksysteme, bei denen das Ladegut von einem oder mehreren unabhängigen, schienengebundenen Transportrobotern (ARC: Autonomous Robotic Carrier) befördert wird. Das Be- und Entladen der ARCs kann entweder manuell durch einen Anlagenbediener, vom Transportroboter selbst oder durch stationäre Vorrichtungen (z.B. Roboterarm) erfolgen. Dabei kann das Ladegut form- oder kraftschlüssig gehalten werden.

Die Lagerplatzverwaltung und Optimierung der Transportfahrten erfolgt über das Lagerverwaltungssystem (LVS). Die vom LVS generierten Fahraufträge werden an das Servus Orderboard übermittelt und von dort über WLAN oder SERAN (Servus Area Network; Hochfrequenz-Funktechnologie) an die ARCs weitergeleitet. Das Servus-System wird dezentral geregelt, was bedeutet, dass die ARCs sich und ihre Assistenten (z.B. Heber, Weichen, Ladestationen) selbstständig steuern und eigenständig den schnellsten Weg zum Ziel berechnen.

Das übergeordnete System (SEAS: Servus Efficient Agents System) übernimmt die auftragspezifische Koordination der ARCs untereinander.

Aus Datensicht sieht das Servus-System vereinfacht wie in Abbildung 1 aus. Hier werden die Fahraufträge im LVS erstellt und über SEAS an die ARCs weitergeleitet. Die ARCs kommunizieren über Telegramme mit den Assistenten. Die ARCs, Assistenten und die verschiedenen Dienste innerhalb der SEAS Systems (z.B. Orderboard, ARC-Agent etc.) senden Log-Daten an SEAS, welches die wichtigsten Informationen (z.B. Betriebsstörungen) zusammenfasst und diese über das Meldesystem an den Kunden bzw. an die Servus-Mitarbeiter sendet.

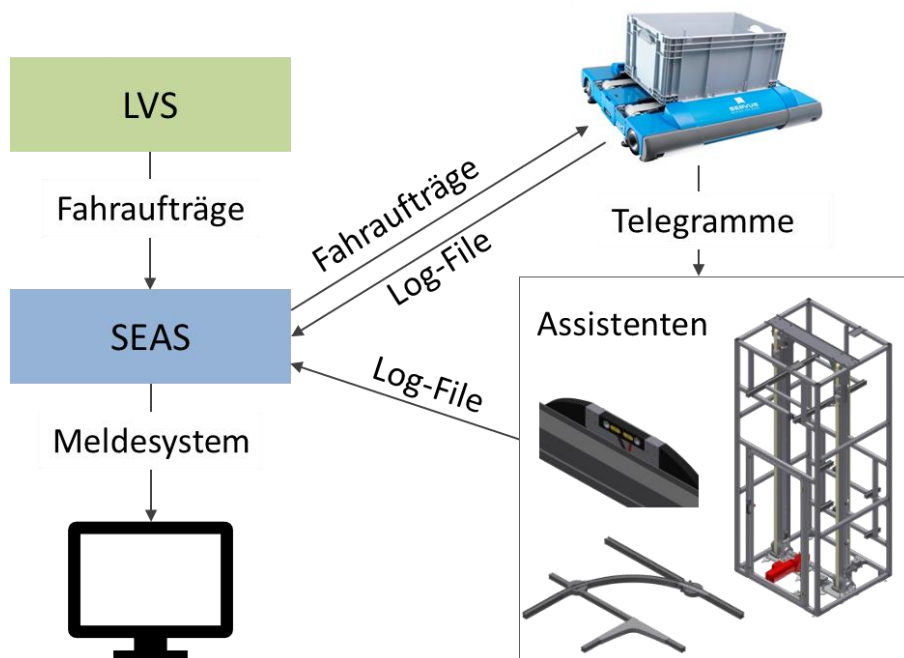


Abbildung 1: Servus-Systems aus Datensicht

2.1.1 Bestandteile des Servus-Systems

Das Servus-System ist aus vielen Einzelkomponenten aufgebaut. In Abbildung 2 ist ein beispielhaftes Layout einer Anlage aufgezeichnet welches aus folgenden Komponenten besteht:

1. Automatisches Kleinteilelager (AKL)
2. Pufferplätze
3. Arbeitsplätze
4. Weichen
5. Drehweichen
6. Fahrstrecke
7. Heber

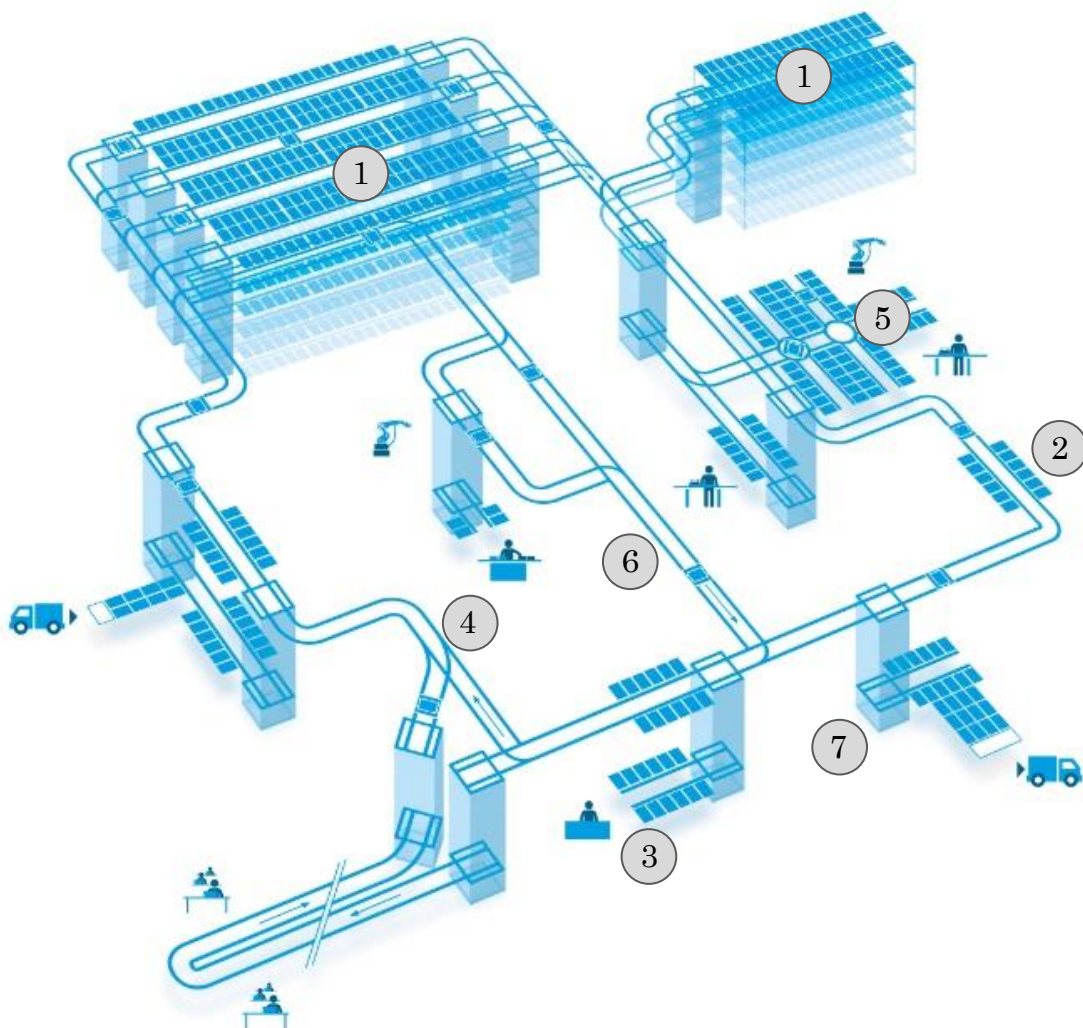


Abbildung 2: Beispiel eines Anlagenlayouts. 1) AKL, 2) Pufferplatz, 3) Arbeitsplatz, 4) Weiche, 5) Drehweiche, 6) Fahrstrecke, 7) Heber,

2.1.1.1 Fahrstrecke

Die Fahrstrecke ist mit Geraden und Kurven individuell konfigurierbar. Sie besteht aus zwei unterschiedlichen Profilen – dem Führungsprofil und dem Fahrprofil. Alle Streckenabschnitte zusammen bilden das Servus Streckensystem. Je nach Situation ist die Fahrstrecke von der Decke abgehängt, am Boden oder an der Wand montiert.



Abbildung 3: Servus Fahrstrecke

2.1.1.2 Transportroboter (ARC)

Die ARCs (Autonomous Robotic Carrier) bewegen sich mit einer Geschwindigkeit von max. 2,0 m/s selbständig auf dem Streckensystem bzw. auf der Fahrstrecke im Regal. Jeder ARC steuert sich gemäß seinem aktuellen Fahrauftrag autonom und entscheidet situationsabhängig, wann eine Transporteinheit auf- bzw. abzuladen ist.

Die ARCs können mit einem Lastaufnahmemittel ausgestattet sein, mit welchem er die Transporteinheiten auf- und ablädt. Dieses Lastaufnahmemittel wird vom ARC direkt angesteuert. Die ARCs sind standardmäßig mit einer Sensorik ausgestattet, um selbständig die Position der Transporteinheiten entlang der Strecke zu erkennen und diese beim Auf- bzw. Abladen zu überwachen.

Die ARCs kommunizieren über SERAN mit dem Auftragsverwaltungssystem und den Assistenten. Der ARC-Agent fungiert dabei als Kommunikationsvermittler zwischen ARCs und SEAS-Software-Komponenten (z.B. Auftragsverwaltung). Mit der Distanzmessung erkennen die Transportroboter Hindernisse bis max. 3m Entfernung, reagieren entsprechend selbstständig und fahren weiter, sobald das Hindernis entfernt ist.

Die Energieversorgung erfolgt über eine Kombination aus Superkondensator und Akku, welche über Ladestationen aufgeladen werden. Um die Ladezeiten mit den geplanten Stillstandszeiten des ARCs zu kombinieren, befinden sich die Ladestationen meist an den Umschlagsplätzen bzw. in den Hebern.

Für die Energieversorgung im Standby-Betrieb, insbesondere für die Speicherung von Daten, sorgt der Akku. Dieser gewährleistet einen Standby-Betrieb von ca. 14 Tagen.



Abbildung 4: ARC mit Lastaufnahmezinken

2.1.1.3 RFID-Knoten

RFID-Knoten dienen den ARCs als Informationsstellen zur Orientierung im Wegenetz. Sie sind an definierten Stellen entlang der gesamten Strecke positioniert. Nach einem Neustart bewegen sich die ARCs solange auf den Schienen vor und zurück, bis sie auf einen dieser RFID-Knoten treffen und somit ihren Standort erkennen.



Abbildung 5: Beispiel für einen RFID-Knoten

2.1.1.4 Weiche/Einmündung

Mit Weichen/Einmündungen können Streckenabschnitte aufgetrennt und zusammengeführt werden. ARCs können so ohne Stillstand den Streckenabschnitt wechseln. Der ARC wählt aufgrund seines Routings autonom den schnellsten Weg zu seinem Ziel und signalisiert der Weiche/Einmündung über SERAN, wohin er fahren möchte. Der Weichenhebel der Stellglindeinheit und die Backe der Bereichsschließungseinheit werden entsprechend verstellt. Das Verstellen erfolgt elektrisch über einen Schrittmotor. Die Steuerungen der Schrittmotoren prüfen, ob die Stellglindeinheit und die Bereichsschließungseinheit die korrekte Position erreicht haben.

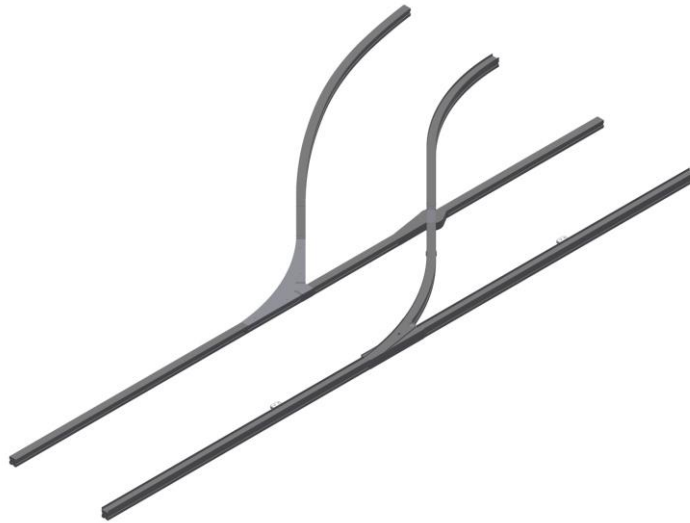


Abbildung 6: Beispiel für eine Weiche

2.1.1.5 Heber

Die Heber bestehen im Wesentlichen aus zwei oder vier Linearachsen und einem Verfahrschlitten, der entlang der Linearachsen vertikal verfährt. Der Schlitten wird durch einen Zahnriemen bewegt, der über die obere und untere Umlenkung geführt wird. Die Antriebsmotoren der Linearachsen sind über Verbindungswellen mit den Umlenkungen verbunden.

Die Steuerung der Heber erfolgt durch eine SPS, welche jeweils im zugehörigen Schaltschrank untergebracht ist.

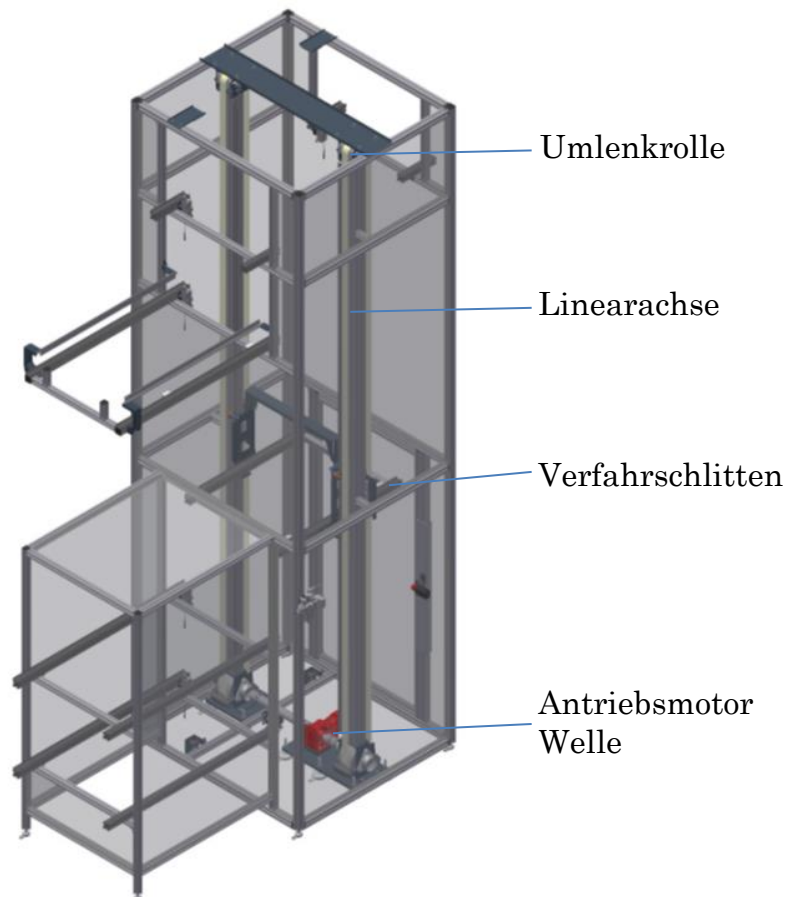


Abbildung 7: Beispiel für einen 2-achsigen Heber

2.1.1.6 Paternoster

Der Paternoster besteht im Wesentlichen aus einem Rahmen und zwei Antriebseinheiten. Jede Antriebseinheit besteht aus zwei Zahnriemen, die jeweils über die obere und untere Umlenkung geführt werden. Zwischen den Zahnriemen sind mehrere Streckenprofile installiert, auf welche der ARC auffährt.

Die Antriebseinheiten bewegen sich synchron gegeneinander und verfahren den ARC dadurch vertikal. Der Antriebsmotor ist über einen Zahnriemen und eine Antriebswelle mit den unteren Umlenkungen der Antriebseinheit verbunden.

Die Steuerung der Paternoster erfolgt auch hier durch eine im zugehörigen Schaltschrank untergebrachte SPS.

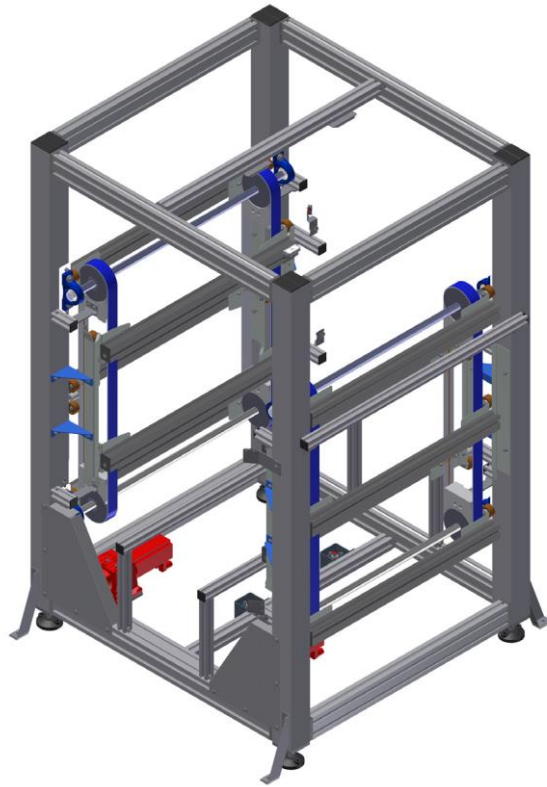


Abbildung 8: Beispiel für einen Paternoster

2.1.1.7 Arbeitsplätze

Arbeitsplätze sind Anbindungen zum Bedienungspersonal und dienen der Aus- und Einlagerung von Transporteinheiten. Über ein Display können hier Aufträge an das LVS erteilt werden.

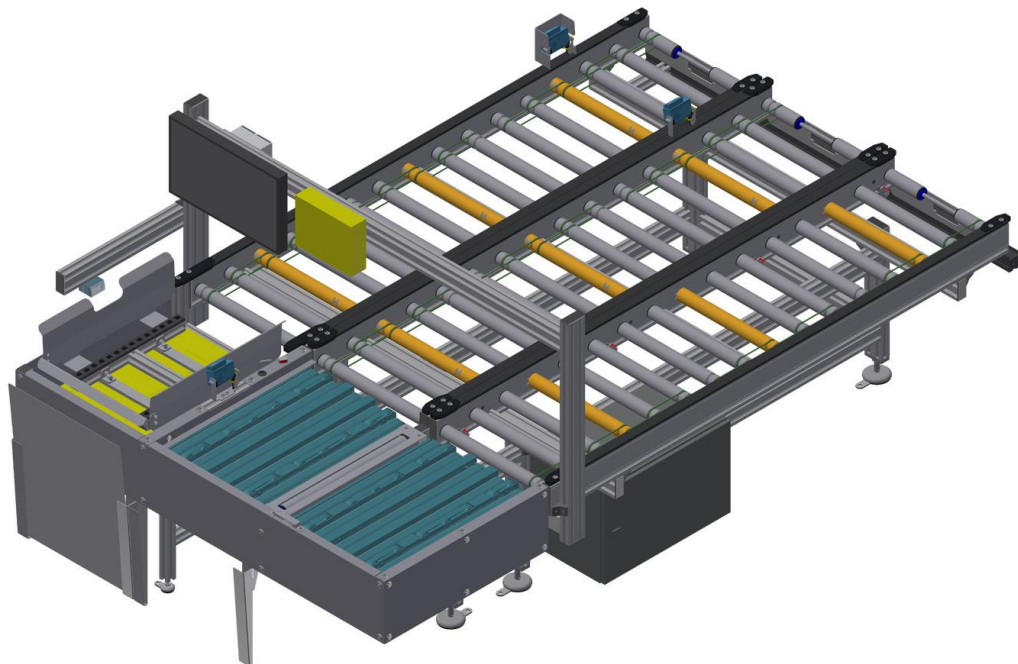


Abbildung 9: Beispiel für einen Arbeitsplatz

2.1.1.8 Pufferplätze

An den Pufferplätzen entlang der Strecke kann der ARC Ladehilfsmittel aufnehmen und abgeben. Sie dienen als temporäres Zwischenlager und zur Glättung von Leistungsspitzen.

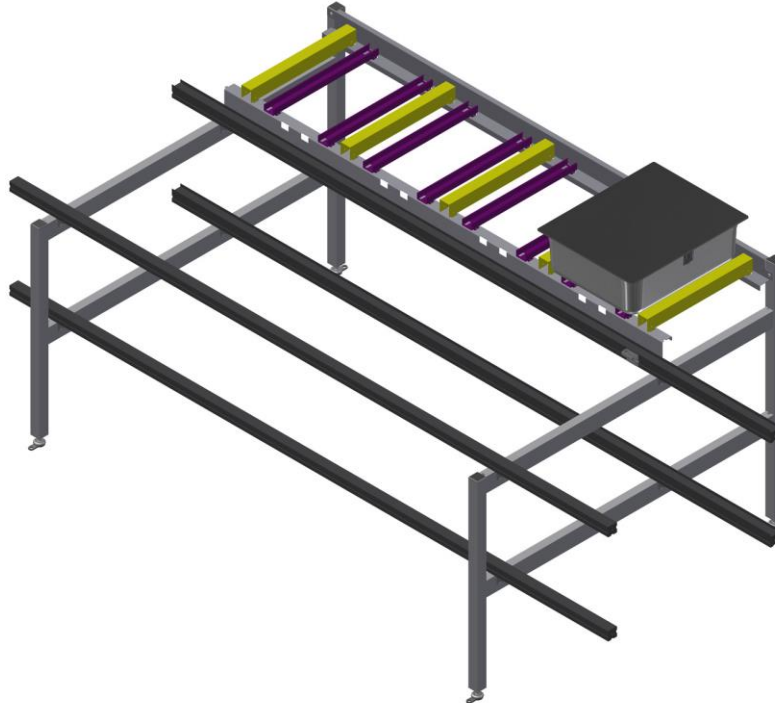


Abbildung 10: Beispiel für einen Pufferplatz

2.2 Datenbank

Um eine Auswertung der Kundenanlagen zu ermöglichen, wurde ein Maschinendatenerfassungstool erstellt, welches die Daten der SEAS-Datenbank beim Kunden an einen Analyseserver bei Servus schickt. Für diese Aufgabe werden Produkte der Firma Elasticsearch verwendet. Wie in Abbildung 11 ersichtlich, werden die Daten durch „logstash“ auf dem Kundenserver gesammelt und dann über eine VPN-Verbindung auf einer Elasticsearch-Datenbank bei Servus Intra-logistics gespeichert.

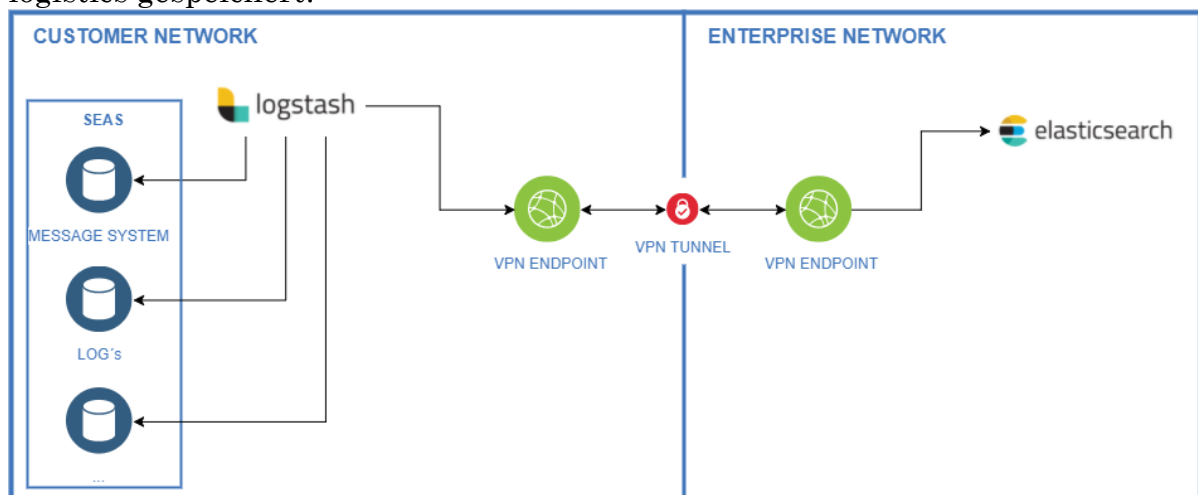


Abbildung 11: Ablauf der Datenerfassung

2.2.1 Elasticsearch

Bei Elasticsearch handelt es sich um eine Open-Source-Suchmaschine, die auf Lucene aufbaut und Daten in einem NoSQL-Format (JSON) speichert. Zum besseren Verständnis wird hier kurz auf die grundlegenden Konzepte von Elasticsearch eingegangen (vgl. [ELA19]).

Cluster

Ein Cluster besteht aus einem oder mehreren Knoten, auf denen die Daten gespeichert werden. Diese Verteilung der Daten verbessert die Speicher- und vor allem die Suchgeschwindigkeit.

Knoten

Als Knoten wird ein Server bezeichnet, der Teil eines Clusters ist. Je nach Aufgabe werden die Knoten (Nodes) in zwei Kategorien aufgeteilt:

- Master-Node: Ein einzelner Master-Node steuert sämtliche Knoten in einem Cluster.
- Data-Nodes: Diese Knoten enthalten die gespeicherten Daten und führen Operationen wie suchen, erstellen, löschen etc. durch.

Index

In einem Index werden Dokumente gespeichert, welche ähnliche Charakteristiken aufweisen. Ein Index kann beispielsweise alle abgeschlossenen Fahraufträge enthalten.

Dokument

Ein Dokument ist die grundlegende Informationseinheit, die in einem Index gespeichert werden kann. In unserem Fall ist dies z.B. ein einzelner Fahrauftrag oder eine Heberbewegung. Ein Dokument in Elasticsearch ist analog zu einer Zeile in einer SQL-Datenbank. Dabei ist aber zu beachten, dass die Dokumente schemalos sind. Die Dokumente in einem Index müssen daher nicht zwangsläufig die gleichen Eigenschaften besitzen. Dokumente werden im JSON-Format (JavaScript Object Notation) gespeichert. Ein Index kann beliebig viele Dokumente enthalten

Shard

Ein Index kann mitunter sehr große Datenmengen beinhalten und wird daher in Teilstücke, sogenannte Shards, aufgeteilt und auf verschiedene Knoten gelegt. Dies verbessert die Performance bei Suchanfragen und verhindert, dass ein einzelner Knoten speichertechnisch an seine Grenzen gerät.

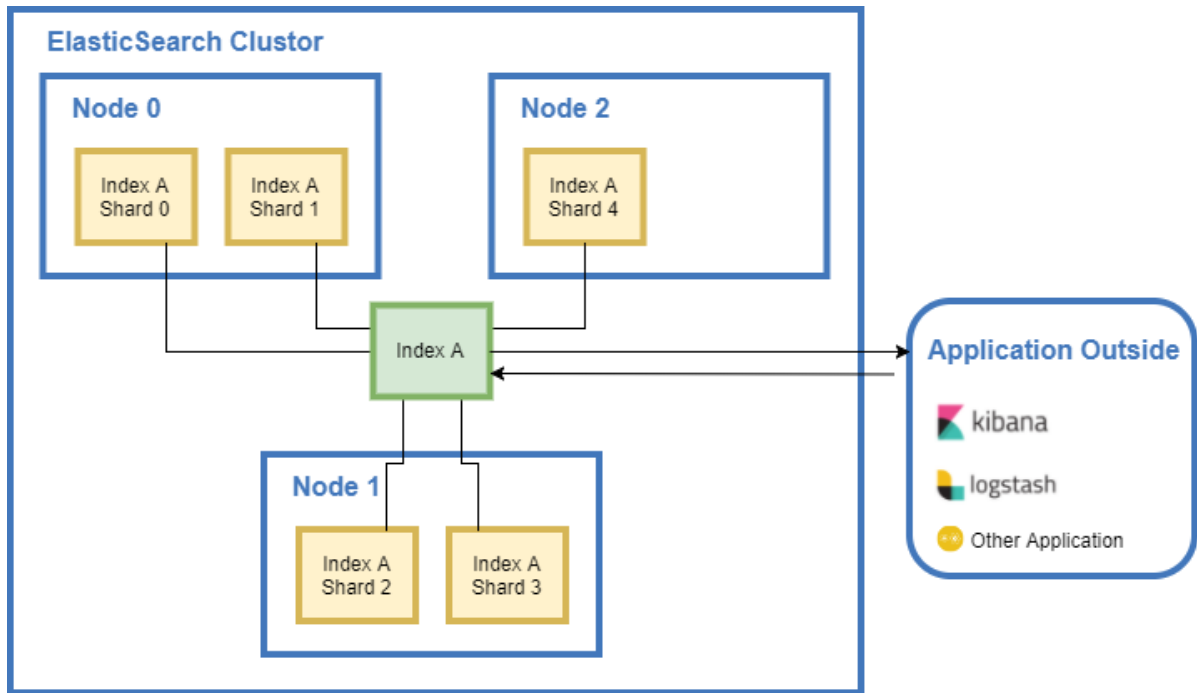


Abbildung 12: Aufbau eines Elasticsearch-Clusters

3 Modellbildung

3.1 Kennzahlen

Bei der Abwicklung von Logistikprozessen fallen eine Vielzahl von Informationen an. Da diese aber nur über eine begrenzte Aussagekraft verfügen und sich schwer interpretieren lassen, werden sie mit geeigneten Methoden und Modellen zu Kennzahlen verdichtet.

Grundsätzlich wird bei Kennzahlen, wie in Abbildung 13 dargestellt, zwischen Absolut und Verhältniszahlen unterschieden (vgl. [MAL79], S. 5)

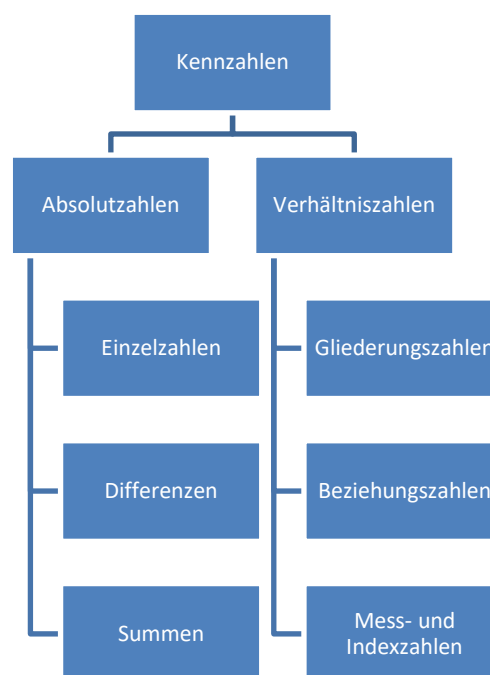


Abbildung 13: Gliederung von Kennzahlen (nach [MAL79], S. 5)

Die **Absolutzahlen** werden weiter in Einzelzahlen (z.B. Anzahl der ARCs im System), Differenzen (z.B. Differenz von SOLL- und IST-Fahrzeit) und Summen (z.B. Anzahl der beförderten Transporteinheiten) unterteilt.

Eine **Verhältniszahl** stellt den Quotienten zweier statistischer Messgrößen dar. Die Gesamtheit der beiden Größen kann identisch oder verschieden sein. Verhältniszahlen werden in Gliederungszahlen, Beziehungszahlen und Messzahlen unterteilt.

Eine **Gliederungszahl** gibt den Anteil einer bestimmten Teilmenge von der Grundgesamtheit an. Gliederungszahlen haben somit definitionsgemäß immer einen Wert zwischen null und maximal eins. In unserem Fall ist das Verhältnis von Transportzeit zur Gesamtzeit ein Beispiel für solch eine Gliederungszahl.

Mit Hilfe von **Beziehungszahlen** werden zwei prinzipiell verschiedene Größen, welche allerdings in einem sachlichen Bezug zueinander stehen, miteinander verglichen. Beispiel dafür sind die Anzahl der Fahrten pro Stunde und die Geschwindigkeit (m/s).

Von **Mess- und Indexzahlen** wird gesprochen, wenn zwei Messgrößen ins Verhältnis gesetzt werden, welche prinzipiell den gleichen Sachverhalt beschreiben, sich aber in einer zeitlichen, räumlichen oder in einer sonstigen Komponente unterscheiden.

Die Messgrößen, welche die Grundlage zur Berechnung der Indexzahlen darstellen, können beispielsweise zu verschiedenen Zeitpunkten oder von verschiedenen Anlagen erhoben werden. Eine solche Indexzahl wäre etwa der Vergleich der Transportfahrten aus zwei verschiedenen Jahren (vgl. [CK17], S. 65).

Für den Vergleich von verschiedenen Standorten, Komponenten etc., sind Verhältniszahlen den Absolutzahlen vorzuziehen, da diese viel bessere Anhaltswerte liefern. Der Vergleich der Fehlerhäufigkeit zweier Anlagen ist beispielweise weniger aussagekräftig als die Anzahl der Fehler pro ARC, weil bei ersterer die Größe des Systems nicht mitberücksichtigt wird.

Die Kennzahlen für unseren Anwendungsfall werden thematisch in folgende Bereiche unterteilt:

- Materialfluss
- Komponenten
- Verfügbarkeit

3.1.1 Materialfluss

- **Beförderte TE**

Die Anzahl der beförderten Transporteinheiten pro definierte Zeiteinheit stellt den „Key Performance Indicator“ des Servus-Systems dar (vgl. [VDI14], S. 8). Diese Kennzahl kann entweder als Absolutwert oder im Verhältnis zum vertraglich festgelegten Sollwert dargestellt werden.

- **Rundenzeiten**

Als Rundenzeit wird jene Zeit verstanden, die ein ARC benötigt, um eine Transportfahrt von A nach B zu absolvieren, dort einen neuen Behälter aufzuladen und diesen zurück nach A zu bringen.

Die Rundenzeit stellt einen Absolutwert dar. Dieser kann verwendet werden, um die Annahmen, welche bei der Projektierung der Anlage getroffen wurden zu kontrollieren und bessere Anhaltswerte für zukünftige Projekte zu erzielen.

Damit die Rundenzeit für aussagekräftigere Leistungsanalysen verwendet werden kann, muss sie mit anderen Zeitpunkten verglichen werden – sprich zu einer Indexzahl gemacht werden. Somit können eventuell auftretende Staus durch Überlastung, Verbesserungen bzw. Verschlechterungen durch Updates oder Umbauten erkannt werden.

- **Doppelspiele**

Als Doppelspiel wird die Kombination aus einer Ein- und Auslagerfahrt verstanden (vgl. [MAR09], S. 346). Das bedeutet, dass der ARC beim Abgeben einer TE in derselben Station wieder einen Behälter mitnehmen kann. Doppelspiele reduzieren die Anzahl der Leerfahrten und steigern dadurch die Leistung der Anlage. Dieser Wert stellt somit eine wichtige Kenngröße zur Beurteilung der Systemeffizienz dar.

3.1.2 Komponenten

- **Auslastung**

Die Auslastung stellt ein Verhältniswert zwischen tatsächlicher und maximal möglicher Leistung dar. Sie kann sowohl von den ARCs, als auch von den einzelnen Assistenten berechnet werden. Bei der Darstellung der Assistentenauslastungen muss berücksichtigt werden, dass eine Auslastung von 100% gemäß Warteschlangentheorie nicht erreichbar ist und dies dem Benutzer auch so mitgeteilt werden muss (vgl. [GUD10], S. 494). Die Auslastung kann zur Beurteilung der Effizienz des Systems und zur Auffindung von Bottlenecks verwendet werden.

- **Zeiten im Assistenten**

Die Gesamtzeit, die ein ARC zum Durchfahren eines Assistenten benötigt, kann in verschiedene Bereiche unterteilt werden. Beim Heber erfolgt die Unterscheidung zum Beispiel zwischen Warte-, Einfahr-, Verfah- und Ausfahrzeit. Dabei handelt es sich um Absolutwerte, welche der Projektierung als Anhaltswert für zukünftige Anlagen dienen, Überlastungen durch lange Wartezeiten anzeigen und Fehler (z.B. lange Kommunikationszeiten) aufzeigen.

- **Heberfahrten / Weichenbewegungen**

Um Aussagen über den Verschleiß der beweglichen Teile der Assistenten treffen zu können, ist es interessant wieviel Schaltzyklen eine Weiche hat oder wie oft ein Heber beschleunigt und wieder abbremst. Deshalb werden alle Assistentenbewegungen aufgezeichnet.

- **Grenzdurchsatz**

Jede Komponente der Logistikanlage hat einen Grenzdurchsatz. Darunter versteht man z.B. die maximale Anzahl an ARCs, die ein Heber von Knoten A nach B transportieren kann. Der Grenzdurchsatz limitiert somit die maximale Leistung der Anlage und kann zu Bottlenecks führen.

- **Aufgabenverteilung**

Unter Aufgabenverteilung versteht man, wieviel Zeit der ARC mit den verschiedenen Aufgaben, wie z.B. Parken, Leerfahrten oder Transportfahrten, etc. verbringt. Dies gibt neben der Auslastung auch Auskunft über die Effizienz der Anlage.

- **Fahrzeitenverteilung**
Um Optimierungspotential zu erkennen ist es wichtig zu wissen, wieviel Zeit der ARC für die einzelnen Abschnitte im System benötigt. Verbringen die ARCs beispielsweise bei einer Fahrt vom AKL zu einer Kommissionierstation nur 15% der Zeit auf der Strecke, hat es nur einen geringen Einfluss auf die Leistung, wenn die max. Fahrgeschwindigkeit erhöht wird.
- **Anzahl der ARCs im System**
- **Gefahrene Strecke der ARCs**
Wie schon bei den Heberfahrten gibt die gefahrene Strecke Auskunft über den Verschleiß der beweglichen Bauteile.
- **ARC Betriebsstunden**
Für alle Bauteile im ARC, deren Lebensdauer nicht von der gefahrenen Strecke abhängt, werden die Betriebsstunden für die Berechnung der Lebensdauer herangezogen.

3.1.3 Verfügbarkeit

- **Verfügbarkeit**
„Als Verfügbarkeit wird in der Theorie die Wahrscheinlichkeit bezeichnet, ein System zu einem vorgegebenen Zeitpunkt in einem funktionsfähigen Zustand vorzufinden (nach [FEM89], S. 3).“ Die genaue Berechnung der Verfügbarkeit und deren Mindestwert wird vertraglich zwischen Kunde und Auftragnehmer festgelegt.
- **“Mean time between failure” (MTBF) und „mean time to repair“ (MTTR)**
Die MTBF (durchschnittliche Betriebsdauer zwischen den Störungen) und die MTTR (durchschnittliche Dauer zur Fehlerbehebung) sind beides Absolutwerte, die dabei helfen den Verfügbarkeitswert besser zu verstehen. Durch die alleinige Betrachtung der Verfügbarkeit kann beispielsweise nicht bestimmt werden, ob auf der Anlage ständig Fehler auftreten oder ob die Reparatur bzw. Feststellung von vereinzelt Störungen lange Zeit in Anspruch nimmt (vgl. [VDI04], S.3).
- **Betriebsstörung: Fehlerart – Komponente – Ort**
Zur Analyse der auftretenden Fehler, werden bei jeder Betriebsstörung die Fehlerart, die betroffene Komponente und der Ort der Störung aufgenommen. Dies ermöglicht eine schnellere Erkennung und Behebung von Fehlerhäufungen.

3.2 Ablauf der Modellbildung

In den folgenden Abschnitten wird der Weg von den Ausgangsdaten, die wir auf dem Analyseserver vorliegen haben, zu den gewünschten Darstellungen beschrieben.

Dazu müssen, wie in Abbildung 14 gezeigt, zuerst die Ausgangsdaten aufbereitet werden. Mit diesen Daten können dann durch verschiedensten Methoden, die im vorherigen Abschnitt erklärten Kennzahlen berechnet werden.

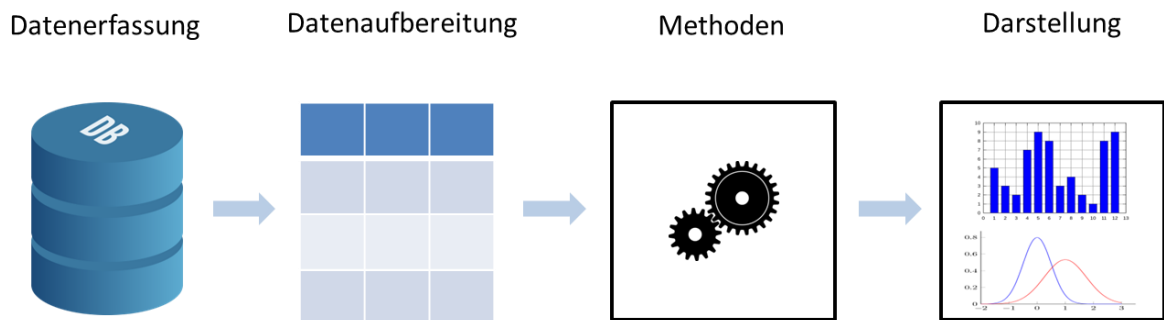


Abbildung 14: Ablauf der Modellbildung

Schematisch sieht das Datenmodell wie in Abbildung 15 dargestellt aus. Bei den Feldern in der linken Spalte handelt es sich um die Ausgangsdaten (rot), in der zweiten Spalte um die Datenaufbereitung (blau) und Hilfsmethoden (grün), in der dritten Spalte um die Methoden (grün) und rechts um die Kennzahlen bzw. aggregierten Daten, die dann für die Darstellung verwendet werden können.

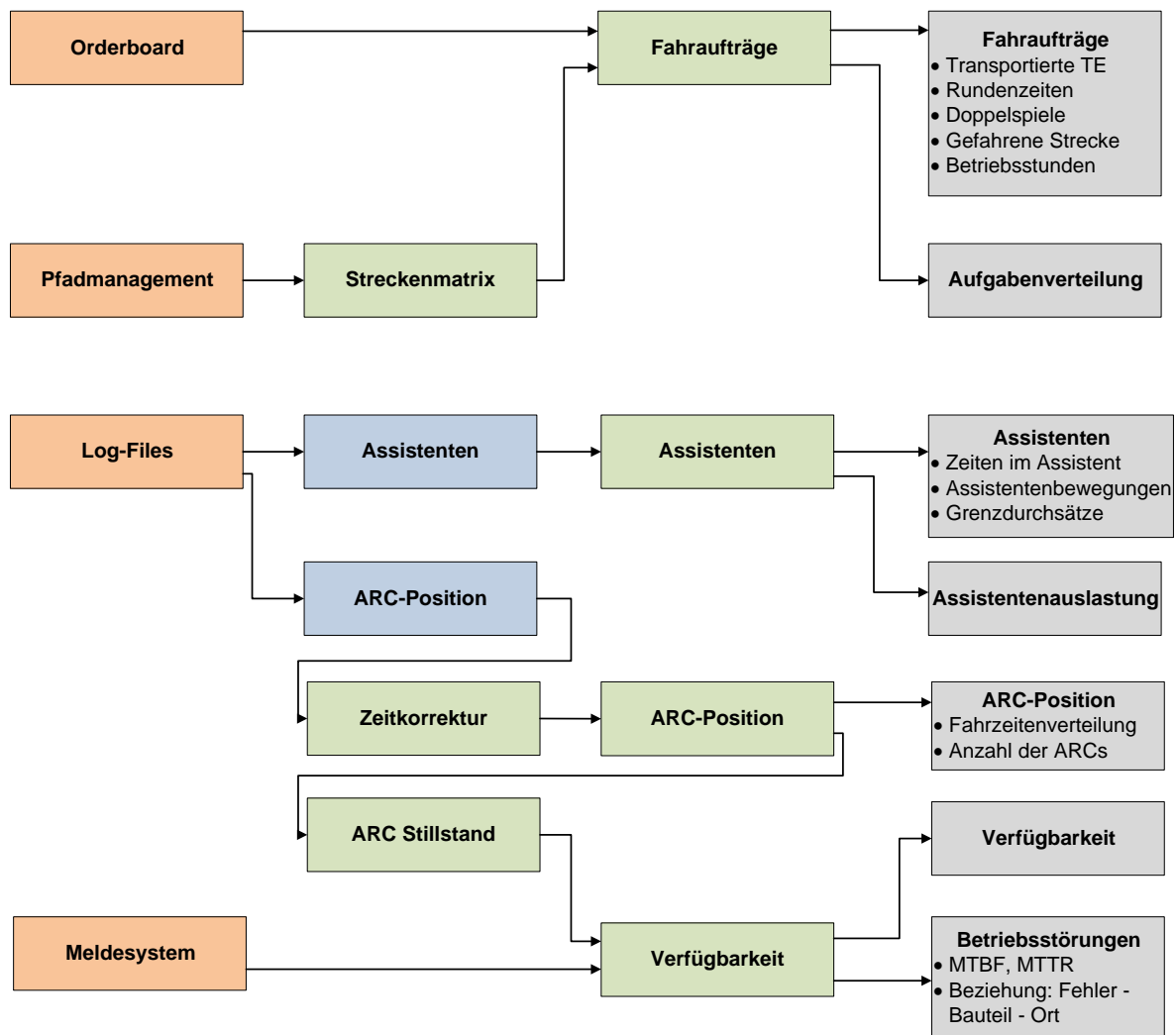


Abbildung 15: Schematischer Aufbau des Datenmodells mit den Ausgangsdaten (rot), der Datenaufbereitung (blau), den Methoden (grün) und den Kennzahlen (grau).

3.3 Ausgangsdaten

3.3.1 Orderboard

Im Orderboard sind alle Fahraufträge abgelegt, welche vom Lagerverwaltungssystem generiert und von den ARCs abgearbeitet werden. Für jeden Auftrag stehen unzählige Daten zur Verfügung. Die wichtigsten für unsere Auswertungen sind:

- Externe Auftragsnummer: Auftragsnummer des LVS
- Order-ID: Fortlaufende interne Fahrauftragsnummer
- Behälter-ID
- ARC-ID: Nummer des ARCs, welcher diesen Auftrag bearbeitet
- ARC-Aktion: Parken, fahren, abholen oder abgeben
- Zieladresse
- Zeitstempel, an denen der Auftrag erstellt, akzeptiert, gestartet und abgeschlossen wurde

3.3.2 Log-Files

Alle Komponenten des Servus-Systems, seien es Assistenten, ARCs, Dienste, Controller etc. speichern Log-Daten zur späteren Analyse ab. Ein Log-Eintrag hat immer folgenden Aufbau:

- Zeitstempel: Diesen erhält der Log-Eintrag beim Eintreffen auf dem Server
- Log-Level: Markiert die Wichtigkeit dieses Eintrags (Debug, Info, Warnung, fatal)
- Originator: Absendername
- Instanz
- ARC-ID
- Meldetext: Log-Nachricht als Volltext
- Thread: Hier werden Zusatzinformationen als Integerwert mitgeloggt (z.B. ARC-Laufzeit seit dem Aufwachen aus dem Stand-by)

3.3.3 Meldesystem

In diesem Abschnitt werden für den Kunden oder die Servus-Mitarbeiter relevante Meldungen gespeichert. Dies können Störungsmeldungen, allgemeine Informationen oder Warnungen sein. Die wichtigsten Informationen einer solchen Meldung sind:

- Absendername: Komponente oder Dienst, von der die Meldung stammt
- Einbauort: Ort, an der die Komponente verbaut ist oder an der sich der ARC gerade befindet
- Profil: Gibt an, ob es sich um eine Betriebsstörung oder Meldung handelt
- Meldetext
- Zeitpunkte, an denen die Meldung erstellt und wieder aufgehoben wurde

3.3.4 Pfadmanagement

Das Pfadmanagement enthält alle Daten der Knoten und Kanten im System.

Die wichtigsten Daten der Knoten sind:

- Knoten-ID
- Beschreibung: Kurze Positionsbeschreibung des Knotens
- Assistenten-ID: Falls sich der Knoten in einem Assistenten befindet, ist hier dessen ID angegeben
- Registrierung: Gibt an, ob es sich bei diesem Knoten um einen Anmeldepunkt handelt (z.B. Registrierungspunkt vor einem Heber)
- Ladestation: Markiert den Platz der Ladestationen

Für die Kanten stehen folgende Parameter zur Verfügung:

- Kanten-ID
- Beschreibung
- Startknoten
- Endknoten
- Erlaubte Fahrtrichtungen: Hier ist hinterlegt ob sich der ARC auf dieser Kante vorwärts, rückwärts oder in beide Richtungen bewegen darf
- Erlaubte Geschwindigkeit
- Länge der Kante in mm
- Kosten: Zur Berechnung des schnellsten Pfades sind für jede Kante sogenannte Kosten hinterlegt

3.4 Datenaufbereitung

Die Daten aus den Log-Files liegen als Volltext vor. Um sie für die Auswertungsmethoden nutzen zu können, müssen sie zuerst aufbereitet werden.

3.4.1 Assistenten

Für die Auswertung der Assistenten müssen die Telegramme, mit welchen die ARCs und Assistenten kommunizieren, aus den ganzen Log-Files herausgefiltert und aufbereitet werden. Auf die Bedeutung der einzelnen Telegramme wird im Kapitel 3.5.2.1 eingegangen. Der Log-Text eines solchen Telegramms sieht folgendermaßen aus:

```
[Processing ARC telegram: ((TelegramNo:107, Arc:1, TelegramType:Preregistration, ArcAssistantBehavior:SingleLift, AssistantInstance:11, AckRequired: True), SourceVertex:32592, DestinationVertex:31591, Order: 5924874, ArcWantsToLoadBattery: True, IsConditionalRegistration: False)] 20 0C 00 0B 00 01 6B (01) 00 00 7F 50 00 5A 68 0A 7B 67 00 05
```

Für unsere Anwendung benötigen wir den Telegrammtyp, den Startknoten (SourceVertex) und den Zielknoten (DestinationVertex).

3.4.2 ARC-Position

Die ARCs senden alle zwei Sekunden und beim Überfahren eines Knotens ihre aktuelle Position an den Log-Server. Der Eintrag sieht dabei folgendermaßen aus:

```
Akt.Order 18859637:2:1 (t=13001 h=20201 i=0) OrderInProgressDriving
```

Daraus können wir den Knoten hinter dem ARC (TailVertex: t), den Knoten vor ihm (HeadVertex: h), die aktuelle Position auf der Kante (i) und der aktuelle Auftragsstatus auslesen.

3.5 Methoden

3.5.1 ARC-Aufträge

Mittels ARC-Auftragsmethode werden aus den Daten des Orderboards die einzelnen Aufgaben ermittelt. Jede Aufgabe besteht dabei aus den in Abbildung 16 aufgelisteten Eigenschaften.

ARC-Auftrag
I: Anlagenname (t)
I: Arc (#)
I: Auftrag (t)
I: Einlagerung (bool)
I: Auslagerung (bool)
I: Startzeit (time)
I: Endzeit (time)
I: Dauer [s] (#)
I: Distanz [m] (#)
I: Startadresse (t)
I: Startbereich (t)
I: Startbereich (Doppelspiel) (t)
I: Startbereich (indirektes Doppelspiel) (t)
I: Startbereich (AKL-Gasse) (t)
I: Zieladresse (t)
I: Zielbereich (t)
I: Zielbereich (Doppelspiel) (t)
I: Zielbereich (indirektes Doppelspiel) (t)
I: Zielbereich (AKL-Gasse) (t)
I: Doppelspiel (bool)
I: indirektes Doppelspiel (bool)
I: Gassenwechsel (bool)

Abbildung 16: Eigenschaften eines Fahrauftrags und deren Datentypen: Zahl (#), Text (t), Zeit (time) und boolesche Variable (bool)

3.5.1.1 Doppelspiele

Unter Doppelspiel wird verstanden, wenn ein ARC eine Ein- Auslagerfahrt kombiniert. Dafür werden sogenannte Doppelspielbereiche definiert. Bei jeder Einlagerfahrt erfolgt die Kontrolle, ob der ARC zuvor im selben Bereich eine Auslagerung getätigt hat. Im AKL kann zwischen drei verschiedenen Doppelspielarten unterschieden werden. Am effizientesten ist das Doppelspiel, bei dem

der ARC in derselben Gasse und Ebene die Fahrten kombiniert und somit kein Assistent benötigt. Beim indirekten Doppelspiel bleibt der ARC in derselben Gasse, muss aber die Ebene mithilfe einer Heberfahrt wechseln. Die ineffizienteste Art ist der Gassenwechsel. Hier werden zwei Heberfahrten benötigt, um nach dem Abgeben die nächste TE wieder aufzunehmen.

3.5.1.2 Gefahrene Strecke

Die gefahrene Strecke der ARCs kann entweder direkt über die Positionsdaten oder indirekt mittels Fahraufträge geschehen.

3.5.1.2.1 Direkte Ermittlung der gefahrenen Strecke

Wie schon bei der Datenaufbereitung beschrieben, senden die ARCs alle zwei Sekunden und beim Überfahren eines Knotens ihre aktuelle Position. Mit Hilfe der Pfadnetzdaten, in denen jeder Kante eine Länge zugewiesen ist, kann so die gefahrene Strecke aufsummiert werden.

3.5.1.2.2 Indirekte Ermittlung der gefahrenen Strecke

Da sich die ARCs immer den kürzesten Weg zu ihrem Ziel suchen, kann im Vorfeld eine Matrix mit den Entfernungen der einzelnen Adressen zueinander aufgestellt werden. Zur Ermittlung der zurückgelegten Strecke kann dann die Start- und Zieladresse jeder einzelnen Fahrt in diese Matrix eingesetzt werden.

Beispiel:

In Tabelle 1 ist ein Auszug der ermittelten Fahraufträge mit den jeweiligen Start- und Zieladressen dargestellt. Anhand der Streckenmatrix, siehe Tabelle 2, können nun die zurückgelegten Strecken ermittelt werden. Bei der ersten Transportfahrt mit der Order-ID 507264 werden, wie in der Streckenmatrix blau markiert, 27,1m zurückgelegt.

Tabelle 1: Fahraufträge mit Start- und Zieladressen

Order-ID	Status	ARC	Startadresse	Zieladresse	Strecke [m]
507264	Transport	1	102-904-01-011-02	102-901-01-001-01	27,1
507302	Anfahrt	1	102-901-01-001-01	102-904-01-011-02	32,6
507302	Transport	1	102-904-01-011-02	102-010-13-021-01	53,3
507313	Anfahrt	1	102-010-13-021-01	102-901-01-002-02	67,8
507313	Transport	1	102-901-01-002-02	102-901-01-001-01	7,3
507341	Parkfahrt	1	102-901-01-001-01	102-511-02-001-01	15,6
507342	Parken	1	102-511-02-001-01	102-511-02-001-01	0

Tabelle 2: Streckenmatrix

Zieladresse \ Startadresse	102-001-01-001-01	102-001-01-002-01	102-001-01-003-02	...	102-901-01-001-01	102-901-01-002-01	...	102-904-01-001-01	102-904-01-011-01
102-001-01-001-01		0.8	1.6		51.5	51		40.9	39.2
102-001-01-002-01	0.8		0.8		52.3	51.5		41.7	40
102-001-01-003-02	1.6	0.8			53.1	52		42.5	40.8
...									
102-901-01-001-01	42.3	43.1	43.9			0.5		30.9	32.6
102-901-01-002-01	41.8	42.6	43.4		7.3			31.4	33.1
...									
102-904-01-001-01	35.1	35.9	36.7		25.4	25.9			1.7
102-904-01-011-01	36.8	37.6	38.4		27.1	27.6		10.5	

3.5.1.2.3 Wahl der Methode

Der Vorteil der direkten Ermittlung liegt in der einfachen Implementierung, da die ARC-Positionsdaten auch für andere Auswertungen analysiert werden müssen. Für die indirekte Ermittlung spricht zum einen die geringeren Datenmengen, die verarbeitet werden müssen (ca. 30 Fahraufträge/h/ARC indirekt und ca. 2400 Positionsdaten/h/ARC direkt). Ein weiterer Vorteil für die indirekte Methode und auch ausschlaggebend für die Wahl dieser ist, dass für die Erstellung der Fahrauftragsdaten und somit aller Materialflusskennzahlen nur die Orderboarddaten und keine Log-Files benötigt werden. Dies bietet die Möglichkeit, eine minimalisierte Version des Auswertungstools zu erstellen, welche nicht auf die speicherplatzintensiven Log-Files aufbaut.

3.5.2 Assistenten

Um Aussagen über die Auslastung und durchschnittlichen Zeiten im Assistenten treffen zu können, wird jede Bewegung (voll und leer) aufgenommen. Informationen über den momentanen Zustand der Assistenten erhält man über die Telegramme, mit denen ARCs und Assistenten miteinander kommunizieren. Für jede Assistentenbewegung bzw. für jeden ARC, der den Assistenten passiert, werden dabei folgende Daten aufgenommen (Abbildung 17):

Assistent
I: Anlagename (t)
I: Assistent (t)
I: Typ (t)
I: Startzeit (time)
I: Endzeit (time)
I: OrderId (#)
I: Arc (#)
I: Quellknoten (#)
I: Senkenknoten (#)
I: Arcs im Assistent (#)
I: Gesamtzeit [s] (#)
I: Wartezeit [s] (#)
I: Einfahrzeit [s] (#)
I: Zeit im Assistenten [s] (#)
I: Ausfahrzeit [s] (#)
I: Durchfahrzeit [s] (#)
I: Grenzdurchsatz (#)

Abbildung 17: Eigenschaften einer Assistentenbewegung und deren Datentypen: Zahl (#), Text (t), Zeit (time) und boolesche Variable (bool)

Die Auslastungen werden für fixe Zeitabschnitte berechnet. Die Auflösung, z.B. viertelstündlich oder stündlich, wird in den Konfigurationsdaten definiert. Für jeden Zeitabschnitt werden die Eigenschaften aus Abbildung 18 aufgenommen.

Assistentenauslastung
I: ID (t)
I: Anlagenname (t)
I: Assistent (t)
I: Startzeit (time)
I: Endzeit (time)
I: Auslastung [%] (#)
I: Schaltbelastung [%] (#)
I: Auflösung [min] (#)
I: Schaltdauer [min] (#)
I: Arbeitszeit [min] (#)

Abbildung 18: Eigenschaften der Assistentenauslastung und deren Datentypen: Zahl (#), Text (t) und Zeit (time)

3.5.2.1 Telegrammtypen

Um den Ablauf und den Aufbau dieser Telegramme besser zu verstehen, wird im folgenden Beispiel die Fahrt eines ARCs über einen Heber beschrieben.

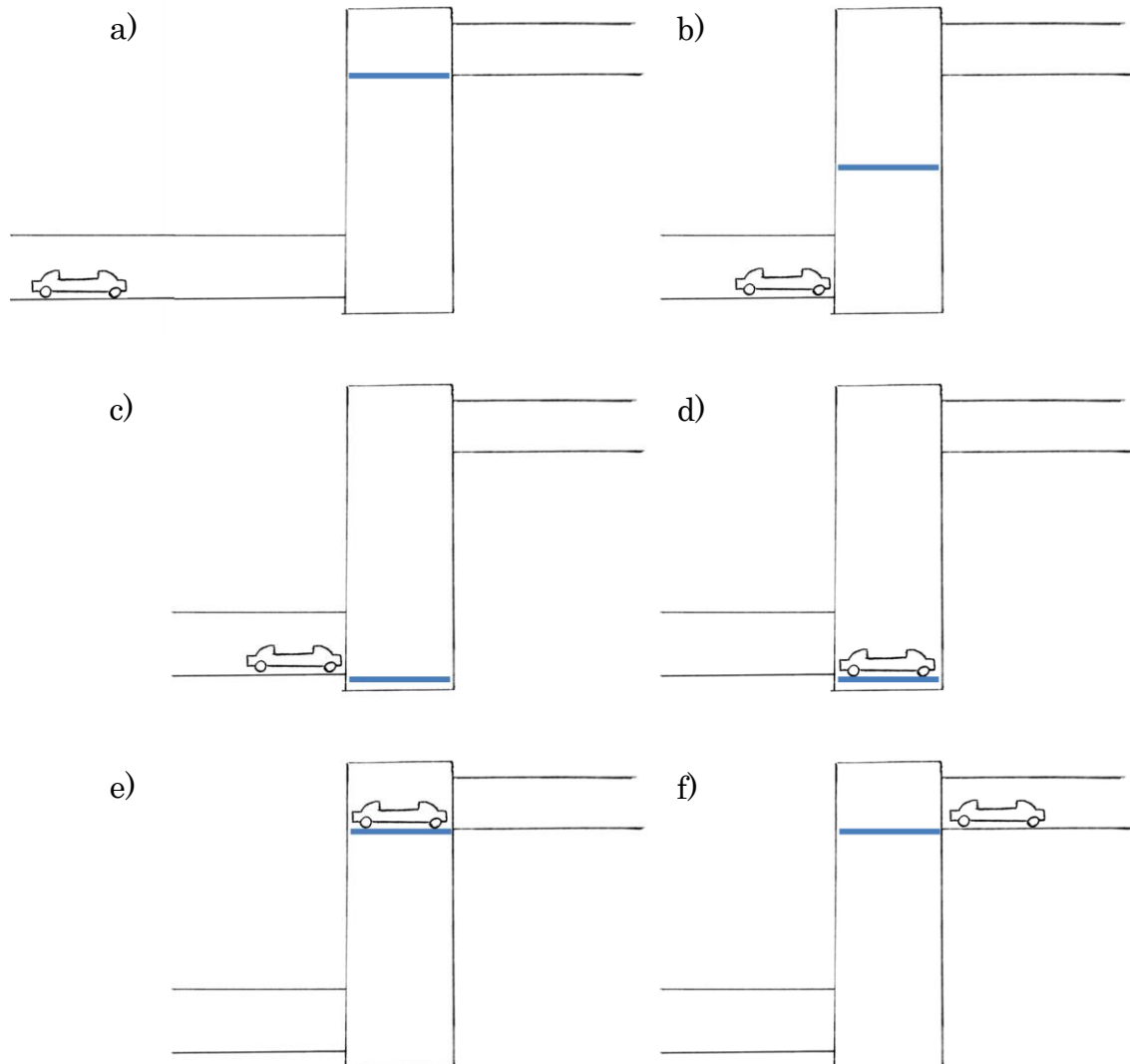


Abbildung 19: Telegrammtypen bei der Fahrt eines ARCs durch einen Heber: Vorregistrierung (a), Registrierung (b), Einfahrtfreigabe (c), Heberfreigabe (d), Ausfahrberechtigung (e), Abmeldung (f)

Wie in Abbildung 19 gezeigt, werden die Telegramme in folgende Typen unterschieden:

- **Vorregistrierung:** 4-5m vor dem Heber sendet der ARC eine Voranmeldung, damit der Heber, falls er unbesetzt ist, in die Richtige Position fahren kann. Dieses Telegramm enthält sowohl Start als auch den gewünschten Zielknoten.
- **Registrierung:** Wenn der ARC vor dem Heber angekommen ist, meldet er sich beim diesem an und sendet wie bei der Voranmeldung nochmal den Start- und Zielknoten.

- Einfahrtfreigabe: Der Heber erteilt dem ARC mit diesem Telegramm die Freigabe zur Einfahrt.
- Heberfreigabe: Der ARC teilt dem Heber mit, dass er sich auf dem zugewiesenen Platz positioniert hat und bereit zum verfahren ist.
- Ausfahrtfreigabe: Nachdem der Heber die Zielposition erreicht hat, erhält der ARC die Freigabe zum Ausfahren.
- Abmeldung: Der ARC meldet sich beim Überfahren des ersten RFID-Knotens ab und gibt den Heber somit für den nächsten Auftrag frei.
- Durchfahrtfreigabe: Manche Heber verfügen über eine Durchfahrtsebene (Start- und Zielknoten befinden sich auf gleicher Höhe). Bei diesen Hebertypen und bei Weichen fordern die ARCs die Erlaubnis an, durch den Assistenten durchzufahren.

Neben diesen Telegrammen werden auf die Start- und Endzeitpunkte jeder Assistentenbewegung mitgeloggt.

3.5.2.2 Grenzdurchsatz und Assistentenauslastung

Zur Beschreibung von Transportanlagen werden diese oft in Quellen und Senken aufgeteilt. Man stellt sich dabei vor, dass die Fördereinheiten, in unserem Fall die ARCs, in Quellen „erzeugt“ werden. Vernetzte Materialflusssysteme können durch Aneinanderreihung von Quellen und Senken in einzelne Komponenten unterteilt werden. Die Anzahl der Fördereinheiten, die sich in einer bestimmten Zeiteinheit von der Quelle zur Senke eines solchen Anlagenteils bewegen, wird als betrieblicher Durchsatz λ bezeichnet. Für die Durchsatzbetrachtung ist es nicht wichtig, nach welchem technischen Prinzip die Komponente funktioniert (vgl. [AF07], S. 11).

Betrachten wir beispielsweise ein Stück einer Fahrstrecke mit der Länge $l = 3,5m$. Die ARCs haben eine Maximalgeschwindigkeit v_{max} von $2 \frac{m}{s}$ und fahren mit einem Mindestabstand s_{min} von $1m$. Angenommen die ARCs fahren mit der Maximalgeschwindigkeit in den Abschnitt ein und aus, errechnet sich der maximale mögliche Durchsatz, auch Grenzdurchsatz γ genannt, wie folgt:

$$t_{min} = \frac{l}{v_{max}}$$

$$\gamma = \frac{1}{t_{min}} \cdot \frac{l}{s_{min}} = \frac{v_{max}}{s_{min}}$$

Die minimale Zeit t_{min} , die ein ARC zum passieren dieses Abschnitts braucht, beträgt 1,75 Sekunden. Somit liegt der Grenzdurchsatz bei $7200 \frac{1}{h}$. Im praktischen Betrieb muss die Bedingung

$$\lambda \leq \gamma$$

immer erfüllt sein. Das Verhältnis vom betrieblichen Durchsatz zum Grenzdurchsatz wird als Auslastungsgrad ρ bezeichnet. Es gilt:

$$\rho = \frac{\lambda}{\gamma} \leq 1$$

Zur Bestimmung des Grenzdurchsatzes von Assistenten (z.B. Hebern) werden deren Zwischenankunftszeiten ermittelt. Diese unterliegen statistischen Schwankungen. Schematisch sehen die Wahrscheinlichkeitsdichtefunktion und Verteilungsfunktion wie in Abbildung 20 dargestellt aus.

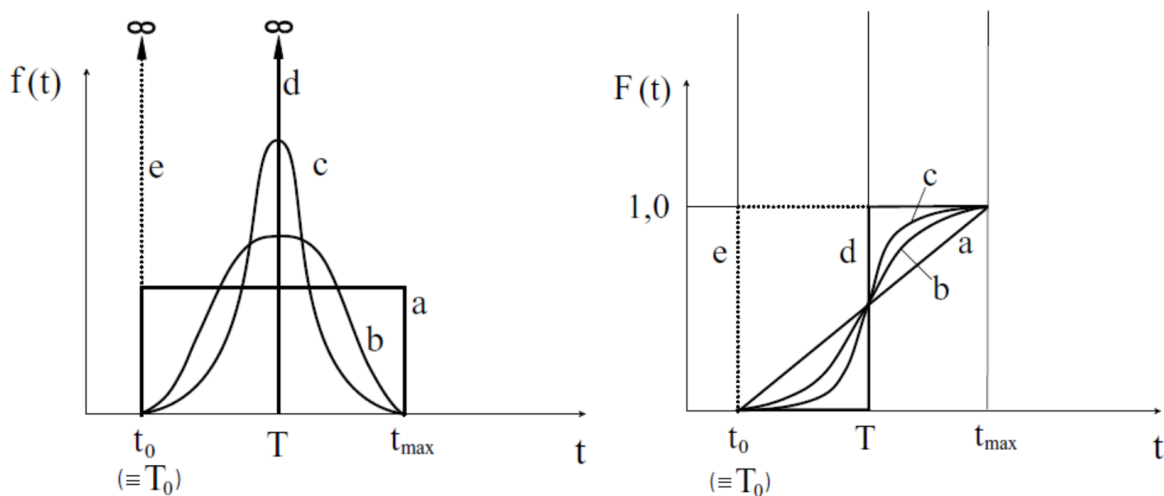


Abbildung 20: Wahrscheinlichkeitsdichtefunktion und Verteilungsfunktion für unterschiedlich streuende Zwischenankunftszeiten ([AF07], Abb. 2.9)

Mit der minimalen Zwischenankunftszeit T_0 errechnet sich der Grenzdurchsatz folgendermaßen:

$$\gamma = \frac{1}{T_0}$$

3.5.2.2.1 Verzweigungen

Mithilfe von Verzweigungen kann der Materialfluss einer Förderstrecke in unterschiedliche Richtungen gelenkt werden. Dabei wird zwischen aktiven und passiven Verzweigungen unterschieden. Der Heber stellt beispielsweise eine aktive Verzweigung dar, da er die ARCs mit einem eigenen Antrieb bewegt. Unter passive Verzweigungen fallen beispielsweise Weichen. Für den Grenzdurchsatz viel wichtiger ist jedoch die Unterteilung in stetig, teilstetig und unstetig.

3.5.2.2.1 Teilstetige Verzweigung

Von einer teilstetigen Verzweigung spricht man, wenn wie Abbildung 21 mindestens eine stetige und eine unstetige Verbindung vorhanden sind. Im Servus-System wäre dies ein Heber mit Durchfahrtsebene oder eine Drehweiche.

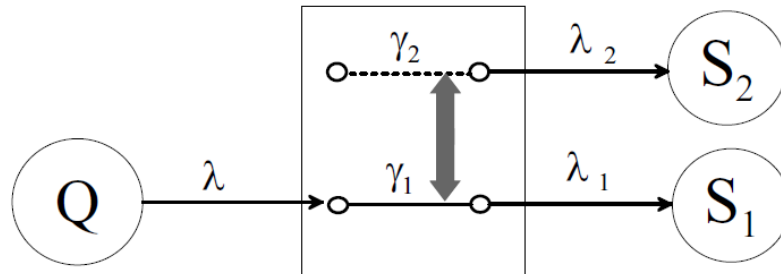


Abbildung 21: Teilstetige Verzweigung mit einer stetigen und einer unstetigen Verbindung ([AF07], Abb. 2.11)

Da für die Verbindung zur zweiten Senke das Fördermittel verfahren werden muss, gilt für die einzelnen Grenzdurchsätze:

$$\gamma_1 > \gamma_2$$

Für das gesamte System gilt folgende Durchsatzbedingung:

$$\frac{\lambda_1}{\gamma_1} + \frac{\lambda_2}{\gamma_2} \leq 1$$

Diese Bedingung sieht grafisch wie in Abbildung 22 dargestellt aus.

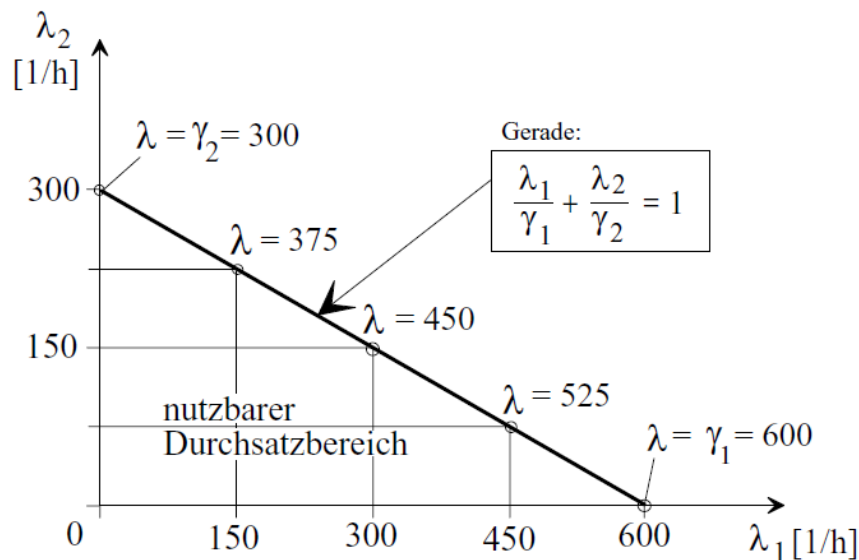


Abbildung 22: Durchsatzbedingung einer teilstetigen Verbindung ([AF07], Abb. 2.12)

Ganz allgemein formuliert, lauten die Durchsatzbedingung und somit auch der Auslastungsgrad für eine teilstetige Verzweigung mit mehreren Verbindungen:

$$\rho_j = \frac{\lambda_j}{\gamma_j}$$

$$\rho = \sum_{j=1}^m \frac{\lambda_j}{\gamma_j} = \sum_{j=1}^m \rho_j \leq 1$$

3.5.2.2.1.2 Unstetige Verzweigung

Bei der unstetigen Verzweigung liegt, wie in Abbildung 23 zu erkennen ist, keine stetige Verbindung vor. Da dieser Verzweigungstyp nur ein Spezialfall der teilstetigen Verzweigung ist, gelten die gleichen Durchsatzbedingungen.

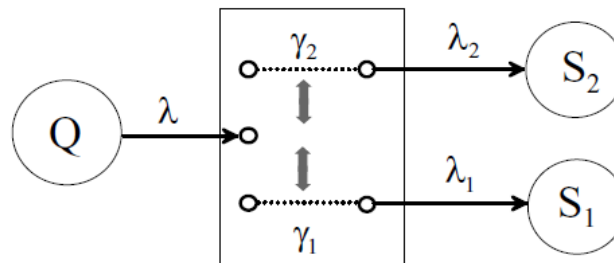


Abbildung 23: Unstetige Verzweigung mit zwei Verbindungen ([AF07], Abb. 2.14)

3.5.2.2.1.3 Stetige Verzweigung

Als Beispiel für die stetige Verzweigung wird, wie in Abbildung 24 dargestellt, eine Weiche mit zwei Richtungen verwendet. In der abgebildeten Stellung können die ARCs ungehindert in Richtung (1) fahren. Wird die Weiche um d verschoben, ist die Richtung (2) eingestellt. Wie bei der teilstetigen Verzweigung behandelt, hat jede Richtung ihren eigenen Grenzdurchsatz γ_j . Bei diesem Verzweigungstyp ist allerdings zu beachten, dass jede Schaltbewegung Zeit benötigt und somit der Grenzdurchsatz gemindert wird. Daher wird bei der stetigen Verzweigung die Durchsatzbedingung um einen Term, in dem die Schaltfrequenz ν und die Schaltzeit t_s mitberücksichtigt werden, erweitert. Für die in Abbildung 24 dargestellte Verzweigung sieht dies wie folgt aus:

$$\frac{\lambda_1}{\gamma_1} + \frac{\lambda_2}{\gamma_2} + \nu t_s \leq 1$$

$$\rho_1 + \rho_2 + \rho_s \leq 1$$

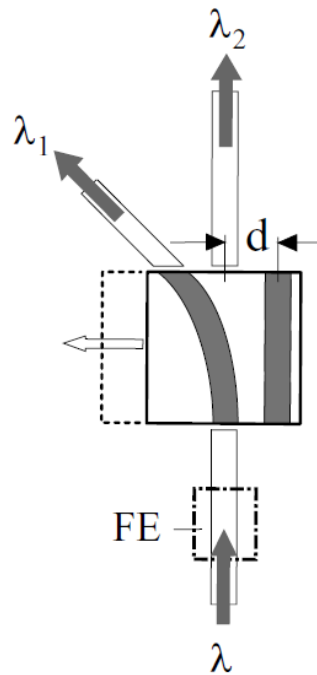


Abbildung 24: Stetige Verzweigung mit zwei Richtungen ([AF07], Abb. 2.10)

Für eine symmetrische, stetige Verzweigung mit zwei Verbindungen ist das Durchsatzdiagramm in Abbildung 25 dargestellt. Ist die Schaltzeit gleich null, treten, wie in Kurve (a) ersichtlich, keine Durchsatzverluste durch Schalten auf.

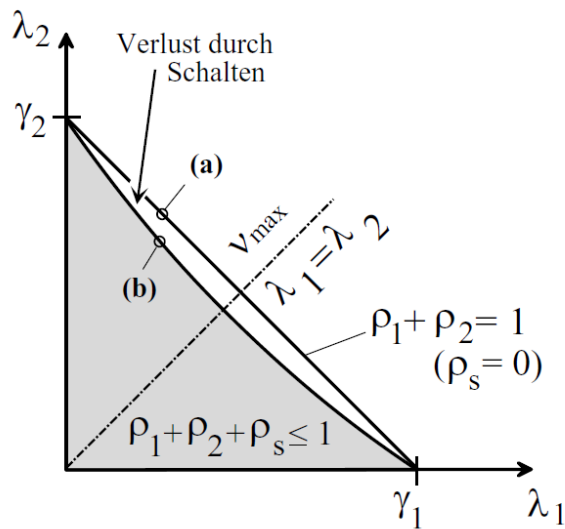


Abbildung 25: Durchsatzdiagramm einer stetigen Verzweigung ohne Durchsatzverluste (a) und mit Durchsatzverlusten (b) ([AF07], Abb. 2.16)

3.5.2.2.1.4 Universelles Materialflusselement

In den vorhergehenden Abschnitten wurden die unterschiedlichen Verzweigungsformen vorgestellt. Analog zu diesen gibt es auch Zusammenführungen, die gleiche Art berechnet werden. Im Gegensatz zur Verzweigung verfügen sie über mehrere Quellen aber nur eine Senke. Diese Betrachtung ist für einfache Assistenten, wie z.B. Weichen und Paternoster, ausreichend. Für die Beschreibung eines AKL-Hebers muss jedoch ein universelles Materialflusselement eingeführt werden, welches wie in Abbildung 26 dargestellt über n Quellen und m Senken verfügt.

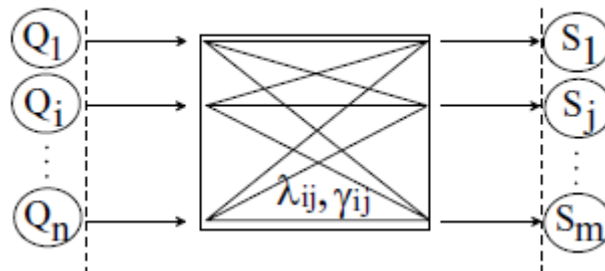


Abbildung 26: Universelles Materialflusselement ([AF07], Abb. 2.26)

Der Betriebliche Durchsatz errechnet sich hier mit:

$$\lambda = \sum_{i=1}^n \sum_{j=1}^m \lambda_{ij}$$

Die Auslastungsgrade ρ_{ij} errechnen sich mit den partiellen Grenzdurchsätzen γ_{ij} und den betrieblichen Durchsätzen λ_{ij} .

$$\rho_{ij} = \frac{\lambda_{ij}}{\gamma_{ij}} \leq 1$$

Die Durchsatzbedingung ergibt sich mit (vgl. [GUD10]):

$$\sum_{i=1}^n \sum_{j=1}^m \frac{\lambda_{ij}}{\gamma_{ij}} + \sum_{i=1}^n \sum_{j=1}^m v_{ij} t_{s_{ij}} \leq 1$$

3.5.2.2.2 Beispiel

Für dieses Beispiel betrachten wir einen Heber mit sechs Anschlüssen. Dabei gehen wir bei der Darstellung davon aus, dass alle Anschlüsse sowohl Quelle, als auch Senke sind.

Der Betriebliche Durchsatz in der ausgewählten Stunde beträgt:

$$\lambda = \begin{pmatrix} 0 & 2 & 41 & 0 & 7 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 46 & 0 & 0 & 7 & 2 \\ 0 & 5 & 9 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & 0 & 0 \end{pmatrix}$$

Hier ist ersichtlich, dass die Diagonaleinträge null sind. Bei den Knoten zwei und drei handelt es sich um Senken (horizontale Einträge null) und bei den Knoten eins und vier um Quellen (vertikale Einträge null).

Die minimalen Zwischenankunftszeiten betragen:

$$t_0 = \begin{pmatrix} - & 9 & 10 & - & 11 & 12 \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & 10 & - & - & 11 & 11 \\ - & 9 & 8 & - & - & - \\ - & 9 & 7 & - & - & - \end{pmatrix}$$

Die Schaltfrequenz beträgt:

$$v_s = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 12 & 1 & 1 \\ 9 & 0 & 0 & 0 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 2 & 0 \end{pmatrix} \cdot \frac{1}{3600}$$

Und die minimale Schaltzeit ist:

$$t_{s0} = \begin{pmatrix} - & - & - & - & - & - \\ 2 & - & - & 2 & 3 & 3 \\ 4 & - & - & - & 2 & 2 \\ 3 & - & - & 2 & - & 2 \\ 3 & - & - & 2 & 2 & - \end{pmatrix}$$

Damit errechnet sich der Auslastungsgrad mit:

$$\sum_{i=1}^n \sum_{j=1}^m \frac{\lambda_{ij}}{\gamma_{ij}} + \sum_{i=1}^n \sum_{j=1}^m v_{ij} t_{sij} = 0,35 + 0,03 = 0,38 \leq 1$$

In der betrachteten Stunde liegt die Auslastung dieses Hebbers bei 38%.

3.5.2.2.3 Auslastungsgrad mit Belegungszeiten

Eine weitere Möglichkeit zur Auslastungsberechnung, die in Betracht gezogen wurde, ist die Bildung des Quotienten aus aktiver Zeit zur Gesamtzeit.

Der Auslastungsgrad errechnet sich dann mit der Summe der Ankunfts- und Schaltzeiten dividiert durch den betrachteten Zeitraum t_{ges} .

$$\sum_{i=1}^p \frac{t_i}{t_{ges}} + \sum_{i=1}^{p_s} \frac{t_{s_i}}{t_{ges}} \leq 1$$

p ... Anzahl der beförderten ARCs
 p_s ... Anzahl der Schaltbewegungen

Wenn die Bewegungen in die einzelnen Richtungen aufgeteilt werden, sieht die Gleichung folgendermaßen aus:

$$\sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^p \frac{t_{ijk}}{t_{ges}} + \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^{p_s} \frac{t_{s_{ijk}}}{t_{ges}} \leq 1$$

$$\sum_{i=1}^n \sum_{j=1}^m \frac{\lambda_{ij} \bar{t}_{ij}}{t_{ges}} + \sum_{i=1}^n \sum_{j=1}^m v_{ij} \bar{t}_{s_{ij}} \leq 1, \text{ mit } \bar{t}_{ij} = \frac{\sum_{k=1}^p t_{ijk}}{\lambda_{ij}}$$

Für die Summe der einzelnen Zwischenankunftszeiten kann der betriebliche Durchsatz multipliziert mit der durchschnittlichen Zwischenankunftszeit herangezogen werden. Beim Vergleich dieser Formel mit jener aus Kapitel 0 wird deutlich, dass bei ersterer die durchschnittliche Zwischenankunftszeit und bei letzterer die minimale Zwischenankunftszeit zur Berechnung herangezogen wird.

Diese viel einfacher zu implementierende Auslastungsmethode darf folglich nur dann verwendet werden, wenn die durchschnittliche Zwischenankunftszeit nur geringfügig von der minimalen abweicht.

Für das vorherige Beispiel wurden folgende durchschnittlichen Zeiten ermittelt:

$$\bar{t} = \begin{pmatrix} - & 10 & 13 & - & 12 & 13 \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & 13 & - & - & 12 & 12 \\ - & 9 & 8 & - & - & - \\ - & 11 & 8 & - & - & - \end{pmatrix}$$

Somit ergibt sich die Auslastung zu:

$$\sum_{i=1}^n \sum_{j=1}^m \frac{\lambda_{ij} \bar{t}_{ij}}{t_{ges}} + \sum_{i=1}^n \sum_{j=1}^m v_{ij} \bar{t}_{s_{ij}} = 0,43 + 0,04 = 0,47 \leq 1$$

Die Abweichung liegt bei 9%, was für unsere Anwendung zu hoch ist.

3.5.3 ARC-Position

Aus den aufbereiteten Log-Daten erhalten wir die Positionen der ARCs. Daraus kann ermittelt werden, wie lange sich der ARC auf jeder Kante bzw. jedem Knoten aufhält. In Kombination mit dem Pfadmanagement, aus dem wir die Kantenlängen erhalten, kann dann die durchschnittliche Geschwindigkeit berechnet. Bei jeder Fahrt über eine Kante und beim Warten auf einem Knoten werden die Daten aus Abbildung 27 erfasst.

ARC-Position
I: Anlagename (t)
I: Arc (#)
I: Startzeit (time)
I: Dauer [s] (#)
I: Strecke [m] (#)
I: Geschwindigkeit [m/s] (#)
I: Quellknoten (#)
I: Zielknoten (#)
I: Kante (#)
I: Knoten (#)
I: Bereich (t)
I: Teilbereich (t)
I: Komponente (t)
I: Assistent (t)
I: Warteposition (bool)
I: Aufgabe (t)
I: Startbereich (t)
I: Zielbereich (t)

Abbildung 27: Eigenschaften der ARC-Positionsdaten und deren Datentypen: Zahl (#), Text (t), Zeit (time) und boolesche Variable (bool)

Diese ARC-Positionsdaten werden zur Ermittlung der Fahrzeitenverteilung und der ARC-Stillstände verwendet. Zusätzlich kann über den Verlauf der durchschnittlichen Geschwindigkeit auf Staus in gewissen Bereichen geschlossen werden.

3.5.4 Verfügbarkeit

Die Verfügbarkeit eines Intralogistiksystems kann auf zwei verschiedene Arten definiert werden. Zum einen gibt es die störungsbasierten Verfügbarkeitsrichtlinien und zum anderen die Leistungsverfügbarkeitsmodelle.

3.5.4.1 Störungsbasierte Verfügbarkeitsrichtlinien

„Unter Verfügbarkeit versteht man die Wahrscheinlichkeit, ein System zu einem vorgegebenen Zeitpunkt in einem funktionsfähigen Zustand anzutreffen ([FEM89], S. 3).“

Die Verfügbarkeit ist praktisch gesehen ein Maß für die nutzbare Betriebszeit/Einsatzzeit einer Anlage und damit auch abhängig davon, wie schnell ein Fehler im Durchschnitt behoben werden kann. Die Verfügbarkeit würde in diesem Fall wie folgt berechnet werden:

$$\eta_E = \frac{t_E - t_A}{t_E}$$

t_E ... Einsatzzeit
 t_A ... Ausfallszeit

Diese Formulierung ist bezogen auf die Realität zum Teil stark idealisiert, da bei komplexen Systemen ein Komponentenausfall nicht zwangsläufig ein Totalausfall der gesamten Anlage nach sich zieht.

Für die Anwendung in der Praxis empfiehlt sich daher, die Ausfallszeiten jeder einzelnen Komponente mit einem Gewichtungsfaktor zu multiplizieren. Dies sieht dann wie folgt aus:

$$\eta_E = \frac{t_E - \sum_{i=1}^n k_i \cdot t_{Ai}}{t_E}$$

t_E ... Einsatzzeit
 t_{Ai} ... Ausfallszeit der Komponente
 k_i ... Gewichtung der Komponente

Die Gewichtungsfaktoren werden meist mit dem Kunden zusammen definiert und vertraglich festgehalten (vgl. [FEM89]).

3.5.4.1.1 Beispiel für eine störungsbasierte Verfügbarkeitsrechnung

Um das vorhin beschriebene Konzept besser zu verstehen, wird es anhand einer kleinen Beispielanlage durchgespielt.

Die Anlage in unserem Beispiel besteht, wie in Abbildung 28 zu sehen, aus einem AKL mit zwei Gassen, einer Kommissionierzelle mit einem Ein- und zwei Auslagerbändern, einer Wareneingangs- bzw. Warenausgangszelle und 10 ARCs.

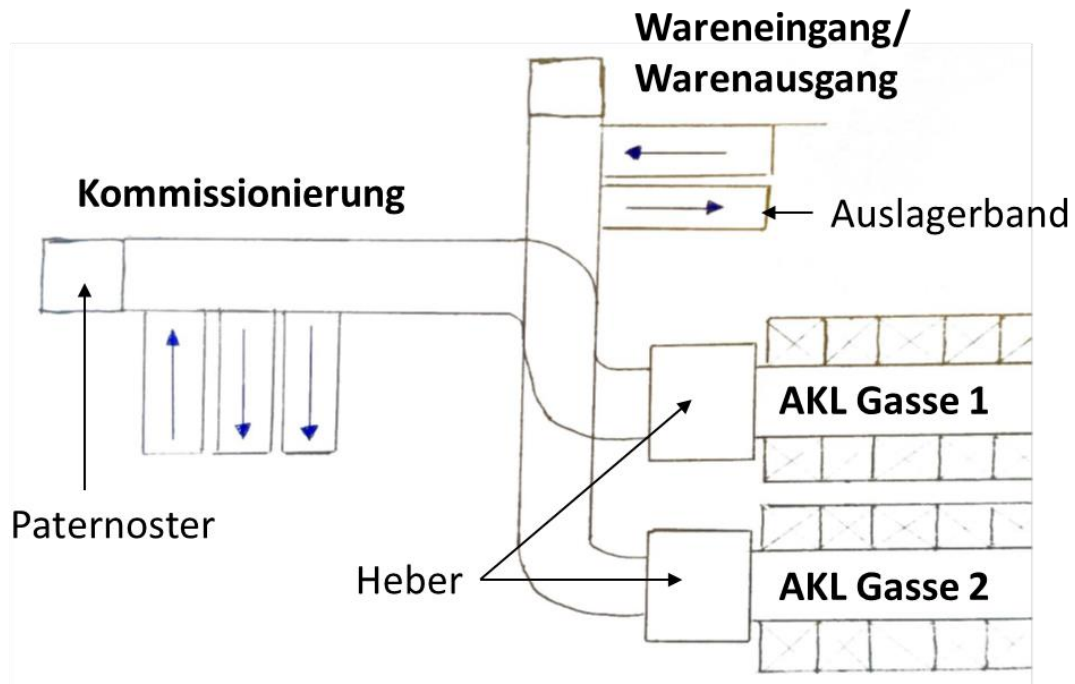


Abbildung 28: Layout der Beispielanlage

Die 100% der Gesamtanlagenverfügbarkeit müssen nun auf die einzelnen Anlagenkomponenten aufgeteilt werden. In unserem Beispiel wird angenommen, dass der Wareneingang bzw. der Warenausgang für den Kunden von geringer Bedeutung ist und daher nur 10% der Gesamtverfügbarkeit ausmacht. Die restlichen Komponenten werden mit jeweils 30% angesetzt. Da das AKL beispielsweise über zwei Gassen verfügt, sind bei einem Ausfall von einem Heber nur 50% von 30% und somit insgesamt 15% als ausgefallen anzusehen. Dazu sei noch angemerkt, dass ein Ausfall einer Komponente dann vorliegt, wenn diese einen Fehler aufweist, durch ein anderes fehlerhaftes Anlagenteil blockiert wird oder nicht mehr bedient werden kann. In Abbildung 29 sind die Gewichtungen in einem Verfügbarkeitsbaum dargestellt.

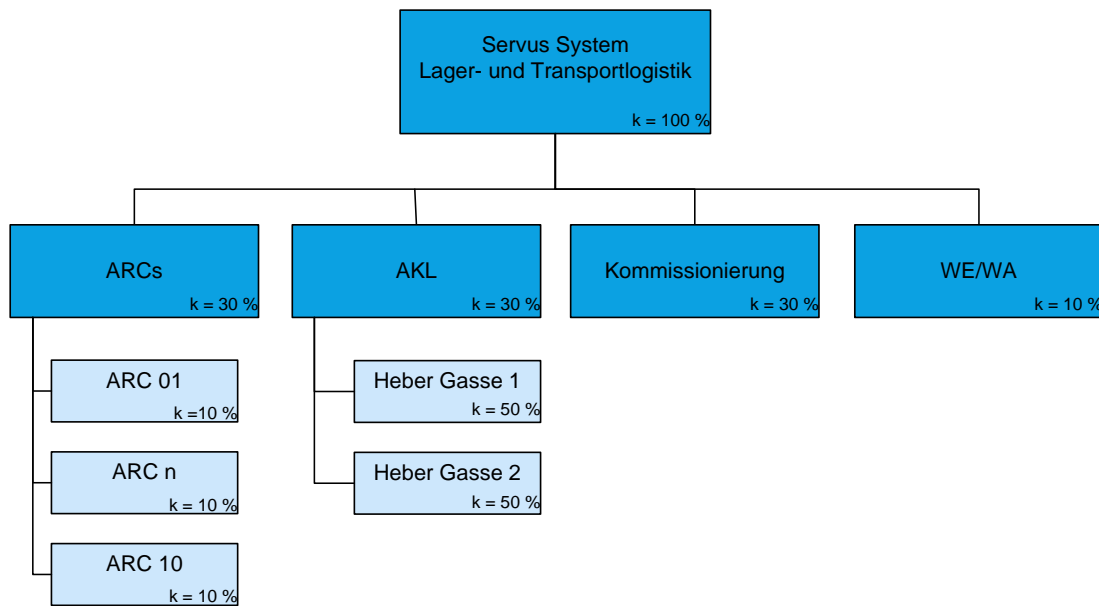


Abbildung 29: Verfügbarkeitsbaum der Beispielanlage

In Abbildung 30 ist ein Beispiel einer Störung dargestellt. Der ARC1 fährt dabei in die Kommissionierstation, um eine TE aufzunehmen. Während dieses Vorgangs tritt ein Fehler auf und der ARC bleibt stecken. Ab diesem Zeitpunkt werden der ARC1 und die Kommissionierung als ausgefallen gezählt. Zwei Minuten nach Entstehung des Fehlers kommt der ARC2 in denselben Bereich und bleibt ebenfalls stehen, da er durch den ARC1 blockiert wird. Nach weiteren 15 Minuten kann die Störung durch das Anlagenpersonal behoben werden und der Stau löst sich auf.

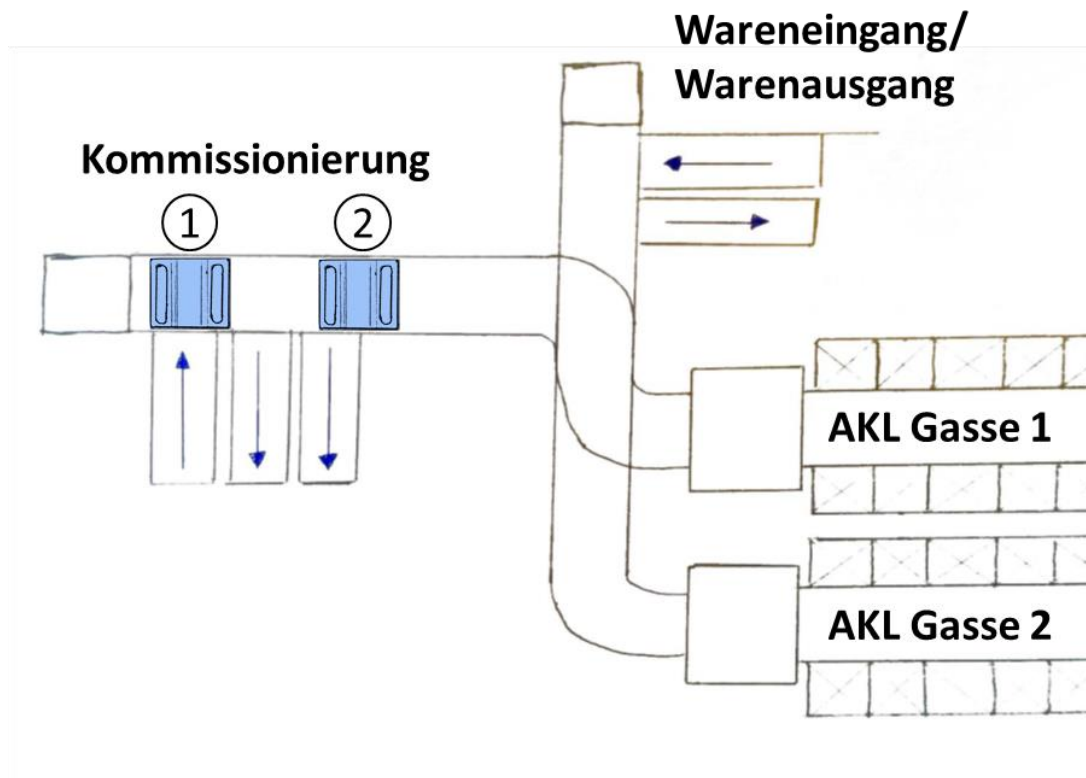


Abbildung 30: Beispiel einer Störung der Anlage

Angenommen dies ist die einzige Störung, errechnet sich die Verfügbarkeit für einen achtstündigen Arbeitstag ($t_E = 480\text{min}$) wie folgt:

Tabelle 3: Verfügbarkeitstabelle

Komponente	Ausfallszeit t_{Ai} [min]	Gewichtung k_i	Gewichtete Ausfallszeit [min]
ARC1	17	3%	0,51
Kommissionierung	17	30%	5,1
ARC2	15	3%	0,45
Gesamt			6,06

$$\eta_E = \frac{t_E - \sum_{i=1}^n k_i \cdot t_{Ai}}{t_E} = \frac{480 - 6,06}{480} = 98,7\%$$

Die Verfügbarkeit beträgt für diesen Tag 98,7%.

3.5.4.2 Leistungsverfügbarkeitsmodell

„Die Leistungsverfügbarkeit gibt den anforderungs- und termingerechten Erfüllungsgrad von zwischen Vertragspartnern (Hersteller und Anwender) vereinbarten Prozessen unter Einhaltung der vereinbarten Rahmenbedingungen an (vgl. [VDI10], S. 1).“

Diese Art der Verfügbarkeit leitet sich von den Geschäftszielen des Kunden ab.

Die Leistungsverfügbarkeit kommt infrage, wenn:

- der Geschäftsprozess definierbar ist,
- die Kenngrößen des Geschäftsprozesses messbar sind,
- Puffer, Redundanzen und Leistungsreserven berücksichtigt werden sollen und
- die Verantwortung für die Gesamtanlage in einer Hand liegt (Generalunternehmer).

Die Leistungsverfügbarkeit kommt nicht infrage, wenn nur einzelne Maschinen betrachtet werden.

Als Basis für die Berechnung können entweder Warte- oder Laufzeiten herangezogen werden. Kommt es bei einem Prozess aufgrund von fehlender Verfügbarkeit zu Wartezeiten (z.B. von Personal oder verknüpften Prozessen), wird die Leistungsverfügbarkeit η_W mit den aufsummierten Wartezeiten T_W und dem Beobachtungsintervall T_B ermittelt:

$$\eta_W = \frac{T_B - T_W}{T_B}$$

Lässt sich bei dem vorliegenden Prozess die Verfügbarkeit in verspäteten Ladeeinheiten n messen, wird die Leistungsverfügbarkeit η_L durch das Verhältnis von pünktlichen Ladeeinheiten zu der Anzahl N bei planmäßiger Leistung beschrieben (vgl. [VDI14]).“

$$\eta_L = \frac{N - n}{N}$$

Der Ablauf zur Ermittlung der Leistungsverfügbarkeit ist in Abbildung 31 dargestellt.

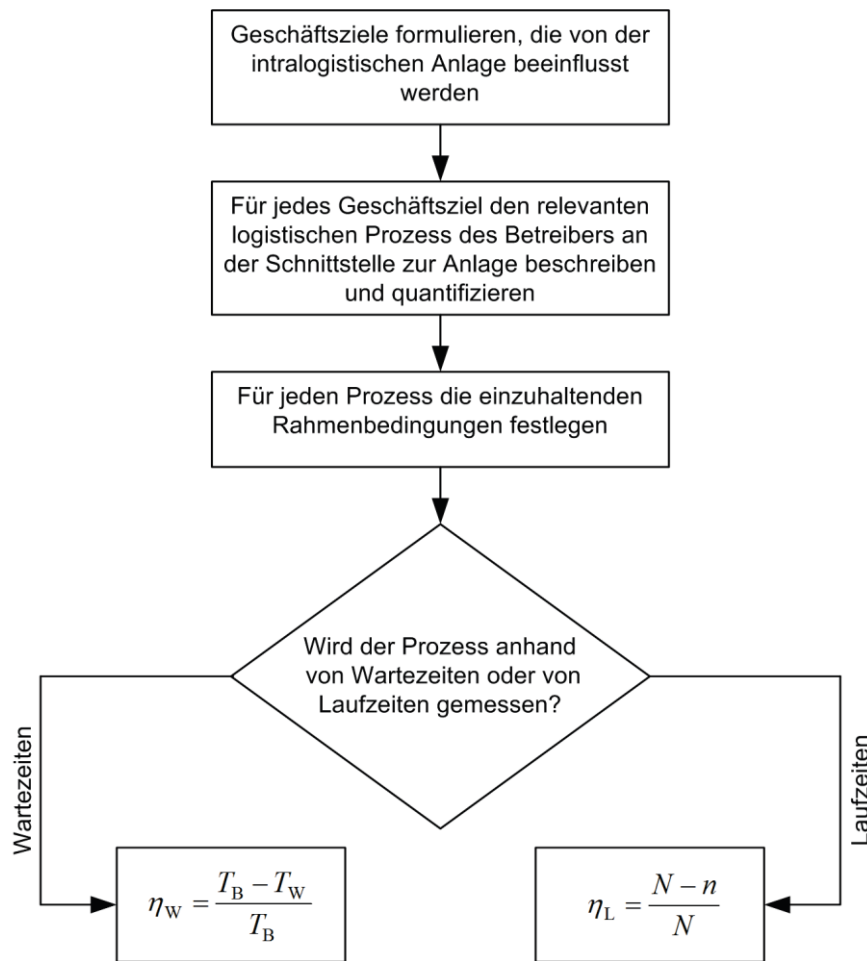


Abbildung 31: Ablaufschema zur Ermittlung der Leistungsverfügbarkeit ([VDI10], Abb.1).“

3.5.4.2.1 Beispiel für die Berechnung der Leistungsverfügbarkeit

Für dieses Beispiel verwenden wir wieder die Anlage aus Abbildung 28. Der vertraglich festgelegter Materialfluss sieht wie in Abbildung 32 dargestellt aus.

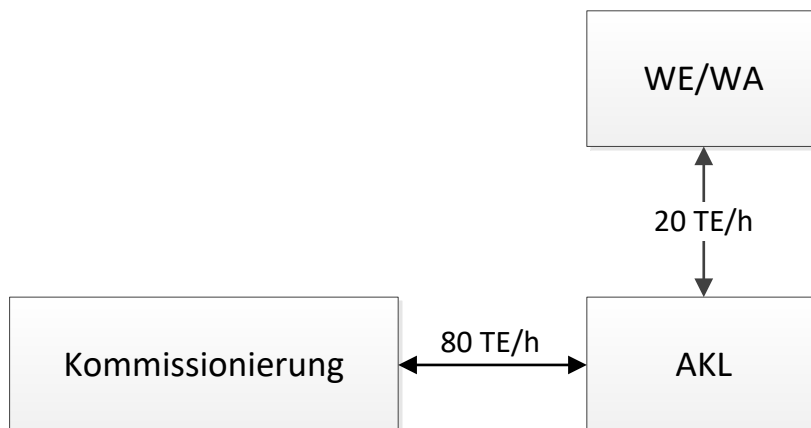


Abbildung 32: Vertraglich festgelegter Materialfluss

Die Taktzeit in der Wareneingangs-/Warenausgangsstation liegt bei 180s und von der Kommissionierstation bei 45s. Die Kommissionierstation gilt somit als nicht verfügbar, sobald der Arbeiter an dieser Station bei vorhandenem Arbeitsvorrat warten muss. Während des Beobachtungszeitraums werden alle Wartezeiten aufsummiert, wobei die Wartezeit erst nach Ablauf von 45s nach dem Aufgeben der letzten TE gilt.

3.5.4.3 Wahl der Verfügbarkeitsmethode

Da für die Servus-Anlagen meist kein genauer Geschäftsprozess abgegrenzt werden kann bzw. die Kunden die Anlage anders als projektiert betreiben, ist es meist nicht möglich, ein Modell für die Leistungsverfügbarkeit zu definieren. Es wird daher auf die störungsbasierte Verfügbarkeitsrechnung zurückgegriffen.

3.5.4.4 Anwendung der störungsbasierten Verfügbarkeitsrechnung

Für die Berechnung der Verfügbarkeit und die Analyse der Störungen werden für alle fehlerhaften und blockierten Komponenten die Daten aus Abbildung 33 ermittelt.

Störung
<ul style="list-style-type: none">• Anlagename (t)• ErrorId (#)• Komponente (t)• Zustand (t)• Bereich (t)• Verfügbarkeitsbereich (t)• Knoten (#)• Kante (#)• Startzeit (time)• Endzeit (time)• Gewichtung [%] (#)• Dauer [min] (#)• gewichtete Dauer [min] (#)• Fehlertext (t)

Abbildung 33: Eigenschaften einer Störung und deren Datentypen: Zahl (#), Text (t) und Zeit (time)

Um diese Daten zu erhalten, wird zuerst das Meldesystem auf Betriebsstörungen durchsucht. Mit Hilfe der ARC-Stillstandsmethode (Kapitel 3.5.5.3) werden die durch diese Störung blockierten ARCs gesucht. Mit Hilfe der Position der blockierten ARCs kann ermittelt werden, welche Bereiche von der Materialversorgung abgeschnitten sind. Diese Bereiche gelten dann ebenfalls als ausgefallen.

Für die Verfügbarkeit werden, wie schon bei der Assistentenauslastung, die Werte für Zeitbereiche mit einer vorgegebenen Auflösung berechnet. Die Datenstruktur sieht dabei folgendermaßen aus.

Verfügbarkeit
I: Anlagenname (t)
I: Startzeit (time)
I: Endzeit (time)
I: Auflösung [min] (#)
I: Verfügbarkeit [%] (#)
I: Ausfallszeit [min] (#)
I: gewichtete Ausfallszeit [min] (#)

Abbildung 34: Eigenschaften der Verfügbarkeit und deren Datentypen: Zahl (#), Text (t) und Zeit (time)

3.5.5 Hilfsmethoden

3.5.5.1 Streckenmatrix

Im normalen Fahrbetrieb suchen sich die ARCs immer den schnellsten Weg. Dies geschieht über Kantenkosten, welche im Pfadmanagement hinterlegt sind. Somit kann für die Streckenmatrix anfangs einmalig der schnellste Pfad von allen Adressen zueinander ermittelt werden.

Diese Aufgabe fällt unter den Bereich der Graphentheorie. Die Graphentheorie ist ein Teilgebiet der Mathematik. Ein Graph G besteht aus einer Menge $V = V(G)$ an Knoten (vertices) und einer Menge $E = E(G)$, den Kanten (edges) von G . E stellt dabei ein ungeordnetes Paar von Knoten dar. Die Anzahl der Knoten wird als n und die Anzahl der Kanten als m bezeichnet. Ein Graph kann, wie in Abbildung 35 dargestellt, in gerichtet und ungerichtet bzw. in Graphen mit Einfach- und Mehrfachkanten unterteilt werden (vgl. [LÄU91], S. 5). Beim Servus-System gibt es keine Mehrfachkanten, was bedeutet, dass es zwischen zwei benachbarten Knoten nur eine Kante gibt. Bei den Kanten wird folglich zwischen Einbahnstrecken und von beiden Seiten befahrbaren Streckenabschnitten unterschieden.

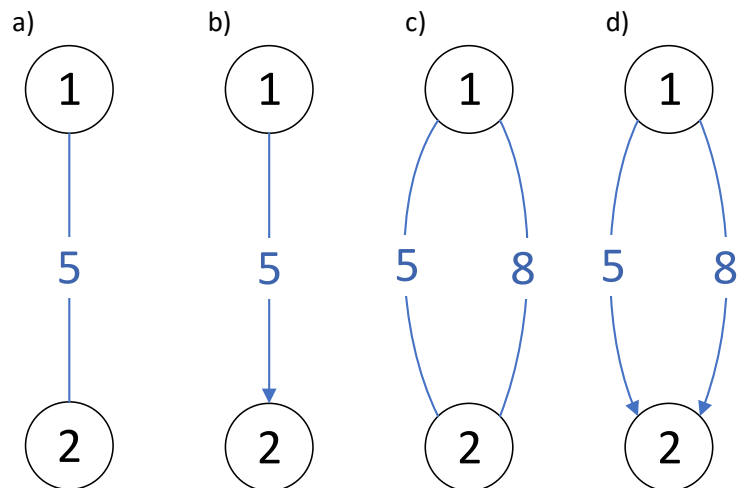


Abbildung 35: Arten von Graphen: ungerichteter Graph (a), gerichteter Graph (b), ungerichteter Graph mit Mehrfachkanten (c) und gerichteter Graph mit Mehrfachkanten (d)

In Abbildung 36 ist ein Beispielgraph dargestellt, der in weiterer Folge zur Veranschaulichung von Methoden der Graphentheorie herangezogen wird.

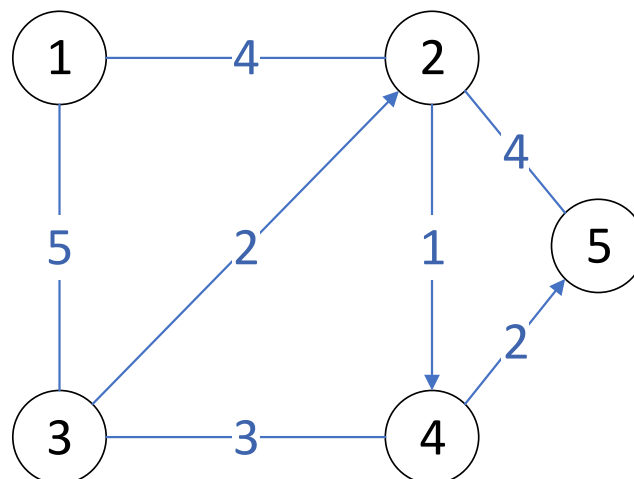


Abbildung 36: Beispielgraph mit $n = 5$ und $m = 7$

Um computergestützt Graphen-Algorithmen anzuwenden, benötigt man zweckmäßige Datenstrukturen für die Darstellung der Graphen. Diese können entweder durch kompakte Tabellen oder wie in unserem Fall durch sogenannte Adjazenzmatrizen dargestellt werden (vgl. [LÄU91], S. 11).

Die Adjazenzmatrix ist eine $n \times n$ Matrix $A = (a_{ij})$ mit den Komponenten

$$a_{ij} = \begin{cases} 1, & \text{falls eine Verbindung von Knoten } i \text{ nach Knoten } j \text{ existiert} \\ 0, & \text{sonst} \end{cases}$$

Die Adjazenzmatrix für den Graphen in Abbildung 36 sieht wie folgt aus:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Zur Ermittlung des kürzesten Pfads zwischen den Knoten eines Graphen stellt die Graphentheorie einige Algorithmen zur Verfügung.

Die bekanntesten sind:

- Dijkstra Algorithmus
- A* Algorithmus
- Bellman-Ford Algorithmus
- Floyd-Warshall Algorithmus

Der Unterschied zwischen diesen Algorithmen ist deren Anwendungsgebiet. Der Dijkstra- und A* Algorithmus sind beides Algorithmen, die dafür ausgelegt sind, den schnellsten Weg zwischen zwei spezifischen Knotenpaaren zu finden. Dies sind sogenannte „single-pair shortest-path“ Algorithmen. Will man mit solch einem Algorithmus eine komplette Streckenmatrix erstellen, muss jede Kombination der Knoten zueinander berechnet werden. Die Laufzeit für diese Operation beträgt im schlechtesten Fall $O(n^4)$.

Der Bellman-Ford Algorithmus findet den kürzesten Pfad zwischen einem Ausgangsknoten und allen anderen Knoten. Für die Streckenmatrix muss jeder Knoten einmal als Quelle eingesetzt werden, was eine Laufzeit von $O(n^2 \cdot m)$ bedeutet.

Der Floyd-Warshall Algorithmus findet mit einer Laufzeit von $O(n^3)$ den kürzesten Weg von allen Knotenkombinationen.

Das Pfadnetz des Servus-Systems ist statisch und somit kann anfangs eine Streckenmatrix mit den kürzesten Wegen aller Knoten zueinander aufgestellt werden. Die „single-pair shortest-path“ Algorithmen scheiden deshalb aus. Da das Wegenetz meist mehr Kanten als Knoten besitzt ($n < m$), hat der Floyd-Warshall Algorithmus einen leichten Laufzeitvorteil gegenüber dem von Bellman-Ford Algorithmus, weshalb für unsere Anwendung auf diesen zurückgegriffen wird.

3.5.5.1.1 Floyd-Warshall Algorithmus

Wie vorhin bereits erwähnt, berechnet der Floyd-Warshall Algorithmus den kürzesten Pfad zwischen allen Knotenpaaren. Der Ablauf wird am Graphen aus Abbildung 36 erklärt.

Zuerst wird die $n \times n$ Streckenmatrix initialisiert. Diese ist ähnlich der Adjazenzmatrix, wobei statt der Einträge mit Wert eins die jeweilige Kantenlänge, in der Diagonale null und für die restlichen Einträge unendlich eingetragen wird. In unserem Beispiel ergibt sich folgende Matrix.

$$S_0 = \begin{pmatrix} 0 & 4 & 5 & \infty & \infty \\ 4 & 0 & \infty & 1 & 4 \\ 5 & 2 & 0 & 3 & \infty \\ \infty & \infty & 3 & 0 & 2 \\ \infty & 4 & \infty & \infty & 0 \end{pmatrix}$$

Der Pseudocode für den Algorithmus sieht, wie in Abbildung 37 dargestellt, aus.

```

for k from 1 to n      // n ist die Knotenanzahl
  for i from 1 to n
    for j from 1 to n
      if distance[i][j] > distance[i][k] + distance[k][j]
        distance[i][j] := distance[i][k] + distance[k][j]
      end if
    next j
  next i
next k

```

Abbildung 37: Pseudocode für den Floyd-Warshall Algorithmus

Hierbei handelt es sich um drei ineinander verschachtelte for-Schleifen, die jeweils über alle Knoten laufen. Daraus ergibt sich auch die Laufzeit von $O(n^3)$. Wenn es von Knoten i zu Knoten j einen schnelleren Weg über Knoten k gibt, wird dieser verwendet.

In folgendem Beispiel werden zum Verständnis ein paar Schritte des Algorithmus durchgeführt. Der Index von S stellt dabei ijk dar.

Im Schritt $S_{ijk} = S_{123}$ wird überprüft ob der Weg von 2 nach 1 und dann von 1 nach 3 kürzer ist als der momentane Wert von 2 nach 3. Der aktuelle Wert von 2-3 ist unendlich und von 2-1-3 ist neun. Somit wird an die Stelle 2-3 der Wert 9 geschrieben.

Bei $S_{ijk} = S_{132}$ wird überprüft ob der Weg 3-1-2 kleiner ist als 3-2. Da dies nicht der Fall ist ($9 > 2$) wird der Wert zwei beibehalten.

Nachdem alle Knoten durchlaufen worden sind, ergibt sich die Streckenmatrix wie in S_{nnn} dargestellt.

$$S_0 = \begin{pmatrix} 0 & 4 & 5 & \infty & \infty \\ 4 & 0 & \infty & 1 & 4 \\ 5 & 2 & 0 & 3 & \infty \\ \infty & \infty & 3 & 0 & 2 \\ \infty & 4 & \infty & \infty & 0 \end{pmatrix} \quad S_{123} = \begin{pmatrix} 0 & 4 & 5 & \infty & \infty \\ 4 & 0 & 9 & 1 & 4 \\ 5 & 2 & 0 & 3 & \infty \\ \infty & \infty & 3 & 0 & 2 \\ \infty & 4 & \infty & \infty & 0 \end{pmatrix}$$

$$S_{132} = \begin{pmatrix} 0 & 4 & 5 & \infty & \infty \\ 4 & 0 & 9 & 1 & 4 \\ 5 & 2 & 0 & 3 & \infty \\ \infty & \infty & 3 & 0 & 2 \\ \infty & 4 & \infty & \infty & 0 \end{pmatrix} \quad S_{nnn} = \begin{pmatrix} 0 & 4 & 5 & 5 & 7 \\ 4 & 0 & 4 & 1 & 3 \\ 5 & 2 & 0 & 3 & 5 \\ 8 & 5 & 3 & 0 & 2 \\ 8 & 4 & 8 & 5 & 0 \end{pmatrix}$$

3.5.5.2 Zeitkorrektur der Log-Files

Bei der Analyse der Log-Daten wurde festgestellt, dass der Zeitstempel, den die Daten beim Eintreffen auf dem Server erhalten, teilweise fehlerhaft ist. Es konnte beobachtet werden, dass die Daten des Assistentencontrollers und die des ARCs oftmals stark voneinander abweichen. Dies macht sich beispielsweise dann bemerkbar, wenn der Assistentencontroller einen ARC in einer Weiche meldet, dieser sich jedoch laut seinen eigenen Angaben noch nicht an dieser Position befindet. Grund dafür sind die verschiedenen Kommunikationswege, über die die Logs zum Server gelangen. Während die Dienste (z.B. Assistentencontroller, Pfadmanagement etc.) direkt auf dem Server betrieben werden, erfolgt die Log-Übertragung der ARCs mittels WLAN. Dabei kommt es netzwerkbedingt zu Latenzzeiten oder kurzen Unterbrüchen, was eine Abweichung des Zeitstempels von bis zu einer Minute nach sich zieht. Somit kann dieser fehlerhafte Zeitwert der ARC-Logs nicht direkt für weitere Auswertungen herangezogen werden.

Zusätzlich zum Zeitstempel schickt der ARC bei jeder Log-Zeile auch seine aktuelle Laufzeit in Millisekunden mit. Dabei handelt es sich um eine Integervariable welche folglich nach ca. 65 Sekunden überläuft und wieder bei null startet. Die Laufzeit stellt für kurze Zeitbereiche < 65 Sekunden einen genauen Relativwert dar, kann aber für die Interpretation größerer Zeitspannen nicht herangezogen werden. Dieser Relativwert und der fehlerhafte Absolutwert können allerdings dazu verwendet werden, einen brauchbaren Schätzwert zu berechnen.

3.5.5.2.1 Zeitkorrektur mittels gewichtetem Mittelwert

Wenn man über mehrere (in unserem Fall zwei) verschiedene Messgrößen verfügt, die jeweils eine unterschiedliche Genauigkeit aufweisen, können diese durch einen gewichteten Mittelwert kombiniert werden. Jeder Messwert erhält entsprechend seiner Genauigkeit eine Gewichtung (w_i) (vgl. [MS09], S 45).

$$\bar{x}_w = \sum_{i=1}^n w_i \cdot x_i, \text{ mit } \sum_{i=1}^n w_i = 1$$

Bei unserer Anwendung errechnet sich der Schätzwert zum Zeitpunkt k (t_k) mithilfe der Gewichte (w_{rel}, w_{abs}), des Zeitstempels ($t_{k,abs}$) und dem Zeitwert ($t_{k,rel}$), welcher sich aus dem alten Schätzwert (t_{k-1}) und der Differenz der Relativzeitwerte (r_k) zusammensetzt. Als Startwert (t_0) wird der erste Zeitstempel verwendet.

$$t_k = w_{rel} \cdot t_{k,rel} + w_{abs} \cdot t_{k,abs}$$

$$t_{k,rel} = t_{k-1} + (r_k - r_{k-1})$$

$$t_0 = t_{0,abs}$$

Wie in Abbildung 38 zu sehen ist, strebt die Abweichung vom Schätzwert zur „realen“ Zeit in Richtung mittlerer Abweichung des Zeitstempels. Wie stark der Wert mit dem Zeitstempel mitschwingt bzw. wie lang er braucht um einen eingeschwungenen Zustand zu erreichen, ist abhängig von den Gewichtungsfaktoren.

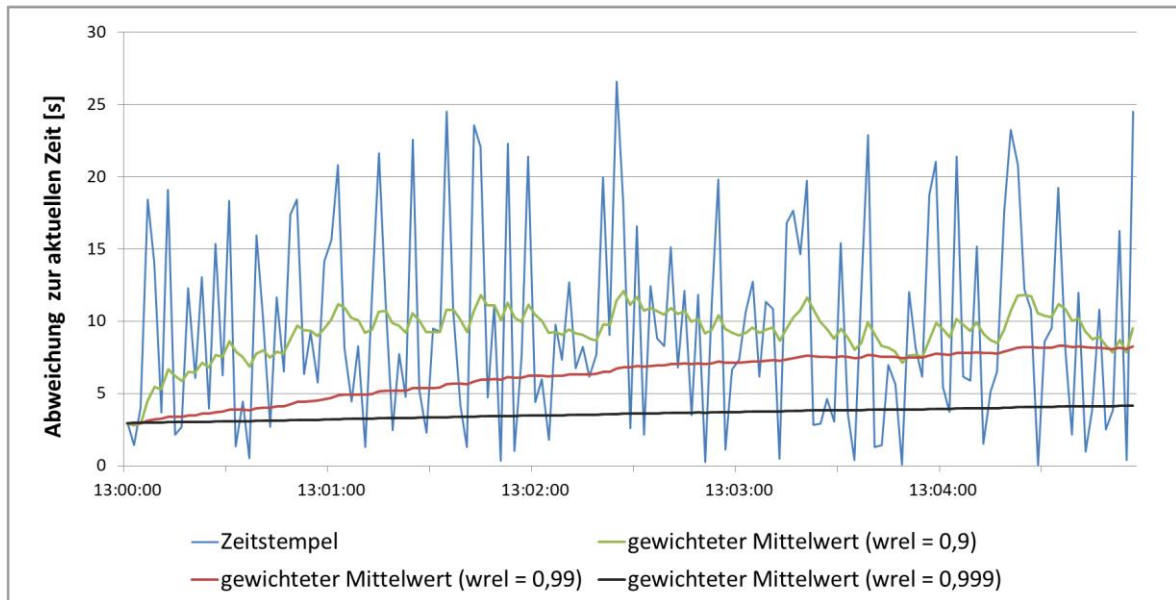


Abbildung 38: Schätzwert abhängig von der Gewichtung w_{rel}

3.5.5.2 Zeitkorrektur mittels Mindestwert

Die Abweichung zwischen Zeitstempel und der „realen“ Zeit kann systembedingt nur positive Werte annehmen – sprich diese sind im Idealfall gleich. Somit wird bei dieser Methode für den Schätzwert (t_k) nur dann der Zeitstempel ($t_{k,abs}$) verwendet, wenn dieser kleiner als der Wert aus den Relativzeiten ($t_{k,rel}$) ist.

$$t_k = \min(t_{k,rel}, t_{k,abs})$$

$$t_{k,rel} = t_{k-1} + (r_k - r_{k-1})$$

$$t_0 = t_{0,abs}$$

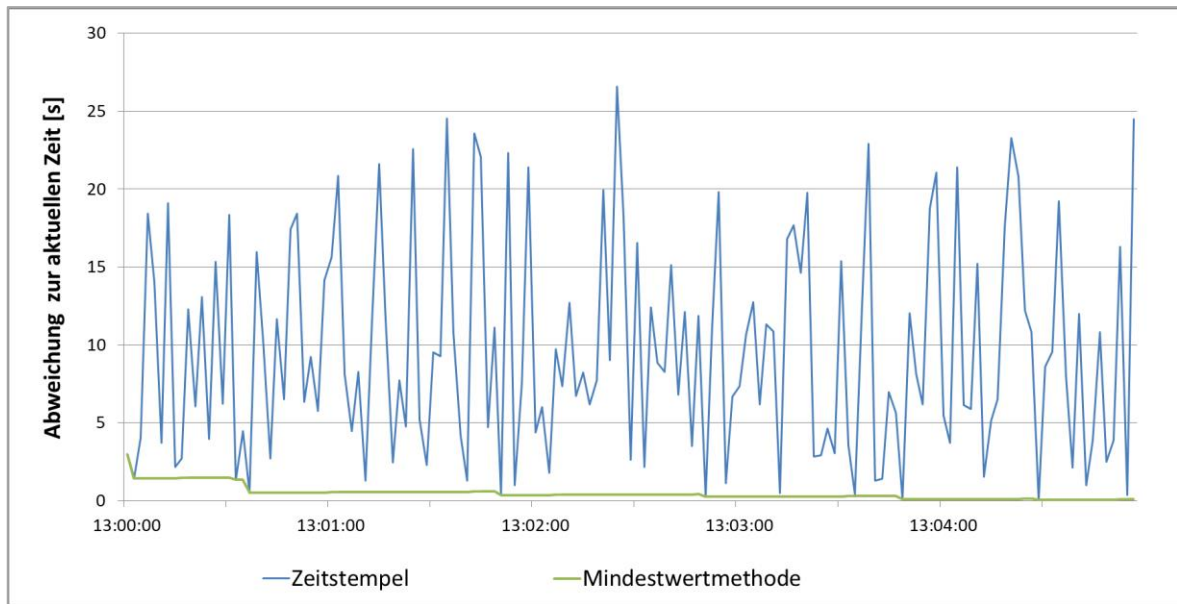


Abbildung 39: Schätzwert der Mindestwertmethode

3.5.5.2.3 Vergleich der Methoden

Für die spätere Auswertung ist es wichtig, dass die Zeitwerte aus den verschiedenen Datenquellen (z.B. Fahraufträge, ARC-Positionen etc.) und der Komponenten untereinander übereinstimmt. Bei der Korrektur mit dem gewichteten Mittel pendelt sich die Abweichung des Schätzwertes bei mehreren Sekunden ein, was für unsere Anwendung einen zu großen Fehler darstellt.

Aus diesem Grund wird die Mindestwertmethode angewendet, da diese nach einer kurzen Einlaufzeit brauchbare Ergebnisse liefert.

3.5.5.3 ARC-Stillstand

Das Auffinden von blockierten ARCs wird verwendet, um die Verfügbarkeit des Systems zu bestimmen. Dafür werden die ARC-Positionsdaten und die Anlagenstörungen benötigt. In Abbildung 40 ist der Ablauf zur Auffindung von blockierten ARCs dargestellt.

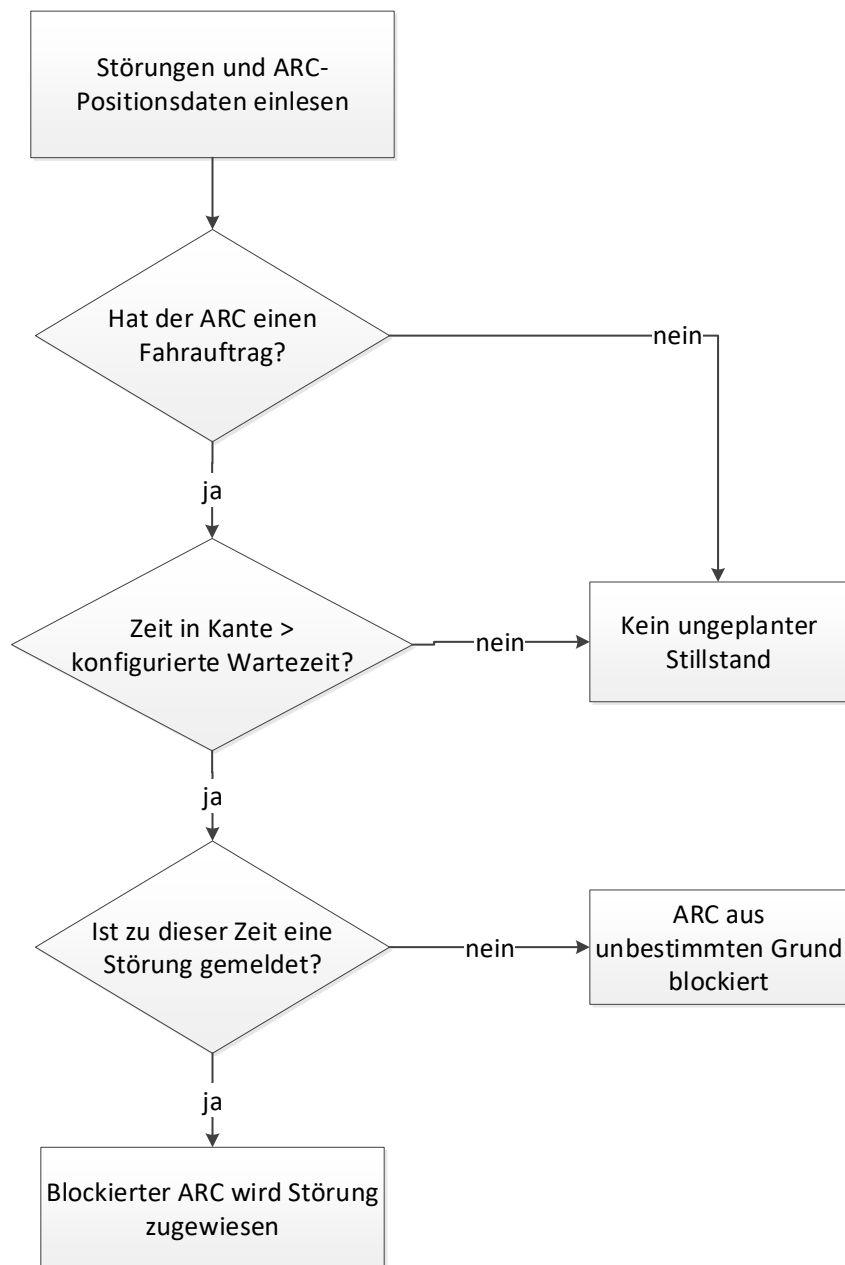


Abbildung 40: Ablaufdiagramm zur Auffindung von blockierten ARC's

3.6 Darstellung

Zur Darstellung der Ergebnisse werden sogenannte Dashboards verwendet. Dashboard kommt aus dem Englischen und bedeutet so viel wie Armaturenbrett. In unserem Fall wird darunter eine grafische Oberfläche verstanden, die verschiedenste Informationen enthält.

Da die Auswertungen einer Vielzahl unterschiedlicher Benutzer zugänglich gemacht werden sollen und um das Potenzial der vorhandenen Daten voll auszuschöpfen, werden die Dashboards in Überblicks- und Detail-Dashboards aufgeteilt. Die Überblicks-Dashboards sollen dem Benutzer eine schnelle Übersicht

über den Status der Anlage und die wichtigsten KPIs (Key Performance Indicator) geben. Darunter fallen:

- **Materialflussdashboard:** Das Materialflussdashboard ist das wichtigste der zwei Überblicks-Dashboards, da die Hauptaufgabe des Systems die Beförderung von Transporteinheiten ist. Kern dieses Dashboards ist ein Diagramm mit den beförderten TE pro Stunde im Verlauf der Zeit, verglichen mit dem verkauften Soll-Wert (rote Linie). Auf dieser Seite sind auch die Doppelspielrate und die Aufgabenverteilung der ARCs zu finden, um die Auslastung des Systems zu bewerten.

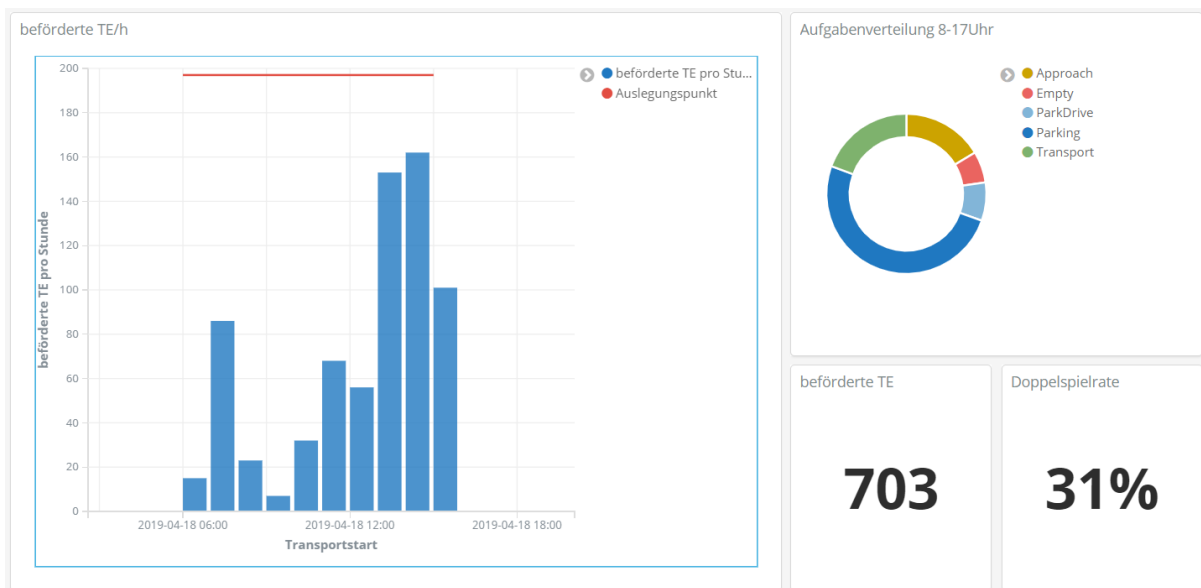


Abbildung 41: Materialflussdashboard

- **Verfügbarkeitsdashboard:** Dieses Dashboard gibt einen schnellen Überblick über die Verfügbarkeit. Einmal als Gesamtwert für den angegebenen Zeitbereich und einmal für jeden Tag (oder anderen Zeitbereich) in einem Balkendiagramm dargestellt (vgl. [VDI92], S. 13) Neben der Verfügbarkeit wird auch die die Anzahl der Störungen nach Komponenten und Fehlerart aufgeteilt. So können Häufungen erkannt werden.

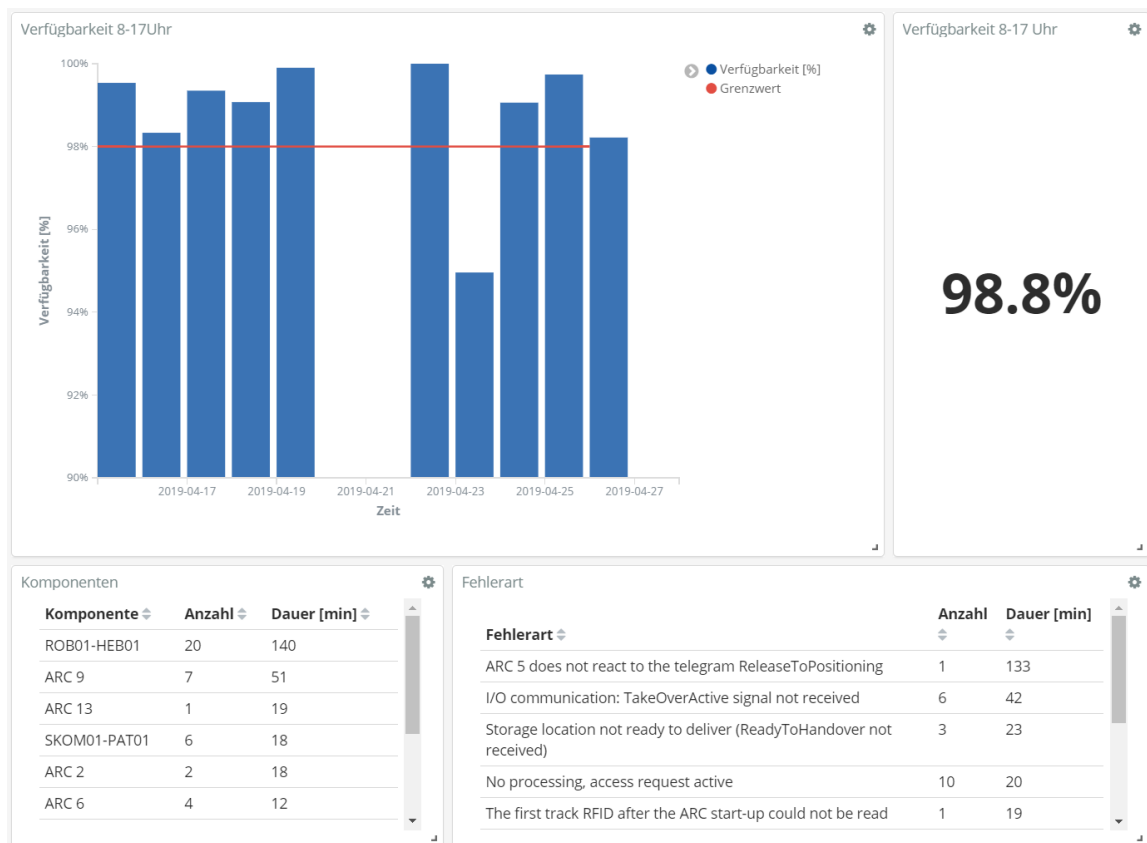


Abbildung 42: Verfügbarkeitsdashboard

Wenn ein Benutzer nicht nur an einem kurzen Reporting der Anlage interessiert ist, sondern genauere Informationen benötigt, um beispielsweise Fehler zu suchen, Optimierungen vorzunehmen oder Bottlenecks zu finden, stehen ihm folgende Detaildashboards zur Verfügung:

- **Assistentendashboard:** Für die Assistenten gibt es zum einen ein Übersichtsblatt, auf dem die Auslastung und die Leistung aller Assistenten dargestellt ist (Abbildung 43). Für genauere Auswertungen können von jedem einzelnen Assistenten die Auslastung im Tagesverlauf, die Systemzeiten, die Anzahl der beförderten ARCs und die Anzahl der Bewegungen ermittelt werden (Abbildung 44).

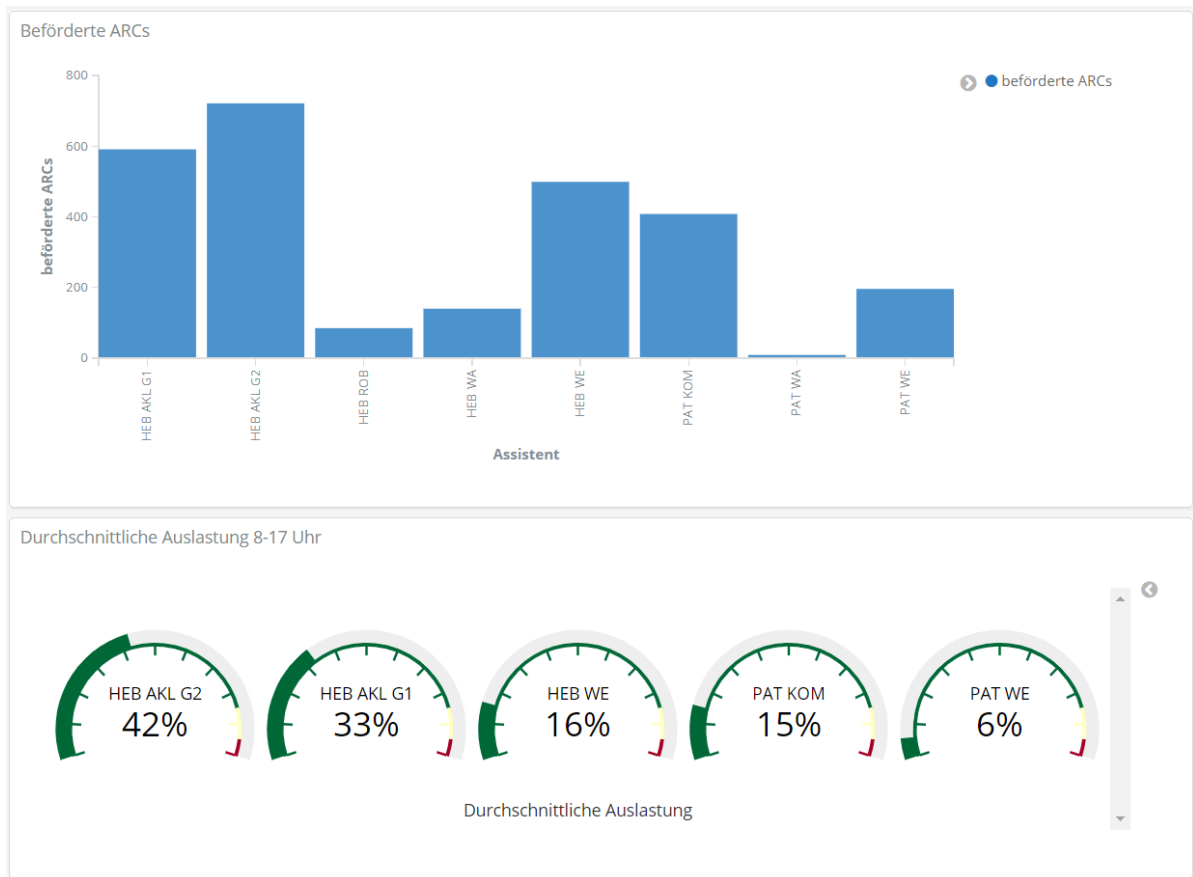


Abbildung 43: Assistenten Übersichtsdashboard

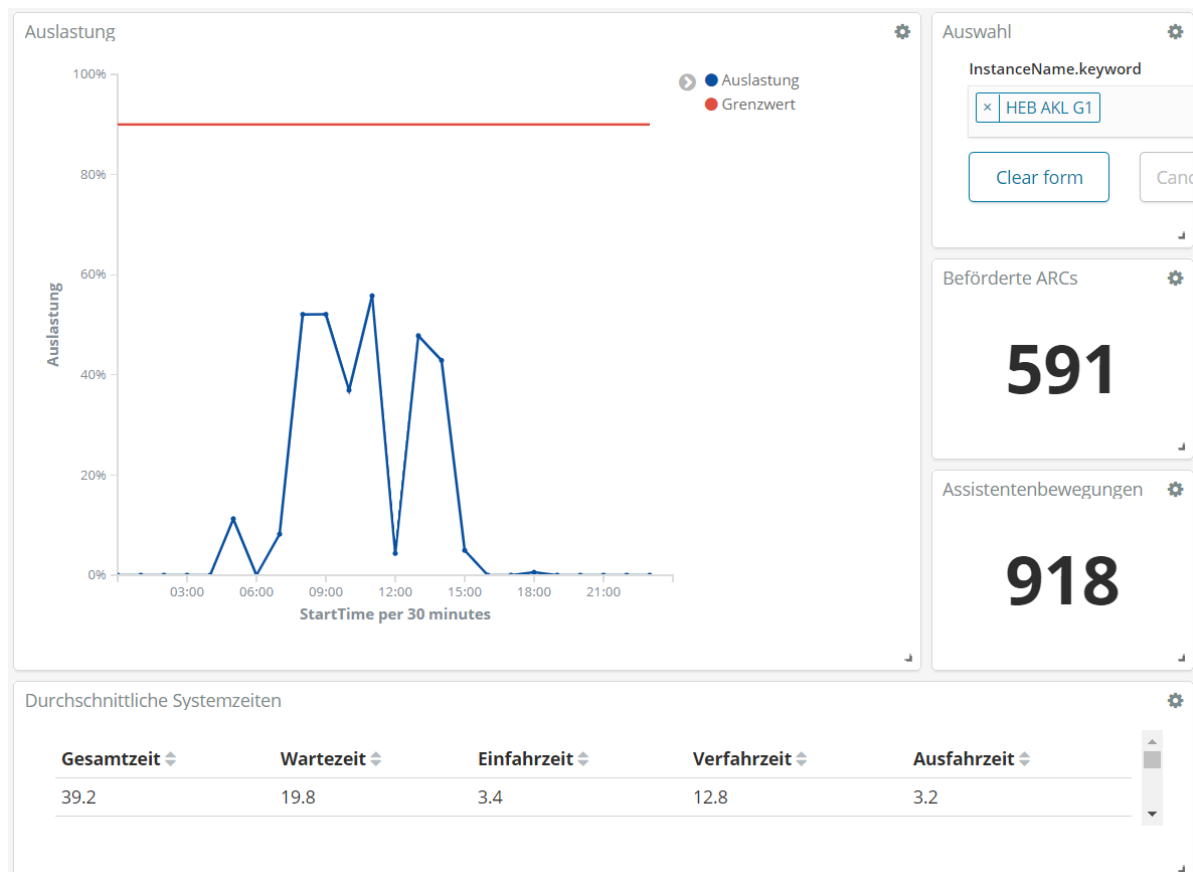


Abbildung 44: Assistenten Detaildashboard

- **ARC Dashboard:** Für die ARCs gibt es wieder ein Übersichtsdashboard, auf dem die Aufgabenverteilung, die häufigsten Fehler, die Betriebsstunden und die gefahrenen Kilometer angezeigt werden können. Durch Anwendung von Filtern ist eine Detailansicht für einzelne ARCs möglich.

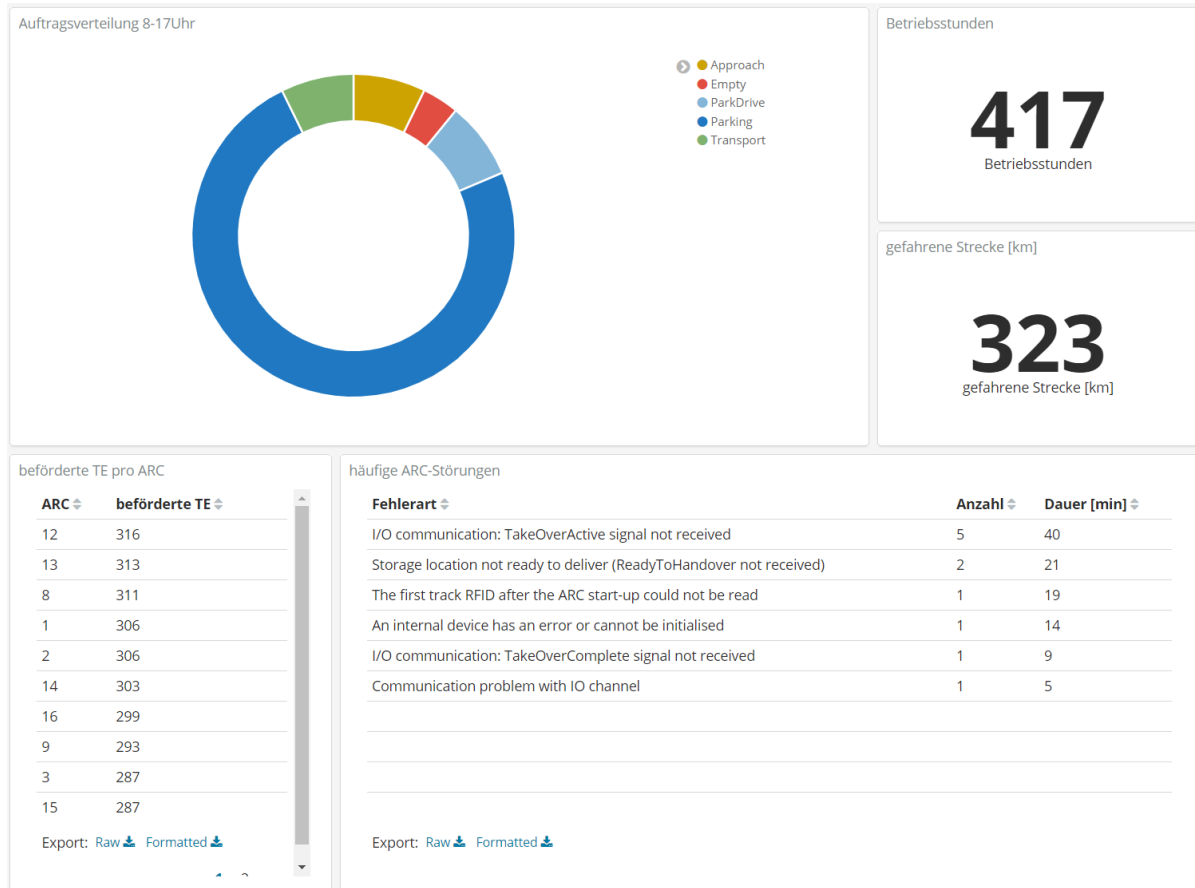


Abbildung 45: ARC Übersichtsdashboard

- **Detailmaterialfluss:** Beim Materialfluss Dashboard werden die wichtigsten Leistungsdaten übersichtsmäßig zusammengefasst. Für den Detailmaterialfluss gibt es hingegen drei Detailansichten. In Abbildung 46 werden die Ein-/Auslagerungen und die Doppelspiele für jede einzelne Station darstellt. Im nächsten Dashboard (Abbildung 47) wird der Materialfluss durch eine Transportmatrix in die unterschiedlichen Fahrten aufgeteilt. Mit den Informationen aus der letzten Detailansicht (Abbildung 48) können die Fahrzeiten zwischen den Bereichen genauer analysiert werden.

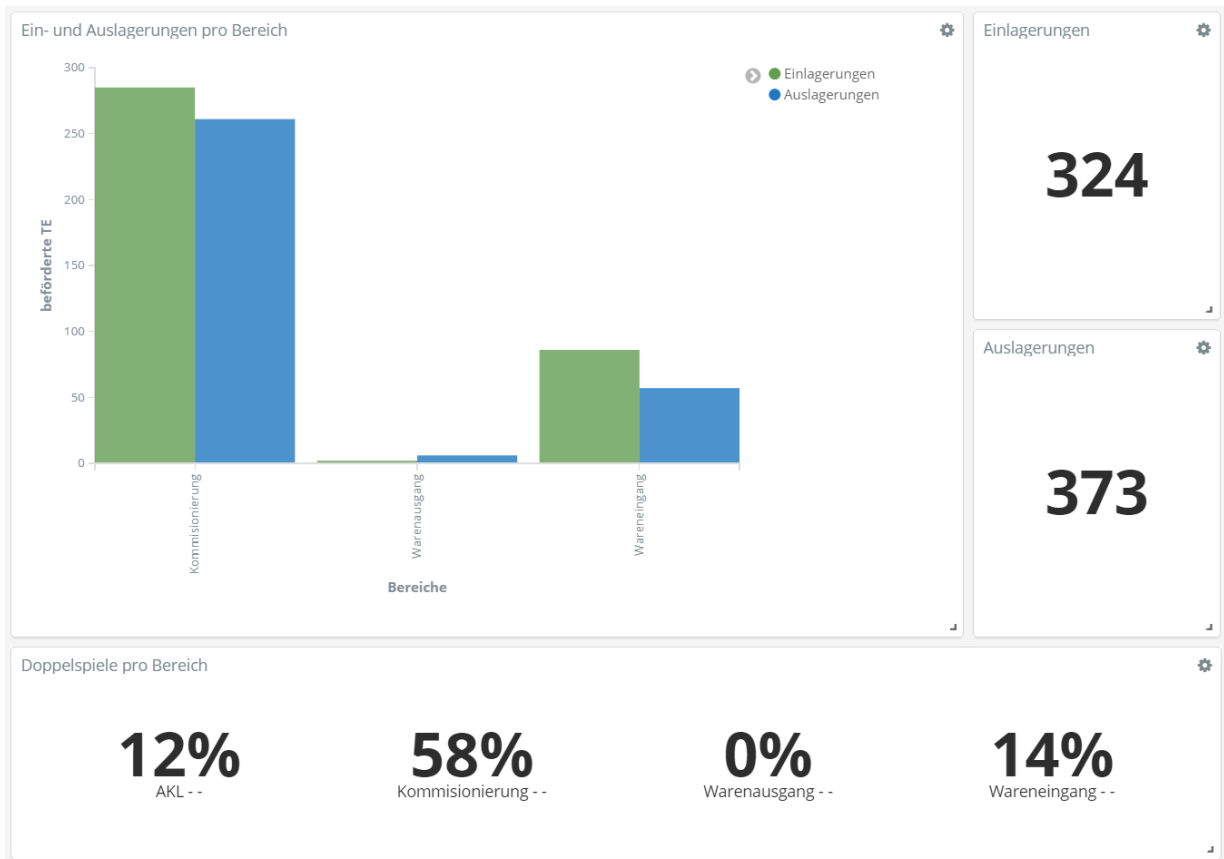


Abbildung 46: Detaillierter Materialfluss mit Stationsübersicht

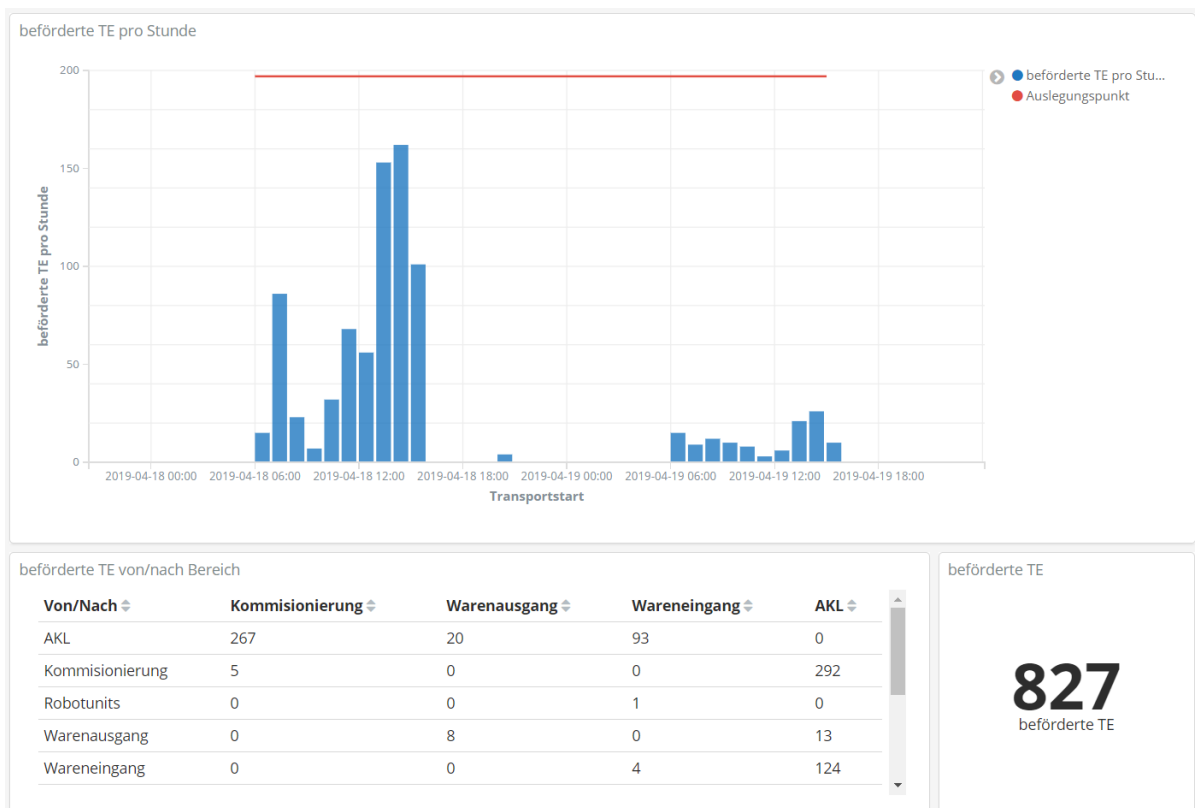


Abbildung 47: Detaillierter Materialfluss mit Transportmatrix

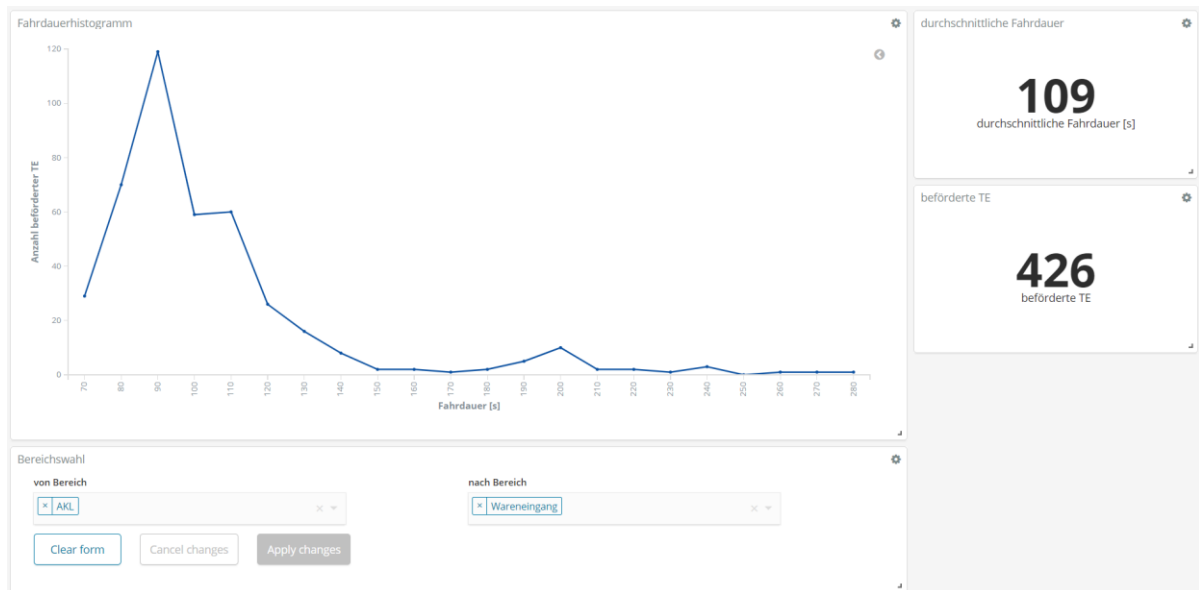


Abbildung 48: Fahrtdauer Dashboard

- Störungsanalyse:** Dieses Dashboard dient der genaueren Analyse von Betriebsstörungen. Dafür werden ein Tagesverlauf mit dem Anlagenausfall und eine Liste der aufgetretenen Fehler zur Verfügung gestellt.

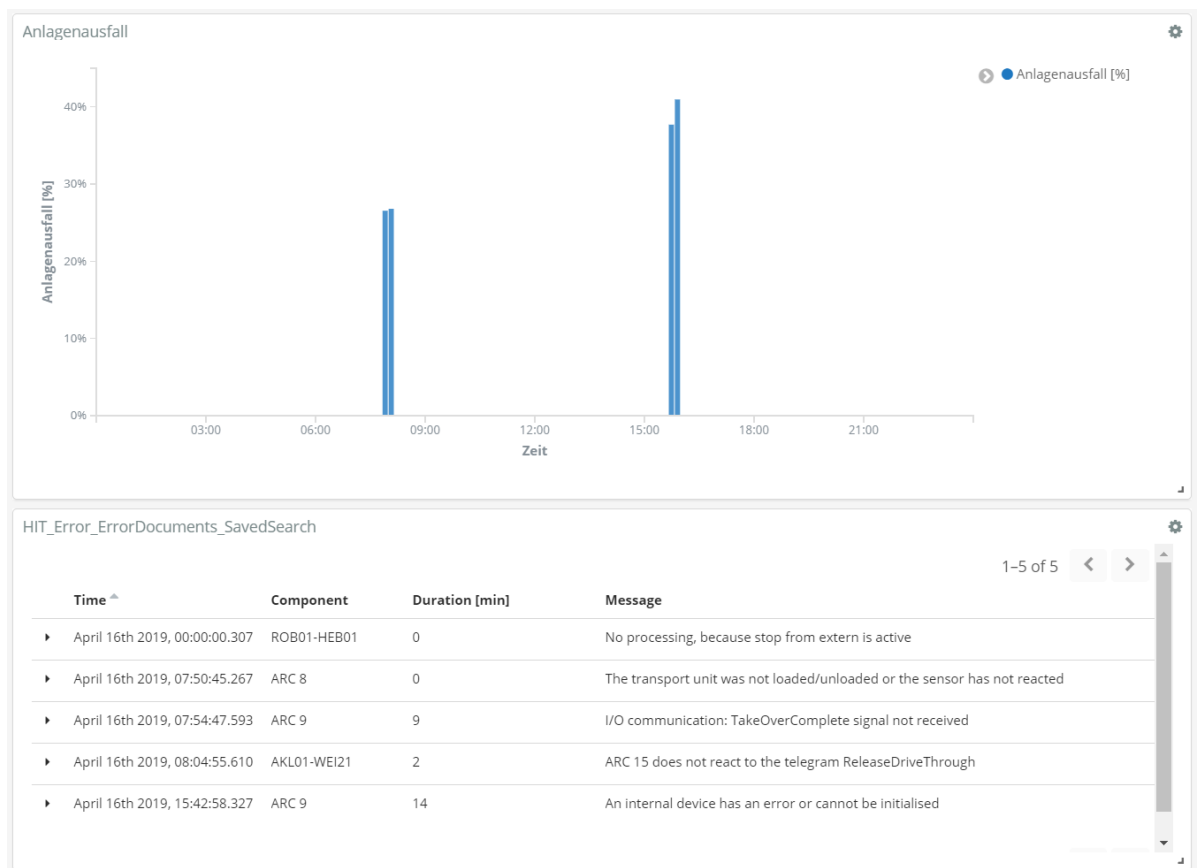


Abbildung 49: Störungsanalyse Dashboard

3.6.1 Kibana

Zur Darstellung der Ergebnisse wird ein Tool namens Kibana verwendet. Dabei handelt sich um eine browserbasierte Analyseplattform, die auf Elasticsearch aufbaut. Mit ihr können die Daten aus den Elasticsearch-Indizes durchsucht, visualisiert und analysiert werden.

Die drei Hauptfunktionen der Software sind:

- **Discover**
In diesem Bereich können durch Anwendung von Filtern die einzelnen Dokumente der Indizes durchsucht werden. In Abbildung 50 wurde beispielweise der „arc_position“ Index nach „PlantName“ = „HIF“ gefiltert. Im rechten oberen Rand kann der gewünschte Zeitbereich ausgewählt werden.

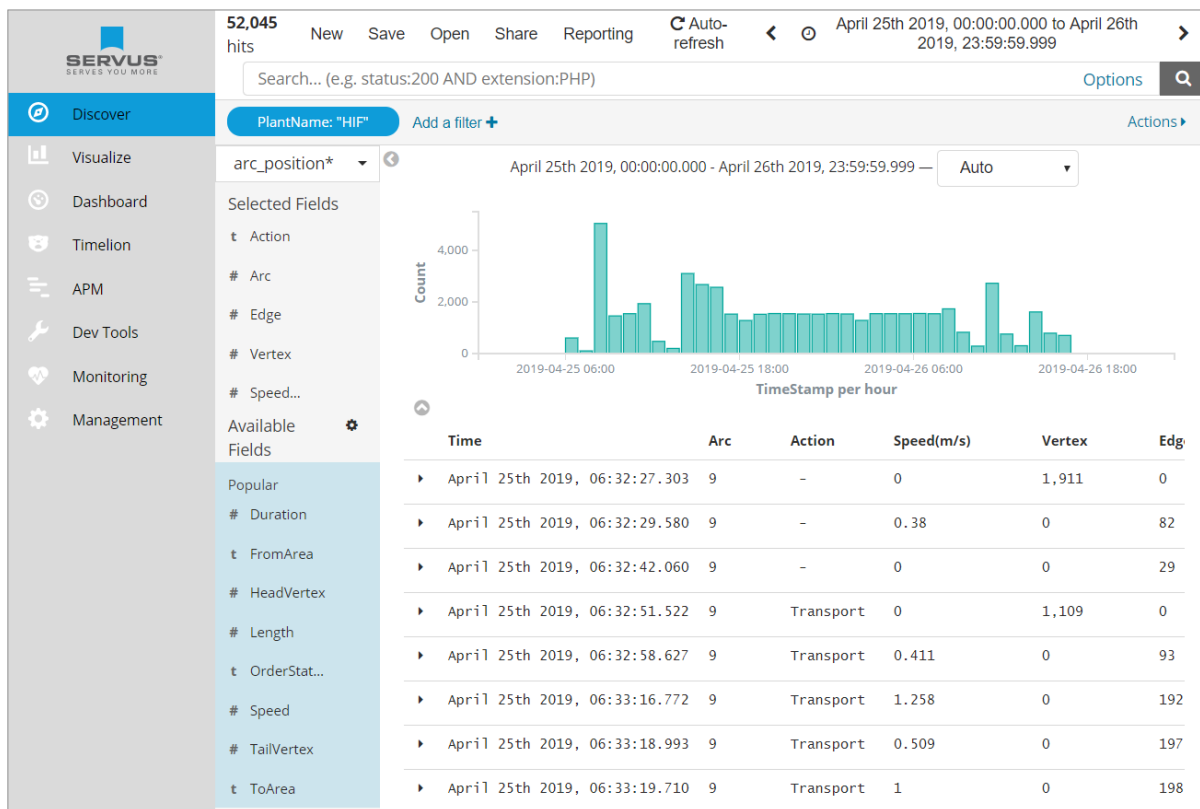


Abbildung 50: Kibana Discover

- **Visualize**
In diesem Reiter können neue Datenvisualisierungen erstellt und gespeichert werden. Kibana bietet hier eine Vielzahl an Möglichkeiten, wie beispielsweise Liniendiagramme, Kreisdiagramme, Balkendiagramme, Pivot-Tabellen, Heat-Maps etc.
- **Dashboard**
Dieser Bereich dient der Visualisierung und Erstellung von Dashboards.

Wie bei den anderen Reitern kann auch hier die Filterung nach verschiedenen Eigenschaften sowie die Auswahl eines gewünschten Zeitbereichs vorgenommen werden.

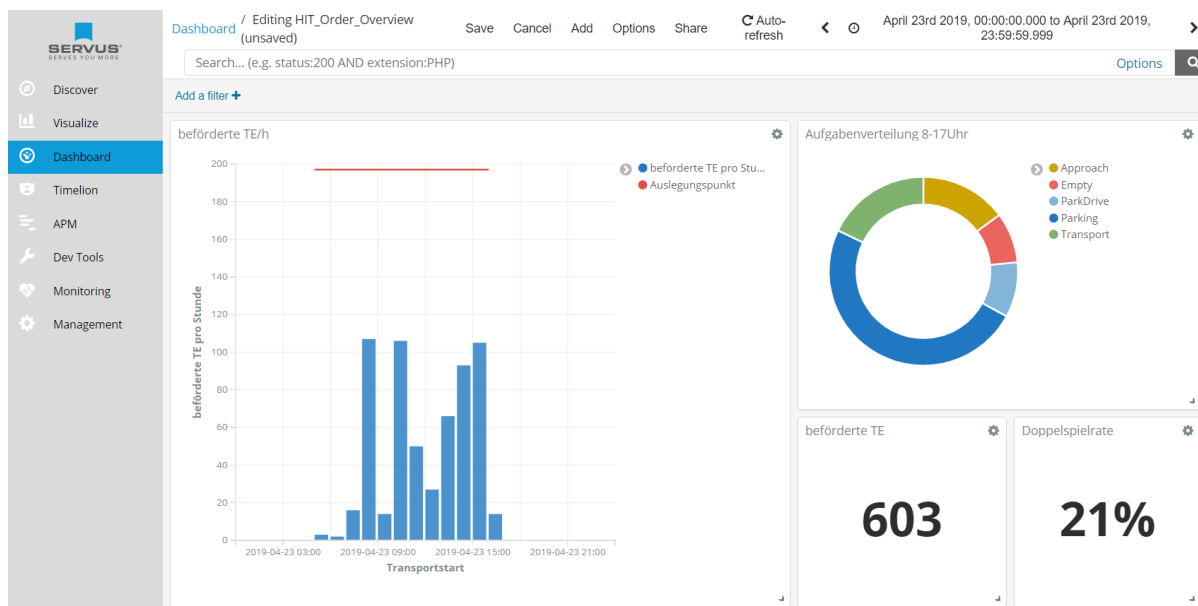


Abbildung 51: Kibana Dashboard

4 Implementierung

Für die Durchführung der Auswertungen wurde ein Programm erstellt, welches auf dem Analyseserver liegt und in regelmäßigen Abständen automatisch ausgeführt wird. Da SEAS in C# implementiert ist, fiel die Wahl auf diese Programmiersprache. Als Entwicklungsumgebung wurde Visual-Studio verwendet.

Da bereits das benutzerfreundliche Visualisierungstool Kibana auf dem Analyseserver installiert ist, wird dieses für die Darstellung der Daten verwendet. Um mit Kibana auf die ausgewerteten Daten zugreifen zu können, müssen diese nach dem Auswerten in der Elasticsearch Datenbank abgespeichert werden.

Für die Kommunikation mit Elasticsearch (suchen, auslesen und schreiben) wurde ein von Elasticsearch zur Verfügung gestellter Client (NEST-Client) verwendet.

4.1 Softwareaufbau

Das Programm besteht aus Klassen, welche die Daten enthalten (Data Model) und aus Klassen, die der Datenverarbeitung (Processing) dienen (Abbildung 52). Beim Start des Programms werden zuerst die Konfigurationsdaten und dann die Ausgangsdaten von der Elasticsearch-Datenbank ausgelesen. Nach der Aufbereitung in den „InputReader“ Klassen werden die Daten analysiert und mit den entsprechenden „ElasticOutput“ Klassen auf der Elasticsearch-Datenbank gespeichert, wo sie dann mit Kibana visualisiert werden können.

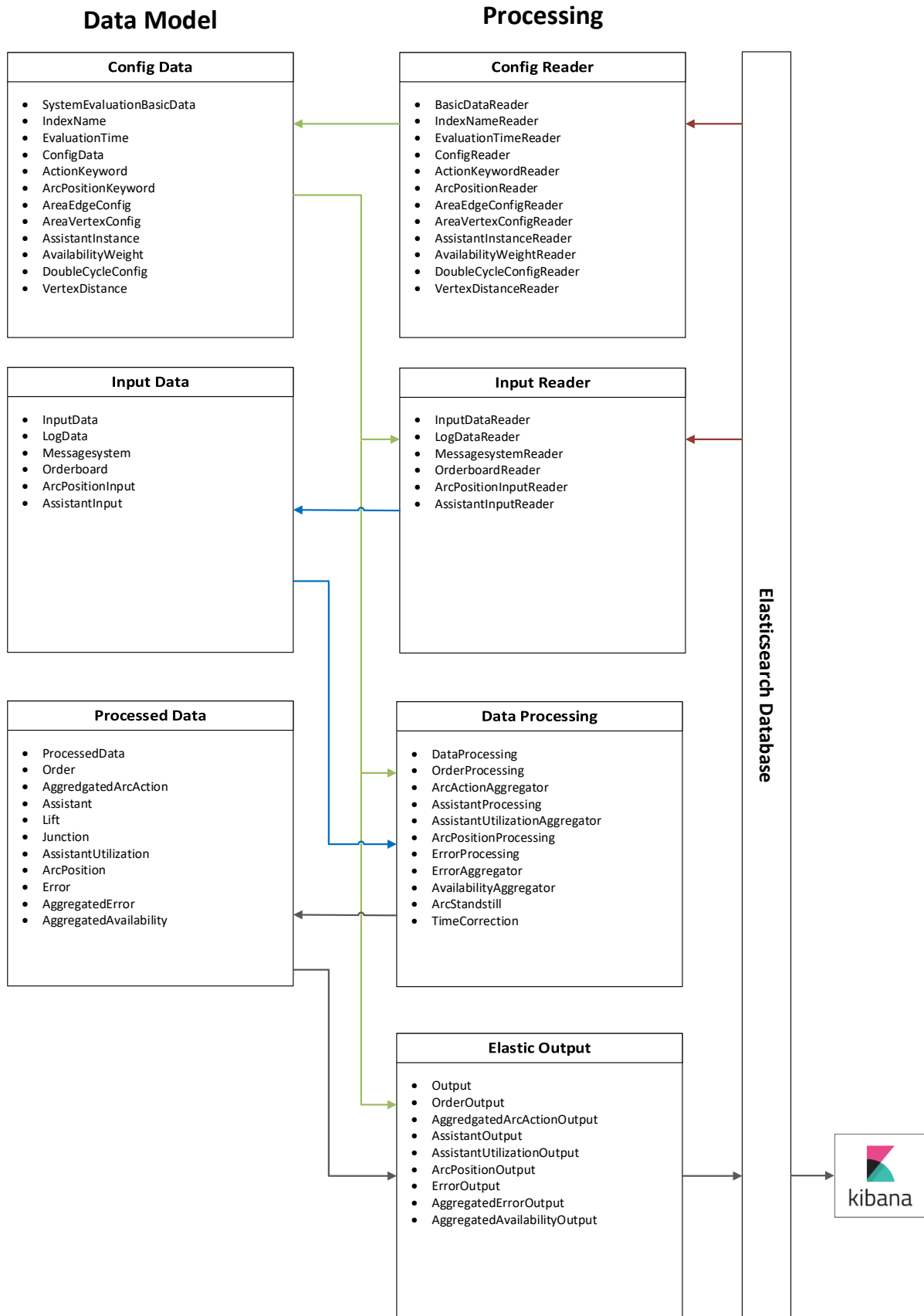


Abbildung 52: Softwareaufbau

4.1.1 Konfigurationsdaten

Beim Ausführen des Programms werden zuerst die Basisdaten ausgelesen. Das sind zum einen die Namen der Indices, wo die restlichen Konfigurations- und Ausgangsdaten gesucht werden und wohin dann die ausgewerteten Daten gespeichert werden. Weiters werden die Zeitpunkte ausgelesen, an denen die Anlagen das letzte Mal ausgewertet wurden.

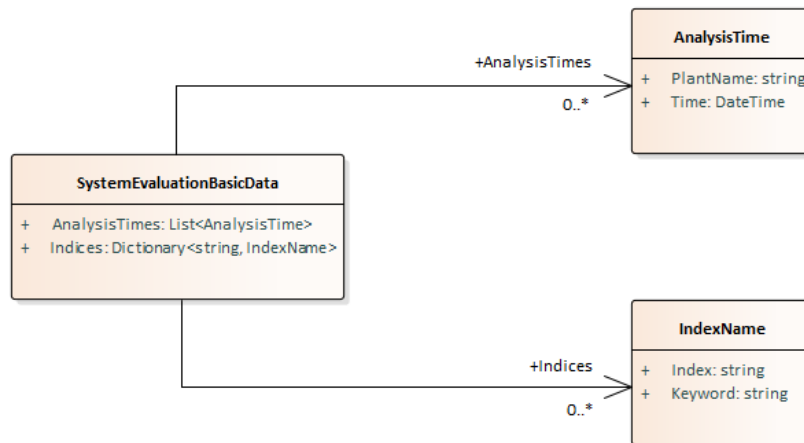


Abbildung 53: Basisdaten für die Anlagenauswertung

Ab diesem Zeitpunkt startet das Programm mit den neuen Auswertungen. Dabei werden zuerst die anlagenspezifischen Konfigurationsdaten erfasst, wie beispielsweise die Definition der Doppelspielbereiche, die Verfügbarkeitsgewichtungen, die Streckenmatrix, die Eigenschaften der einzelnen Assistenten, etc. (Abbildung 54).

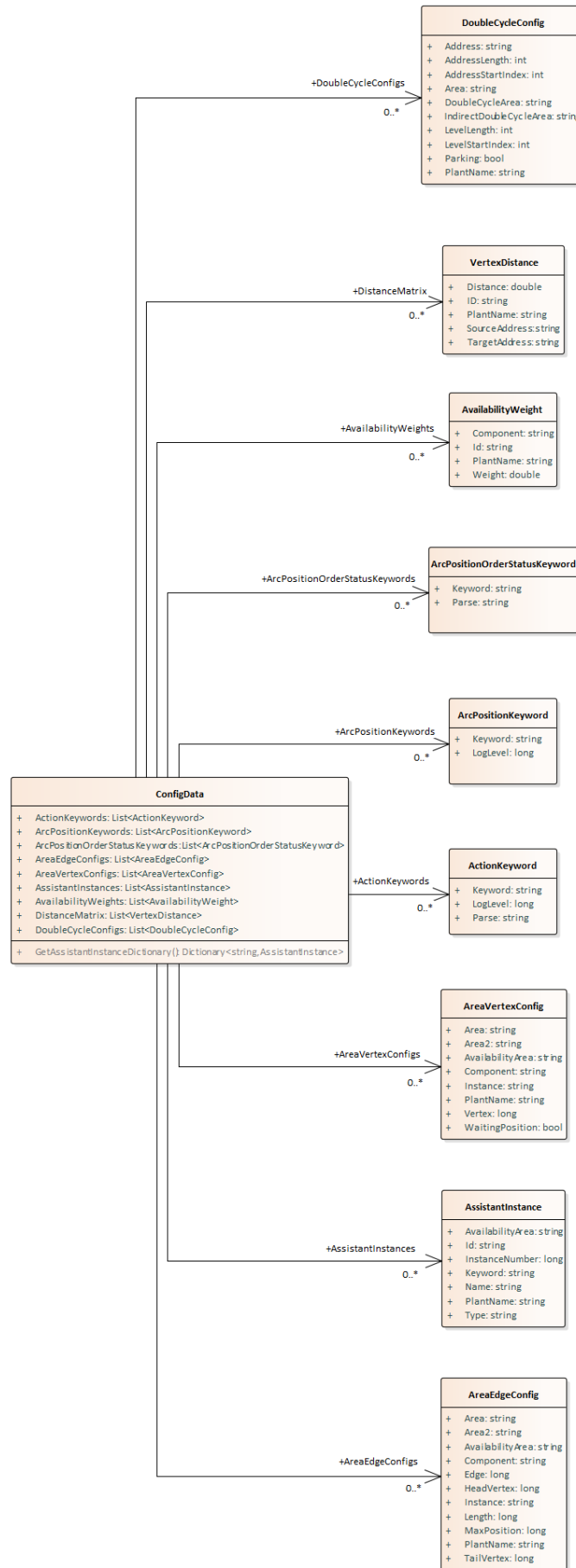


Abbildung 54: Anlagenspezifische Konfigurationsdaten für die Anlagenauswertung

4.1.2 Inputdaten

Nach dem Auslesen der Konfigurationsdaten werden mithilfe des Elasticsearch Nest-Clients die Ausgangsdaten durchsucht, direkt aufbereitet und dann in den Inputdaten abgelegt.

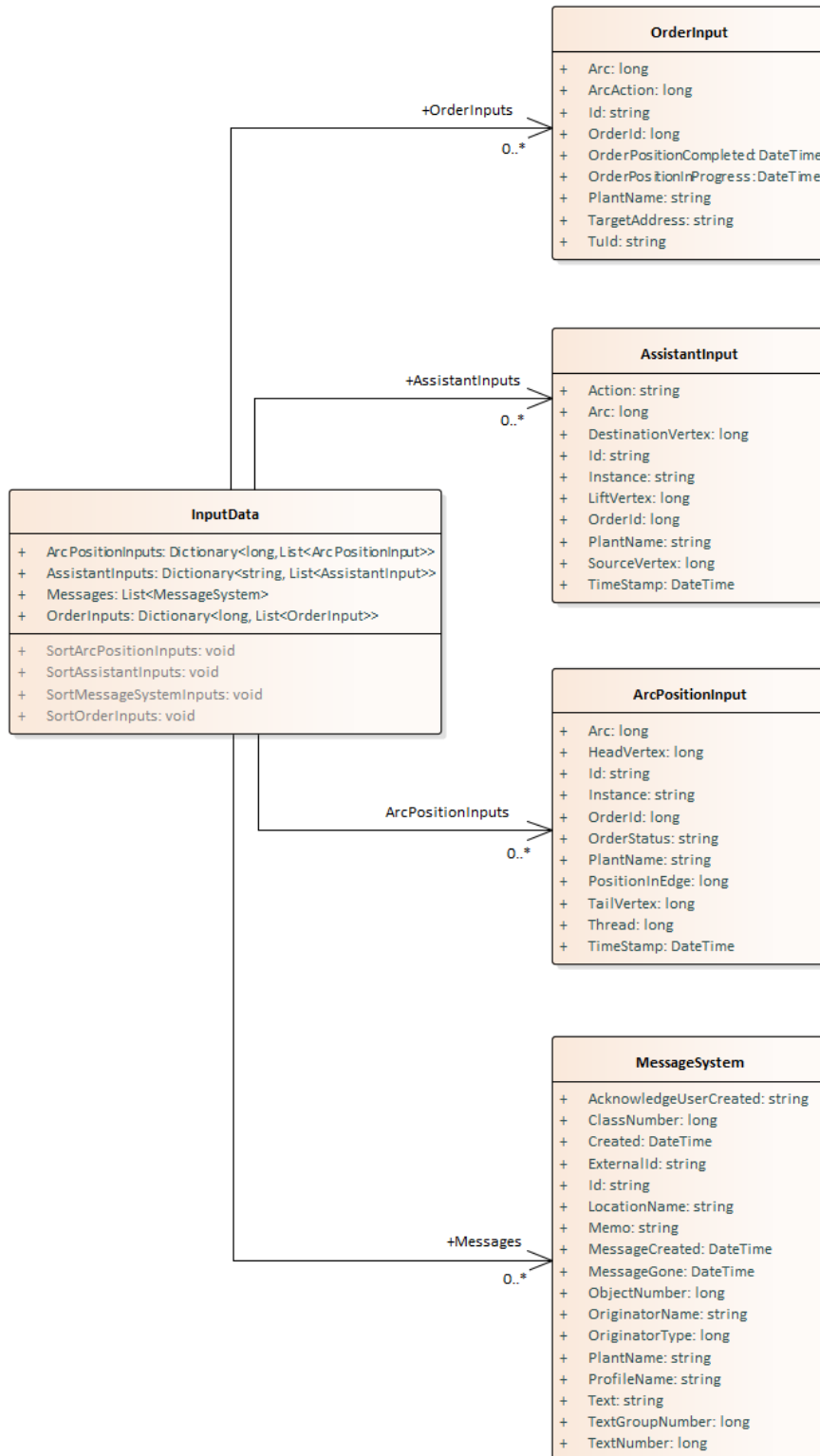


Abbildung 55: Inputdaten für die Anlagenauswertung

4.1.3 Ausgewertete Daten

Mit den entsprechenden Methoden aus Kapitel 3.5 werden die aufbereiteten Inputdaten analysiert und in den „ProcessedData“ Klassen gespeichert.

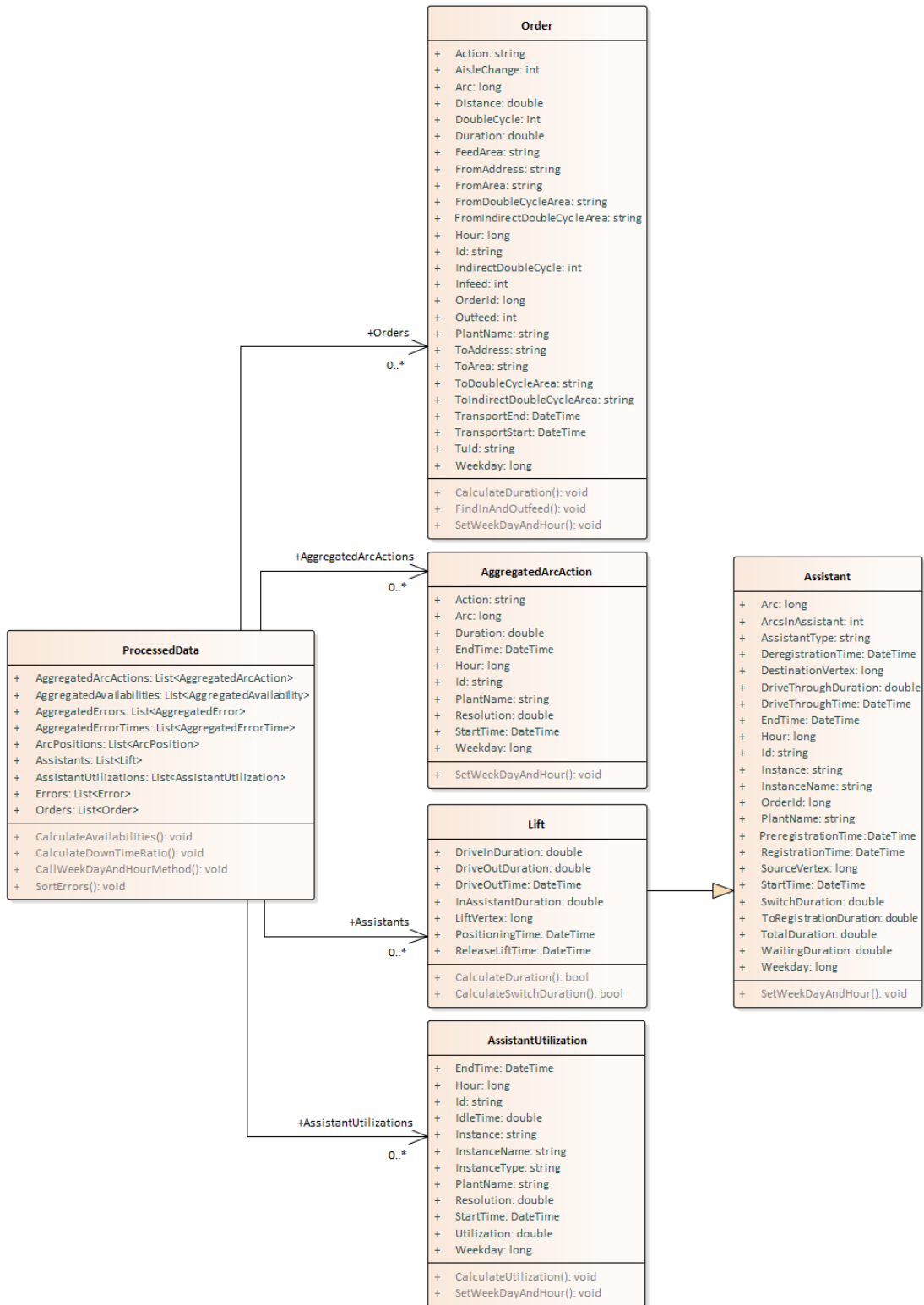


Abbildung 56: Ausgewertete Daten (Fahraufträge, ARC-Aufgaben, Assistentenbewegungen, Assistentenauslastung)

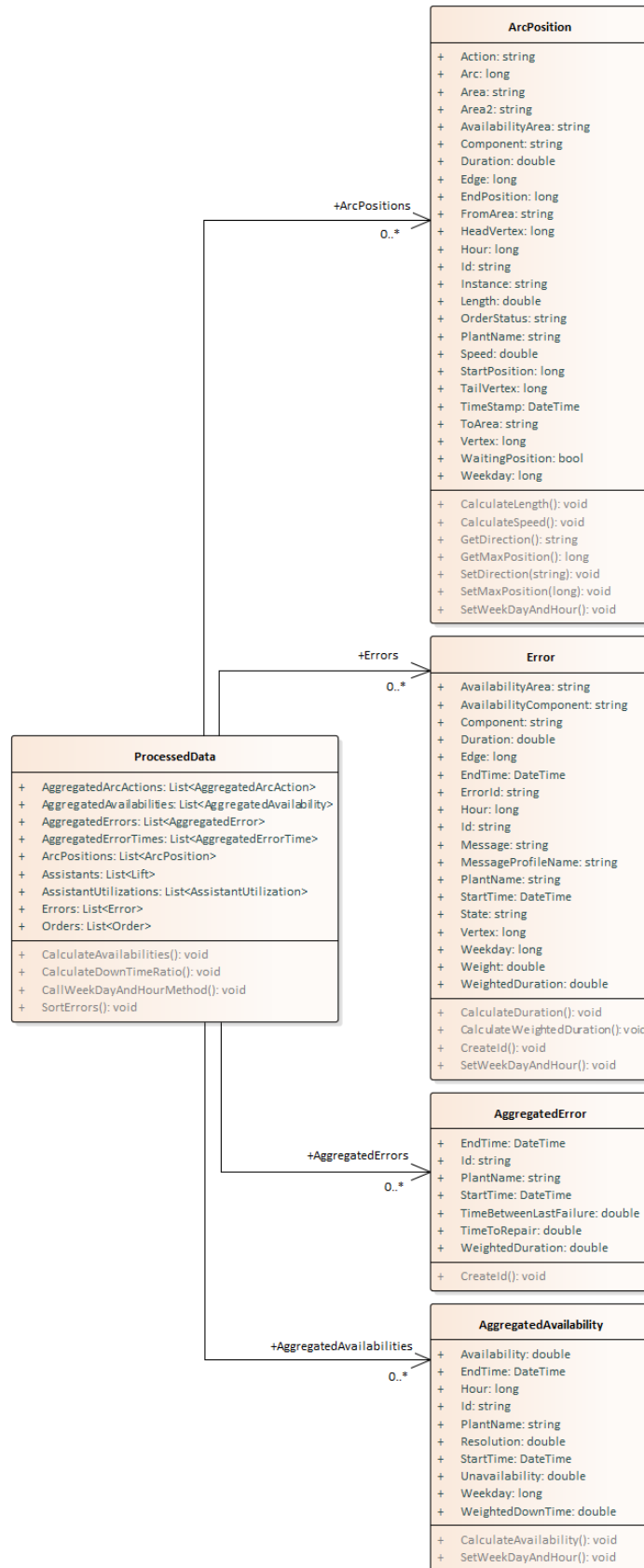


Abbildung 57: Ausgewertete Daten (ARC-Position, Störungen, Verfügbarkeit)

4.2 Verifikation/Validierung

4.2.1 Verifikation

Unter Verifikation versteht man laut ISO 9000 „*die Bestätigung durch einen objektiven Nachweis, dass Anforderungen erfüllt werden* (vgl. [ISO05], S. 21)“. Bei unserer Anwendung wird für die Verifikation überprüft, ob die ausgewerteten Daten aus den Ausgangsdaten richtig aufbereitet und verarbeitet wurden.

Die Verifikation erfolgte in diesem Fall auf zwei verschiedene Arten.

Zum einen wurden im Betrieb der hausinternen Anlage bei Servus Intralogistics die ARCs beobachtet, die verschiedenen Zeiten notiert und dann mit den Ausgangsdaten und den ausgewerteten Daten verglichen. Somit konnten nicht nur die verarbeiteten Daten, sondern auch die Ausgangsdaten verifiziert werden. Für die Verifikation der Verfügbarkeit und Störungsanalyse wurde überprüft, ob Störungen auf der internen Anlage korrekt aufgenommen und dargestellt wurden.

Zum anderen wurden laufend händisch ausgewertete Daten mit den maschinell erhobenen Daten verglichen.

4.2.2 Validierung

Unter Validierung verstehen man laut ISO 9000 „*die Bestätigung durch objektiven Nachweis erbracht, dass die Anforderungen für eine bestimmte Anwendung oder einen bestimmten Gebrauch erfüllt sind* (vgl. [ISO05], S. 22).“ Für unseren Fall bedeutet Validierung, dass die ermittelten Daten, Kennzahlen und Darstellungen den Mitarbeitern und Kunden ermöglichen, die Leistung und Verfügbarkeit der Anlage zu bewerten und zu analysieren. Hierfür wurden Auswertungen von externen Anlagen angefertigt und den Mitarbeitern bzw. Kunden zur Verfügung gestellt und deren Feedback bei der Implementierung berücksichtigt. Beispiele für diese Auswertungen sind in Kapitel 5 zu finden.

5 Ergebnisse

In diesem Abschnitt wird das Potenzial der erstellten Auswertungen anhand realer Auswertungsbeispiele aufgezeigt.

5.1 Fall 1: Doppelspielproblem

Bei einer kürzlich in Betrieb genommenen Anlage galt es die Materialflussdaten auszuwerten. Dabei wurde unter anderem die Doppelspielrate überprüft, wobei sich herausstellte, dass sich diese seit der Inbetriebnahme bei niedrigen Werten eingependelt hatte. In Abbildung 58 sind die durchschnittlichen, stündlichen Doppelspielraten der Gesamtanlage von drei unterschiedlichen Tagen dargestellt.

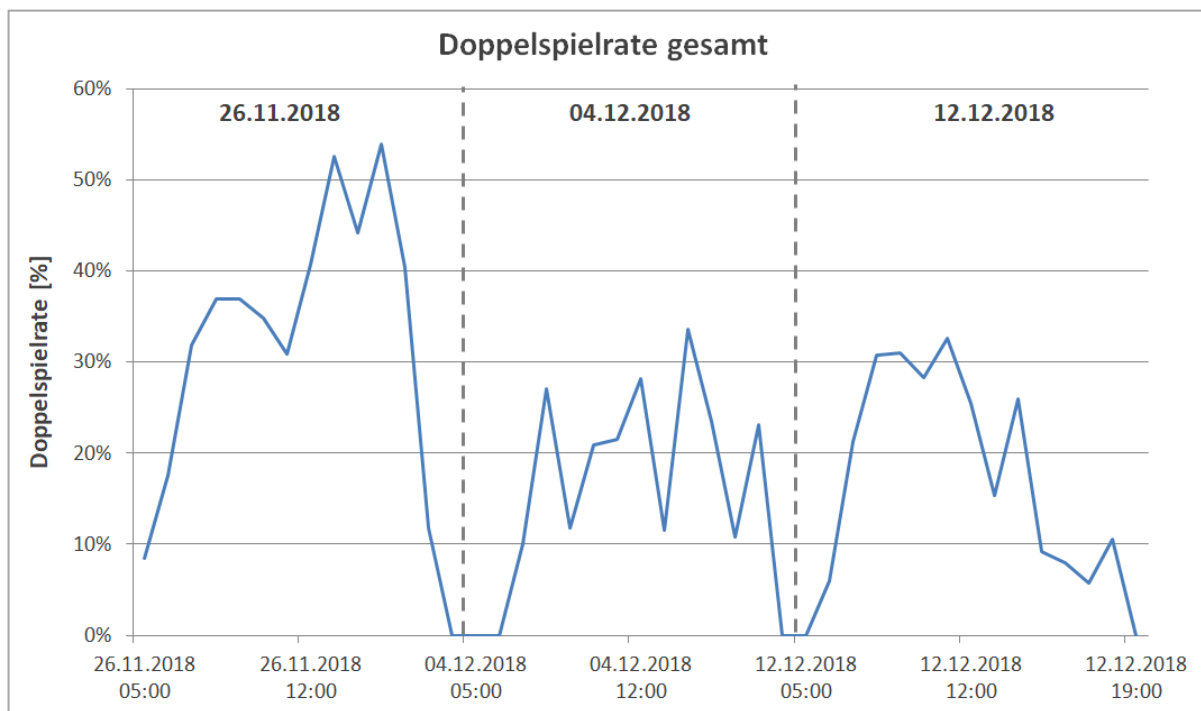


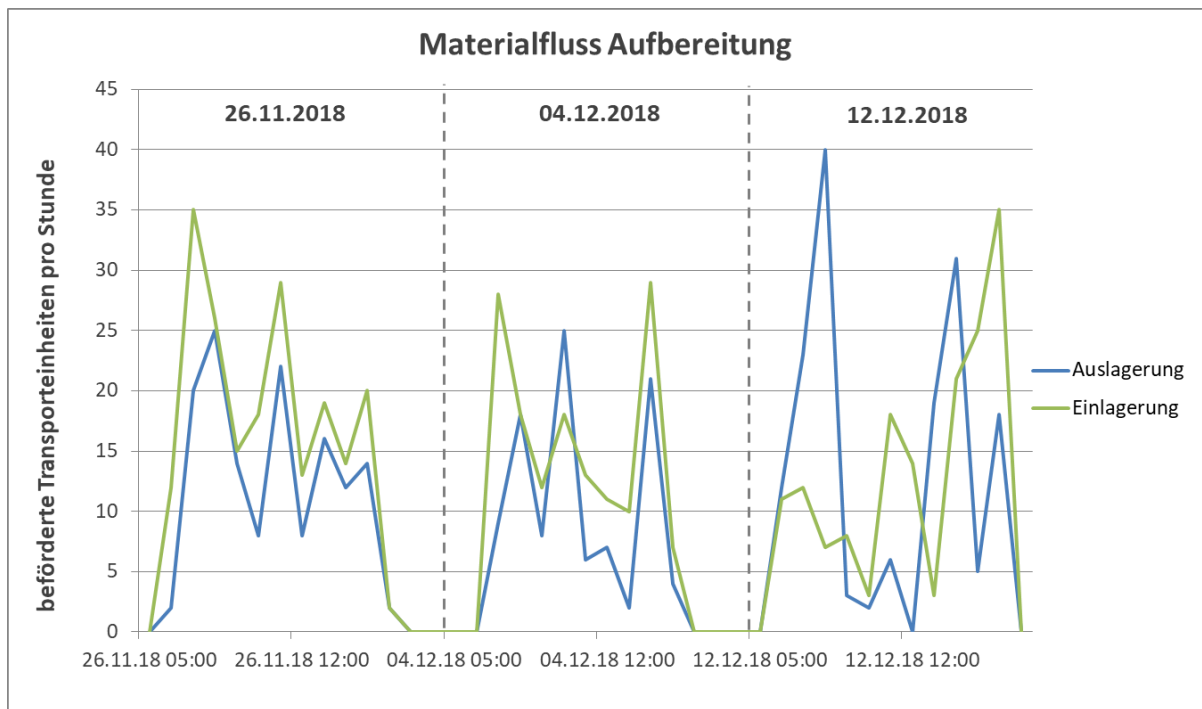
Abbildung 58: Durchschnittliche stündliche Doppelspielrate gesamt

Um diesen rückläufigen Trend der Doppelspiele genauer zu analysieren, wurde die Anzahl der Ein- und Auslagerungen an den jeweiligen Tagen miteinander verglichen, um herauszufinden, ob prinzipiell ein höherer Wert erreichbar gewesen wäre. In Tabelle 4 ist ersichtlich, dass am 26.11.2018 und am 12.12.2018 fast gleichviele Ein- und Auslagerungen stattgefunden haben, die Doppelspielrate am zweiten Tag aber wesentlich schlechter ausfiel.

Tabelle 4: Vergleich der Ein- und Auslagerungen mit der Doppelspielrate

Tag	Doppelspielrate	Einlagerungen	Auslagerungen
26.11.2018	38%	1849	1863
04.12.2018	21%	1461	1266
12.12.2018	24%	1640	1688

Warum die theoretisch erreichbaren Doppelspiele aber dennoch nicht zustande gekommen sind, wird erst ersichtlich, wenn man die Verteilung der Fahrten über den Tag hinweg betrachtet. Dabei stellte sich für fast alle Stationen ein Bild wie in Abbildung 59 ein. Hier sind zwar über den Tag hinweg genug Fahrten vorhanden um eine hohe Doppelspielrate zu erreichen, die ungleichmäßige Verteilung macht es allerdings für die ARCs unmöglich, die Transporte zu kombinieren.



Tag	Einlagerungen	Auslagerungen	Doppelspiele	mögliche Doppelspiele
26.11.2018	203	148	64%	73%
04.12.2018	146	100	38%	68%
12.12.2018	160	171	29%	94%

Abbildung 59: Materialfluss Aufbereitungsstation

Anhand dieser Auswertungen konnte der Kunde darauf aufmerksam gemacht werden, dass die Arbeiter in den Kommissionierstationen das System kontinuierlicher betreiben sollten. Am 12.12.2018 wurde zu Beginn der Schicht vermutlich ein Großteil der TE auf Vorrat ausgelagert, kommissioniert und dann zeitversetzt, gesammelt wieder eingelagert. Bei dieser Arbeitsweise werden nur sehr wenige Doppelspiele generiert und somit kann das Leistungspotential der Anlage nicht voll ausgeschöpft werden.

5.2 Fall 2: Systemupdate

In diesem Fall wurde zur Verbesserung der Kommunikationszeiten der ARCs ein Softwareupdate auf eine Kundenanlage aufgespielt. Um festzustellen, ob dieses Update die gewünschten Verbesserungen der Anlagenleistung bringt, wurde eine Auswertung durchgeführt.

In Abbildung 60 sind die Förderleistungen in TE/h für jeweils eine Woche vor und nach dem Update dargestellt. Es ist erkennbar, dass es in der Woche nach dem Update mehrere Stunden gab, in denen die Anzahl der beförderten Behälter annähernd den Wert von 450 TE/h erreichte. Da die Förderleistung neben den Fahrzeiten aber auch von der Auftragsverteilung abhängt, ist diese Auswertung nur begrenzt aussagekräftig für den Erfolg des Updates. Es wurden deshalb auch die Fahrzeiten zwischen bestimmten Stationen miteinander verglichen.

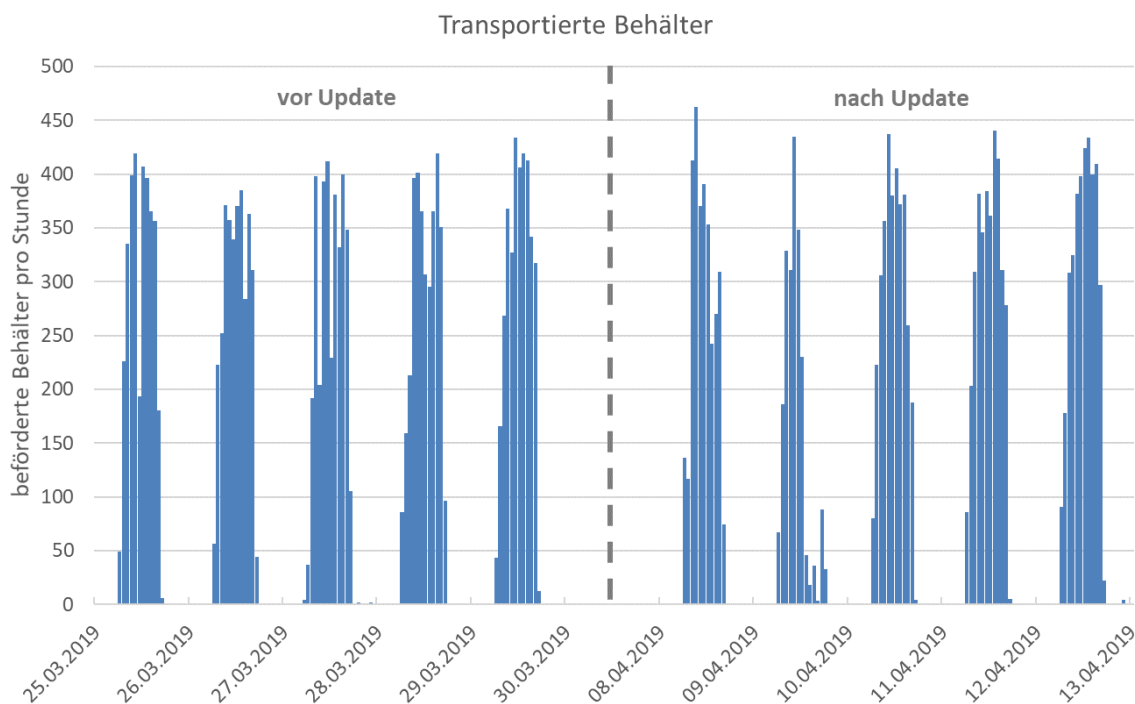


Abbildung 60: Anzahl der beförderten Behälter pro Stunde, vor und nach dem Update

Für diesen Vergleich wurde für jede Fahrtenkombination ein Histogramm, bei welchem die Fahrdauer mit einer Klassenbreite von 10 Sekunden auf der Abszisse und der Anteil an den Gesamtfahrten auf der Ordinate aufgetragen ist, erstellt. Dabei ergab sich für alle Fahrten ein ähnliches Bild, das in Abbildung 61 und Abbildung 62 für zwei Fahrten exemplarisch dargestellt ist.

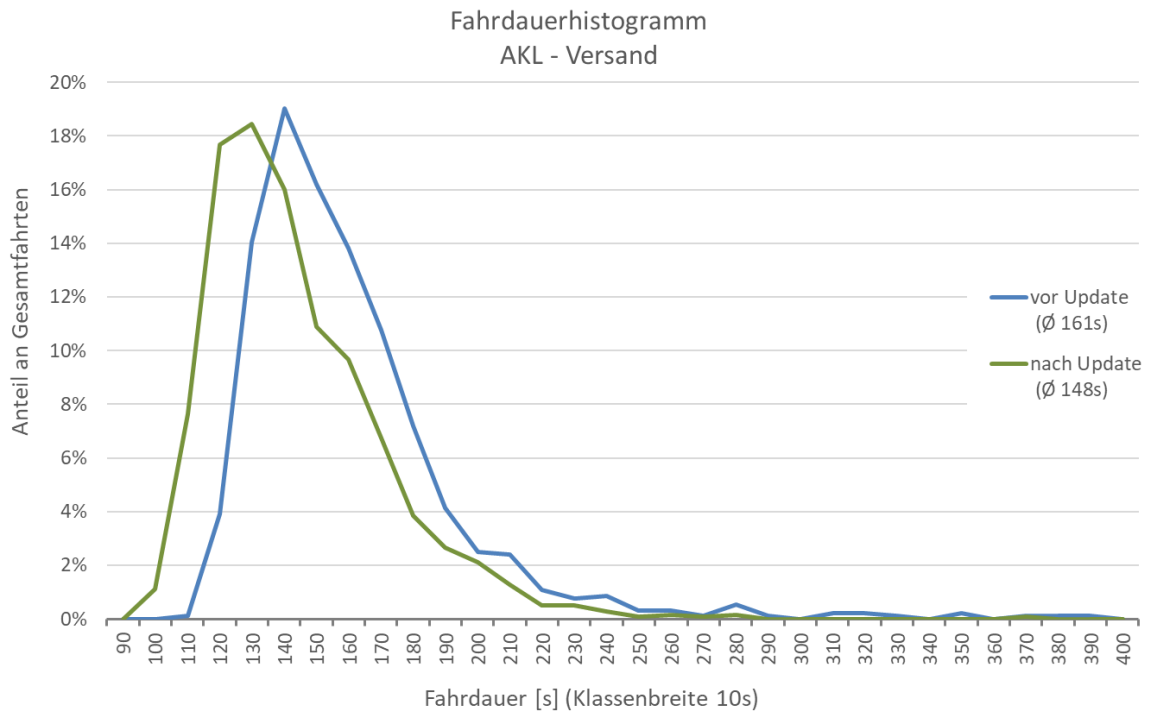


Abbildung 61: Fahrdauerhistogramm AKL-Versand

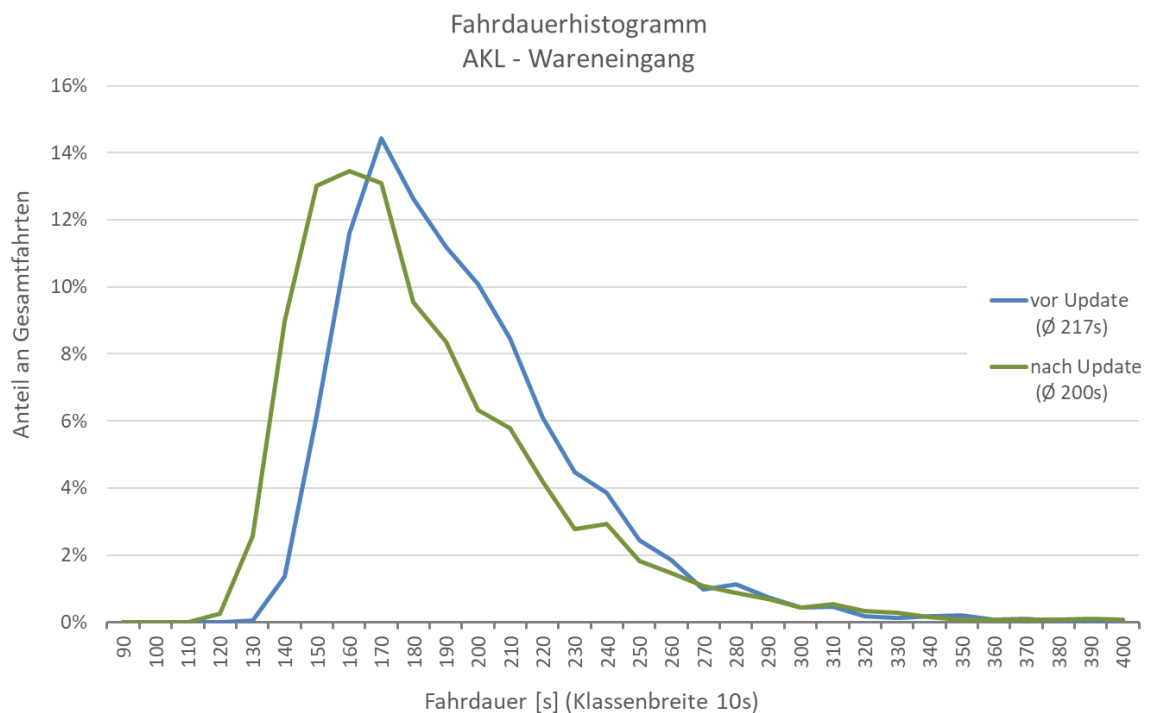


Abbildung 62: Fahrdauerhistogramm AKL-Wareneingang

Bei der Betrachtung der Histogramme und der Berechnung der durchschnittlichen Fahrdauer ist erkennbar, dass die Fahrdauer nach dem Update im Schnitt um ca. 15 Sekunden abgenommen hat. Vor und nach dem Update wurden mehrere tausend Fahrtenkombinationen aufgenommen, weshalb davon ausgegangen werden kann, dass es sich um eine repräsentative Darstellung handelt. Das Update kann somit als erfolgreich angesehen werden.

5.3 Fall 3: Ermittlung der Leistungsreserve

Bei diesem Fall wollte der Anlagenbetreiber seine Produktion ausbauen und benötigte daher eine höhere Leistung der Intralogistikanlage. Aus diesem Grund wurden Materialflussanalysen durchgeführt, um herauszufinden, ob die Anlage in der momentanen Konfiguration die neue Leistungsanforderung erfüllen kann.

Im ersten Schritt wurden die beförderten TE/h einer Woche mit dem Auslegungspunkt verglichen (Abbildung 63). Dies wird bei jeder Auswertung gemacht um einen Überblick der Anlage zu erhalten. Hier ist ersichtlich, dass das System immer nahe und teilweise über der verkauften Leistung betrieben wird. Man kann ebenfalls erkennen, dass von 07:00 bis 12:00 und von 13:00 bis 16:00 Uhr gearbeitet wird. Dieser Zeitraum wird für die folgenden Analysen herangezogen.

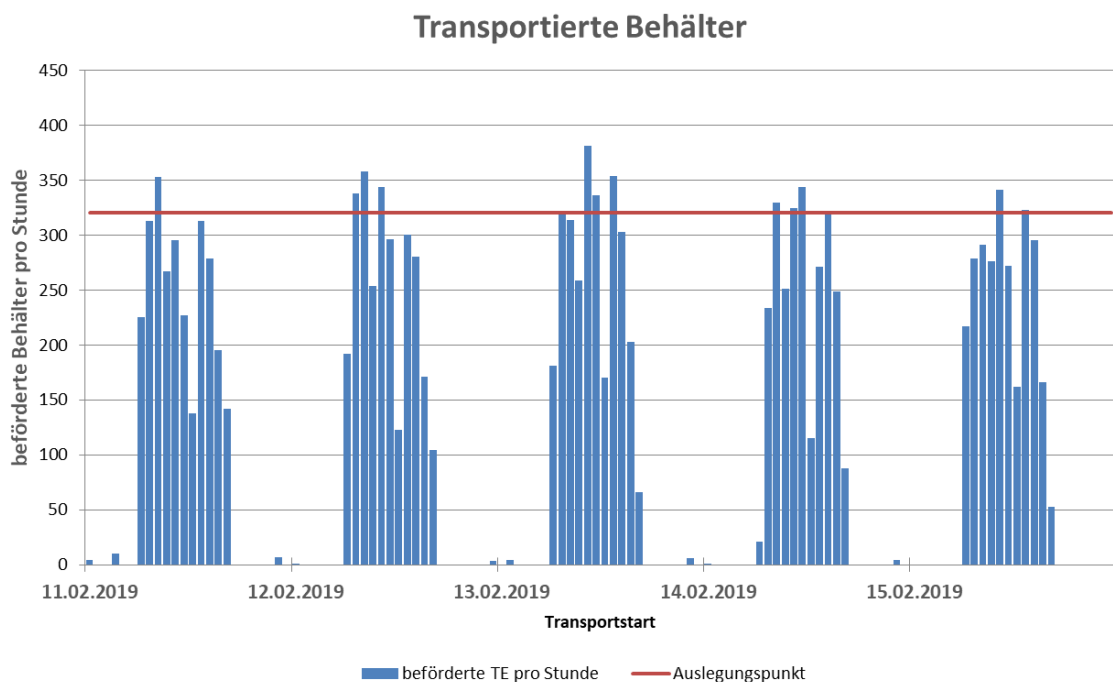


Abbildung 63: Beförderte TE pro Stunde im Fall 3.

Im nächsten Schritt erfolgte die Ermittlung der Aufgabenverteilung der ARCs (Tabelle 5). Hier ist deutlich zu erkennen, dass die ARCs selten und am stärksten Tag (13.02.2019) gar nicht parken. Die ARCs sind somit sehr gut ausgelastet und bieten wenige Leistungsreserven.

Tabelle 5: Aufgabenverteilung der ARCs im Fall 3

Arbeitstag 7-12 / 13-16 Uhr	Transport	Anfahrt	Parkfahrt	Parken
11.02.2019	62%	29%	1%	8%
12.02.2019	63%	29%	1%	7%
13.02.2019	70%	29%	0%	0%
14.02.2019	68%	26%	0%	5%
15.02.2019	65%	28%	1%	6%

Aus diesem Grund müssen dem System für die gewünschte Erhöhung der Leistung mehr ARCs zugeführt werden. Bevor dies geschieht, muss allerdings noch die Auslastung der restlichen Anlagenkomponenten kontrolliert werden, da es sonst im Falle einer Überlastung zu einer erhöhten Staubildung und damit verbunden zu Leistungseinbußen kommen kann.

Da die geraden Streckenabschnitte, Kurven und Weichen sehr hohe Grenzdurchsätze aufweisen, werden für diese Analyse nur Heber und Paternoster betrachtet.

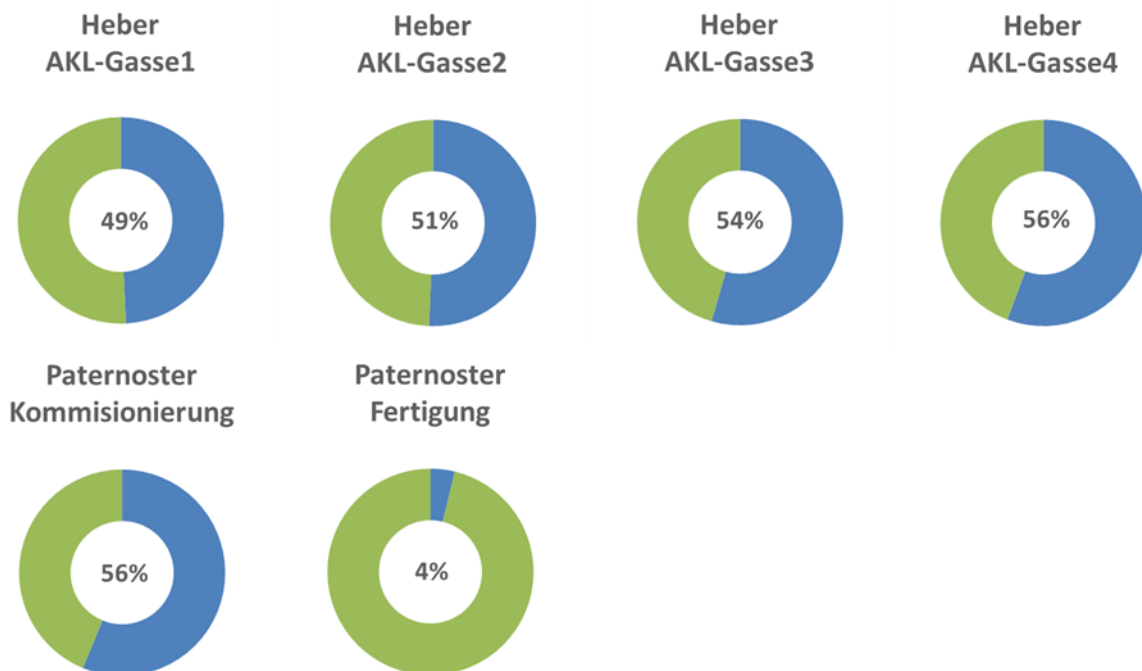


Abbildung 64: Maximale Auslastung der Heber und Paternoster im Fall 3: Auslastung (blau), Leistungsreserve (grün)

Wie in Abbildung 64 ersichtlich, liegt die maximale Auslastung der Heber und der Paternoster bei ca. 50%. Somit kann die Leistung ohne zusätzlichen Aufwand erhöht werden. Es muss allerdings darauf geachtet werden, dass es bei einer Heberauslastung von 85-90% schon zu sehr langen Warteschlangen kommen kann und dieser Wert somit nicht überschritten werden sollte (vgl. [GUD10], S. 494)

5.4 Fall 4: Verfügbarkeit

In diesem Fall reklamierte ein Anlagenbetreiber das häufige Auftreten von Systemfehlern, die er auf einen einzelnen ARC zurückführte.

Ziel war es, diese Theorie mit Hilfe des neuen Auswertungstools zu überprüfen.

Zu Beginn wurden die Verfügbarkeit und die aufgetretenen Fehler der letzten Tage ausgewertet. Dabei stellte sich folgendes Bild ein (Abbildung 65):

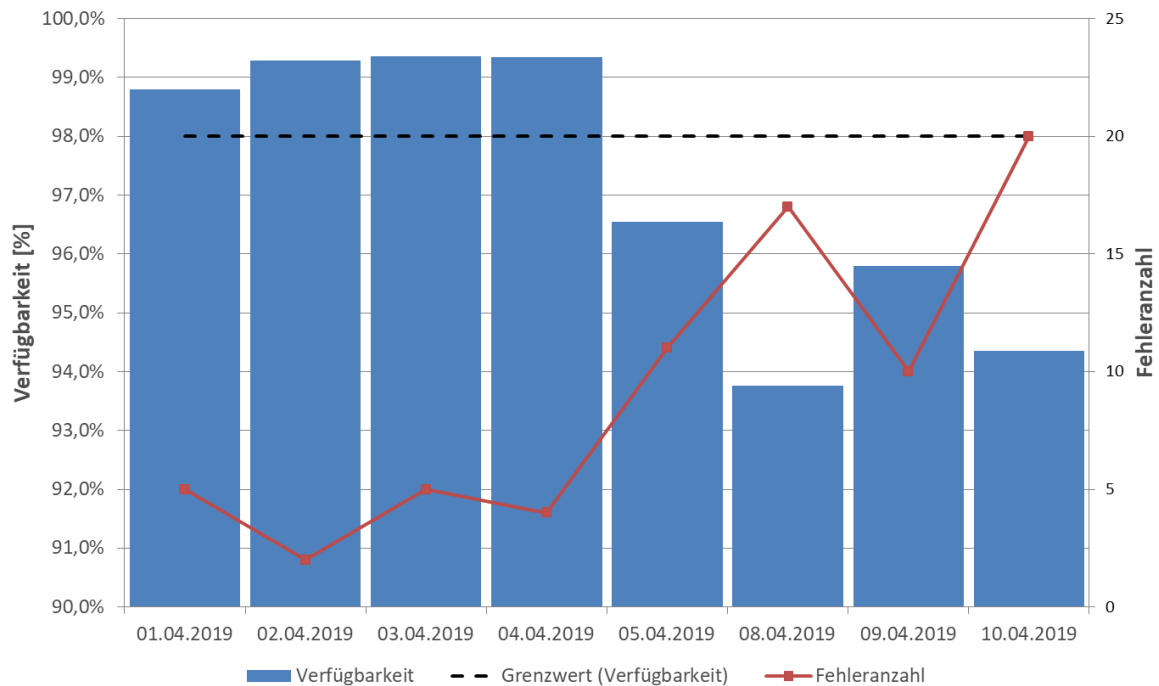


Abbildung 65: Verfügbarkeit und Anzahl der Fehler pro Tag vom 01.04.2019-10.04.2019

Es wird deutlich, dass der definierte Verfügbarkeitsgrenzwert von 98% in den letzten vier Tagen des Beobachtungszeitraums unterschritten wurde. Auch die Anzahl der Fehler stieg währenddessen an. In Tabelle 6 sind jene Komponenten dargestellt, die im ausgewählten Zeitraum die meisten Fehler aufwiesen. Es kann eine Fehlerhäufung bei ARC 14 erkannt werden, wodurch sich der Verdacht des Anlagenbetreibers bestätigt.

Tabelle 6: Anzahl der Fehler und deren Dauer pro Komponente

Komponente	Fehleranzahl	Dauer [min]
ARC 14	31	351
AKL01-HEB21	7	31
ARC 7	5	23
ARC 9	5	63
LOG00-WEI21	5	14
LOG03-HEB11	5	27
ARC 11	4	34
ARC 4	3	31

In Tabelle 7 und Tabelle 8 wird der ARC 14 genauer betrachtet. Wie in Tabelle 7 gezeigt, nahm die Fehlerrate ab dem 05.04.2019 zu. In Tabelle 8 wird deutlich, dass ein blockierter Zinken des ARCs sowie Probleme beim Auf- bzw. Abladen der TE die Hauptfehlerquellen darstellen.

Tabelle 7: Fehler pro Tag von ARC 14

Tag	Fehleranzahl	Dauer [min]
01.04.2019	1	7
02.04.2019	0	0
03.04.2019	2	12
04.04.2019	1	2
05.04.2019	9	91
08.04.2019	9	96
09.04.2019	4	38
10.04.2019	5	106

Tabelle 8 Häufigkeit der einzelnen Fehlerarten bei ARC 14

Fehlerart	Fehleranzahl	Dauer [min]
Transporteinheit wurde nicht auf-/abgeladen oder Sensor hat nicht reagiert	15	244
Mindestens ein Zinken ist blockiert	11	87
Ein internes Modul hat einen Fehler oder kann nicht initialisiert werden	2	7
Nicht aktiviert	2	5
Der Motor reagiert nicht	1	8

Der ARC 14 wurde am 10.04.2019 aus dem System genommen und zur Reparatur zu Servus Intralogistics geschickt. Nach weiteren vier Tagen erfolgte abermals eine Verfügbarkeitsanalyse. In Abbildung 66 ist ersichtlich, dass die Verfügbarkeitswerte nach der Entnahme des fehlerhaften ARCs wieder oberhalb bzw. nahe des Grenzwertes liegt.

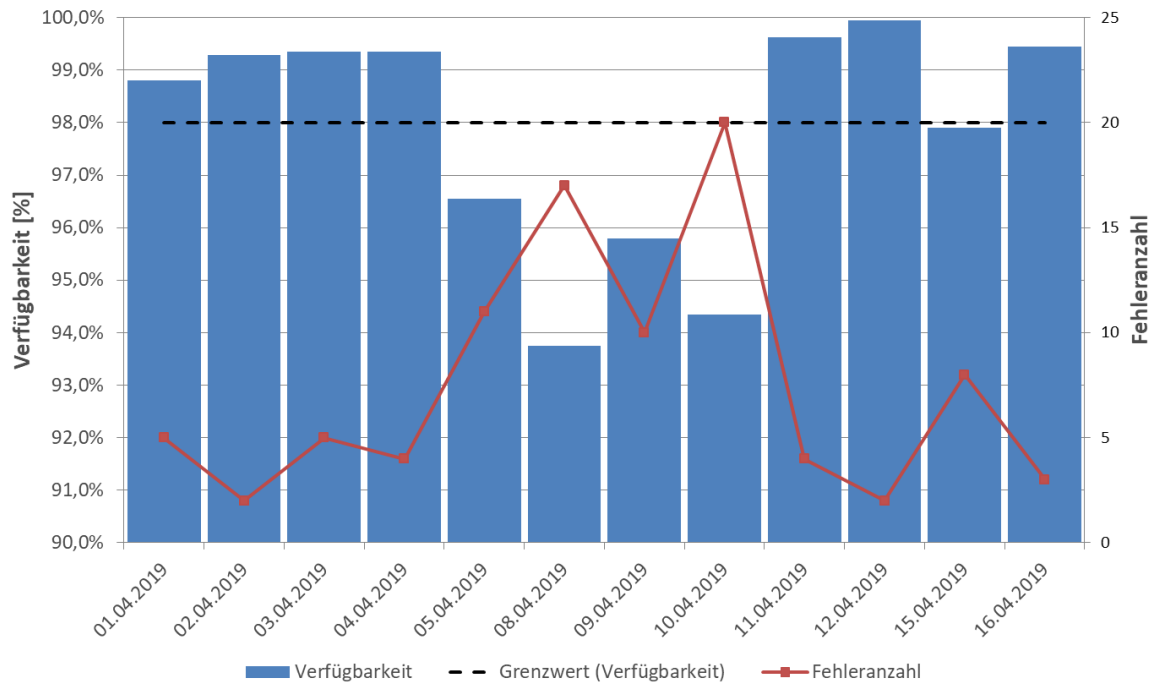


Abbildung 66: Verfügbarkeit und Anzahl der Fehler pro Tag vom 01.04.2019-16.04.2019

Da die Anlagenbetreiber den Support in solch einem Fall meist sehr spät kontaktieren, wäre es sinnvoll die Grenzwerte der Verfügbarkeit und die Fehlerhäufung automatisch zu überwachen. So könnte effizienter und vor allem frühzeitig auf Fehler reagiert werden.

6 Zusammenfassung und Ausblick

Die Servus-Logistikanlagen bestehen aus autonomen, schienengebundenen Transportrobotern (ARCs) und mehreren Assistenten (z.B. Heber, Weichen, Ladestationen). Der Logistikprozess geht vom Lagerverwaltungssystem aus, indem es einen Fahrauftrag an das übergeordnete System (SEAS) sendet, welches den Auftrag an einen ARC weiterleitet. Hat der ARC einen Auftrag erhalten, sucht er sich selbständig den schnellsten Weg zu seinem Ziel. Auf dem Fahrweg steuern die ARCs sich und ihre Assistenten selbständig. Beim Betrieb der Anlagen fallen große Datenmengen auf den Servern der Kunden an. Dazu zählen unter anderem Fahrauftragsdaten, ARC-Positionsdaten, Assistententelegramme, Pfadmanagementdaten und Meldesystemdaten. Diese werden laufend auf einen Elasticsearch-Analyseserver bei Servus transferiert.

Im Rahmen dieser Arbeit wurden Kennzahlen zur Analyse von systeminternen Vorgängen erarbeitet. Der inhaltlich größte Teil befasst sich mit dem Weg von den Ausgangsdaten zu diesen Kennzahlen. Hierfür müssen zuerst die ARC-Positionsdaten und die Telegramme aus den Log-Daten herausgefiltert und aufbereitet werden, um daraus die Daten für jeden einzelnen Auftrag, jede Assistentenbewegung, die ARC-Positionen, die Betriebsstörungen sowie die blockierten Anlagenteile ermitteln zu können. Mit diesen Informationen erfolgt anschließend die Berechnung der festgelegten Kennzahlen und deren Darstellung. Dazu werden im Visualisierungstool Kibana Dashboards erstellt.

Wie im Ergebnisteil gezeigt, bieten die ausgewerteten Daten die Möglichkeit, Fehler, Verbesserungspotential und Engpässe im System rasch zu erkennen. Zurzeit handelt es sich dabei noch um einen Prozess, der aktiv von einem Mitarbeiter oder Kunden gestartet werden muss. Um dies zu verbessern, könnte aufbauend auf dieser Arbeit ein Dienst erstellt werden, der automatisiert den Verlauf der einzelnen Kennzahlen (z.B. Verfügbarkeit, Doppelspielrate, Fahrzeiten, Assistentenzeiten etc.) überwacht und einem Servus-Mitarbeiter Meldung erstattet, falls ein vordefinierter Grenzwert überschritten wird. So könnte schneller auf Probleme reagiert und Verbesserungspotential erkannt werden.

Um das Analyseverfahren effizienter zu gestalten, könnte der Umweg über den Elasticsearch-Analyseserver ausgelassen und die Auswertung direkt auf dem SEAS-Server des Kunden erfolgen. Anstelle der umfangreichen Rohdaten müssten so nur die gewünschten Kennzahlen vom Kunden zu Servus transferiert werden. Die Analyse der Daten könnte mit dem, für diese Arbeit erstellten Programm erfolgen. Dabei müssten lediglich die Input- und Output-Klassen so umprogrammiert werden, dass somit das Auslesen und Beschreiben der SEAS-Datenbank möglich ist.

7 Verzeichnisse

7.1 Abkürzungsverzeichnis

AKL	Automatisches Kleinteilelager
ARC	Autonomous Robotic Carrier
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LVS	Lagerverwaltungssystem
RFID	Radio-Frequency Identification
SEAS	Servus Efficient Agents System
SERAN	Servus Area Network
SPS	Speicherprogrammierbare Steuerung
TE	Transporteinheit
WA	Warenausgang
WE	Wareneingang

7.2 Abbildungsverzeichnis

Abbildung 1: Servus-Systems aus Datensicht	2
Abbildung 2: Beispiel eines Anlagenlayouts. 1) AKL, 2) Pufferplatz, 3) Arbeitsplatz, 4) Weiche, 5) Drehweiche, 6) Fahrstrecke, 7) Heber,.....	3
Abbildung 3: Servus Fahrstrecke	4
Abbildung 4: ARC mit Lastaufnahmezinken.....	5
Abbildung 5: Beispiel für einen RFID-Knoten.....	5
Abbildung 6: Beispiel für eine Weiche	6
Abbildung 7: Beispiel für einen 2-achsigen Heber	7
Abbildung 8: Beispiel für einen Paternoster.....	8
Abbildung 9: Beispiel für einen Arbeitsplatz.....	8
Abbildung 10: Beispiel für einen Pufferplatz.....	9
Abbildung 11: Ablauf der Datenerfassung.....	9
Abbildung 12: Aufbau eines Elasticsearch-Clusters	11
Abbildung 13: Gliederung von Kennzahlen (nach [MAL79], S. 5).....	12
Abbildung 14: Ablauf der Modellbildung	16
Abbildung 15: Schematischer Aufbau des Datenmodells mit den Ausgangsdaten (rot), der Datenaufbereitung (blau), den Methoden (grün) und den Kennzahlen (grau).	17
Abbildung 16: Eigenschaften eines Fahrauftrags und deren Datentypen: Absolutzahl (#), Text (t), Zeit (time) und boolesche Variable (bool)	20
Abbildung 17: Eigenschaften einer Assistentenbewegung und deren Datentypen: Absolutzahl (#), Text (t), Zeit (time) und boolesche Variable (bool)	22
Abbildung 18: Eigenschaften der Assistentenauslastung und deren Datentypen: Absolutzahl (#), Verhältniszahl (/), Text (t) und Zeit (time)	23

Abbildung 19: Telegrammtypen bei der Fahrt eines ARCs durch einen Heber: Vorregistrierung (a), Registrierung (b), Einfahrtfreigabe (c), Heberfreigabe (d), Ausfahrberechtigung (e), Abmeldung (f)	24
Abbildung 20: Wahrscheinlichkeitsdichtefunktion und Verteilungsfunktion für unterschiedlich streuende Zwischenankunftszeiten ([AF07], Abb. 2.9)	26
Abbildung 21: Teilstetige Verzweigung mit einer stetigen und einer unstetigen Verbindung ([AF07], Abb. 2.11)	27
Abbildung 22: Durchsatzbedingung einer teilstetigen Verbindung ([AF07], Abb. 2.12)	27
Abbildung 23: Unstetige Verzweigung mit zwei Verbindungen ([AF07], Abb. 2.14)	28
Abbildung 24: Stetige Verzweigung mit zwei Richtungen ([AF07], Abb. 2.10) ..	29
Abbildung 25: Durchsatzdiagramm einer stetigen Verzweigung ohne Durchsatzverluste (a) und mit Durchsatzverlusten (b) ([AF07], Abb. 2.16)	29
Abbildung 26: Universelles Materialflusselement ([AF07], Abb. 2.26)	30
Abbildung 27: Eigenschaften der ARC-Positionsdaten und deren Datentypen: Absolutzahl (#), Verhältniszahl (/), Text (t), Zeit (time) und boolesche Variable (bool)	33
Abbildung 28: Layout der Beispielanlage	35
Abbildung 29: Verfügbarkeitsbaum der Beispielanlage	36
Abbildung 30: Beispiel einer Störung der Anlage	36
Abbildung 31: Ablaufschema zur Ermittlung der Leistungsverfügbarkeit ([VDI10], Abb.1).“	38
Abbildung 32: Vertraglich festgelegter Materialfluss	38
Abbildung 33: Eigenschaften einer Störung und deren Datentypen: Absolutzahl (#), Verhältniszahl (/), Text (t) und Zeit (time)	39
Abbildung 34: Eigenschaften der Verfügbarkeit und deren Datentypen: Absolutzahl (#), Verhältniszahl (/), Text (t) und Zeit (time)	40
Abbildung 35: Arten von Graphen: ungerichteter Graph (a), gerichteter Graph (b), ungerichteter Graph mit Mehrfachkanten (c) und gerichteter Graph mit Mehrfachkanten (d)	41
Abbildung 36: Beispielgraph mit $n = 5$ und $m = 7$	41
Abbildung 37: Pseudocode für den Floyd-Warshall Algorithmus	43
Abbildung 38: Schätzwert abhängig von der Gewichtung w_{rel}	45
Abbildung 39: Schätzwert der Mindestwertmethode	46
Abbildung 40: Ablaufdiagramm zur Auffindung von blockierten ARCs	47
Abbildung 41: Materialflussdashboard	48
Abbildung 42: Verfügbarkeitsdashboard	49
Abbildung 43: Assistenten Übersichtsdashboard	50
Abbildung 44: Assistenten Detaildashboard	50
Abbildung 45: ARC Übersichtsdashboard	51
Abbildung 46: Detaillierter Materialfluss mit Stationsübersicht	52
Abbildung 47: Detaillierter Materialfluss mit Transportmatrix	52
Abbildung 48: Fahrtdauer Dashboard	53
Abbildung 49: Störungsanalyse Dashboard	53
Abbildung 50: Kibana Discover	54
Abbildung 51: Kibana Dashboard	55
Abbildung 52: Softwareaufbau	57
Abbildung 53: Basisdaten für die Anlagenauswertung	58

Abbildung 54: Anlagenspezifische Konfigurationsdaten für die Anlagenauswertung	59
Abbildung 55: Inputdaten für die Anlagenauswertung	60
Abbildung 56: Ausgewertete Daten (Fahraufträge, ARC-Aufgaben, Assistentenbewegungen, Assistentenauslastung).....	61
Abbildung 57: Ausgewertete Daten (ARC-Position, Störungen, Verfügbarkeit). 62	
Abbildung 58: Durchschnittliche stündliche Doppelspielrate gesamt.....	64
Abbildung 59: Materialfluss Aufbereitungsstation	65
Abbildung 60: Anzahl der beförderten Behälter pro Stunde, vor und nach dem Update	66
Abbildung 61: Fahrdauerhistogramm AKL-Versand.....	67
Abbildung 62: Fahrdauerhistogramm AKL-Wareneingang.....	67
Abbildung 63: Beförderte TE pro Stunde im Fall 3.....	68
Abbildung 64: Maximale Auslastung der Heber und Paternoster im Fall 3: Auslastung (blau), Leistungsreserve (grün)	69
Abbildung 65: Verfügbarkeit und Anzahl der Fehler pro Tag vom 01.04.2019-10.04.2019	70
Abbildung 66: Verfügbarkeit und Anzahl der Fehler pro Tag vom 01.04.2019-16.04.2019	72

7.3 Tabellenverzeichnis

Tabelle 1: Fahraufträge mit Start- und Zieladressen	21
Tabelle 2: Streckenmatrix	21
Tabelle 3: Verfügbarkeitstabelle	37
Tabelle 4: Vergleich der Ein- und Auslagerungen mit der Doppelspielrate.....	64
Tabelle 5: Aufgabenverteilung der ARCs im Fall 3.....	69
Tabelle 6: Anzahl der Fehler und deren Dauer pro Komponente	70
Tabelle 7: Fehler pro Tag von ARC 14	71
Tabelle 8 Häufigkeit der einzelnen Fehlerarten bei ARC 14	71

7.4 Literaturverzeichnis

- [AF07] ARNOLD, Dieter ; FURMANS, Kai: *Materialfluss in Logistiksystemen*. 5., erweiterte Auflage. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2007 (VDI-Buch)
- [CK17] CRAMER, Erhard ; KAMPS, Udo: *Grundlagen der Wahrscheinlichkeitsrechnung und Statistik : Eine Einführung für Studierende der Informatik, der Ingenieur- und Wirtschaftswissenschaften*. 4., korrigierte und erweiterte Auflage. Berlin : Springer Spektrum, 2017 (Springer-Lehrbuch)
- [ELA19] Elasticsearch: *Basic Concepts / Elasticsearch Reference [7.0] / Elastic*. URL
<https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started-concepts.html>. – Aktualisierungsdatum: 2019-04-10 – Überprüfungsdatum 2019-04-25

- [FEM89] FEM: *FEM 9.222 : Regeln über die Abnahme und Verfügbarkeit von Anlagen mit Regelbediengeräten und anderen Gewerken* : VDMA Verlag, 1989
- [GUD10] GUDEHUS, Timm: *Logistik: Grundlagen - Strategien - Anwendungen*. s.l. : Springer Berlin Heidelberg, 2010
- [ISO05] *ISO 9000 Quality management systems*
- [LÄU91] LÄUCHLI, Peter: *Algorithmische Graphentheorie*. Basel : Birkhäuser, 1991 (Programm-Praxis 9)
- [MAL79] MALUCHE, Christoph: *Entwicklung eines Kennzahlensystems für Entwicklung eines Kennzahlensystems für den Produktionsbereich auf der Basis sekundär-statistischer Daten*. Aachen, 1979
- [MAR09] MARTIN, Heinrich: *Transport- und Lagerlogistik : Planung, Struktur, Steuerung und Kosten von Systemen der Intralogistik ; mit 39 Tabellen*. 7., erw. und aktualisierte Aufl. Wiesbaden : Vieweg + Teubner, 2009 (Praxis)
- [MS09] MOSLER, Karl C. ; SCHMID, Friedrich: *Beschreibende Statistik und Wirtschaftsstatistik*. 4., aktualisierte und verb. Aufl. Dordrecht : Springer, 2009 (Springer-Lehrbuch)
- [VDI92] VDI: *3649: Anwendung der Verfügbarkeitsrechnung für Förder- und Lagersysteme*. Berlin : Beuth, 1992
- [VDI04] VDI: *3581: Verfügbarkeit von Transport- und Lageranlagen sowie deren Teilsysteme und Elemente*. Berlin : Beuth, 2004
- [VDI10] VDI: *4486: Zuverlässigkeit in der Intralogistik: Leistungsverfügbarkeit*. Berlin : Beuth, 2010
- [VDI14] VDI: *2692: Shuttle-Systeme für Kleinbehälterlagerung*. Berlin : Beuth, 2014