



FAKULTÄT FÜR INFORMATIK  
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatik

**Softwaregestützte Prozessoptimierung der  
Einststeuerung von Dächern in der MAN-  
Fahrerhausmontage**

**Software-supported process enhancement for  
placement of roofs in the driving cab mounting at  
MAN**

<b>Bearbeiter:</b>	<b>B. Sc. Benjamin Brandenbourger</b>
<b>Aufgabensteller:</b>	<b>Prof. Dr.-Ing. Alois Knoll</b>
<b>Betreuer:</b>	<b>Dipl.-Inf. Michael Geisinger</b>
<b>Abgabedatum:</b>	<b>15. Februar 2011</b>



Ich versichere, dass ich diese Master's Thesis selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

---

Ort, Datum

---

Unterschrift

## **DANKSAGUNG**

Ich möchte mich bei Herrn Prof. Dr.-Ing. Alois Knoll bedanken, der mir die Bearbeitung dieser Masterarbeit ermöglicht hat.

Zu Dank bin ich auch Herrn Dipl.-Inf. Michael Geisinger und Herrn Dipl.-Ing. (FH) Franz Blaschke verpflichtet, die mir jederzeit hilfsbereit mit Ihrer Erfahrung und Rat zur Seite standen. Außerdem danke ich der Abteilung PHFMB-3, die mir benötigte Informationen und wertvolle Hinweise auf unbürokratische Weise vermittelten.

Nicht zuletzt möchte ich mich bei meinen Eltern bedanken, da sie nicht nur mein Studium zum größten Teil finanziert haben, sondern auch ständig ein großes Interesse an meiner Arbeit zeigten und mich so gut es ging unterstützten.

## ZUSAMMENFASSUNG

Die Masterarbeit wurde am Lehrstuhl Robotics and Embedded Systems der Technischen Universität München wie auch am MAN-Werk für Nutzfahrzeuge in München erstellt. Das Thema der Arbeit war die Entwicklung eines Systems zur Prozessoptimierung der Einsteuerung von Fahrzeugdächern in der MAN Dachmontage.

Bis zu Beginn der Arbeit wurde der Prozessfluss intuitiv und manuell geregelt. Nach einer kurzen Ist-Soll-Analyse wurde das Konzept entwickelt, ein verteiltes System basierend auf einer Client/Server-Architektur zu realisieren. Mit der Verabschiedung eines Software-Lastenhefts wurde ein Datenbankmodell erstellt, welches den geforderten Anforderungen gerecht wurde. Daraufhin wurde mit der Implementierung einzelner grafischer Oberflächen begonnen, die in einem regelmäßigen Iterationsverfahren getestet, vorgeführt und angepasst wurden. Nach zahlreichen Tests der einzelnen Komponenten konnte schließlich das Dach Management System (DMS) als Einheit in Betrieb genommen werden, um tatsächliche Arbeitserleichterungen durch Zeitersparnis beobachten zu können.

# INHALTSVERZEICHNIS

<b>1</b>	<b>Einleitung</b> .....	<b>7</b>
<b>2</b>	<b>Zielsetzung</b> .....	<b>8</b>
<b>3</b>	<b>Theoretische Grundlagen</b> .....	<b>9</b>
3.1	Scheduling-Strategien .....	9
3.1.1	Beispiel 1.....	12
3.1.2	Beispiel 2.....	13
3.2	Push-/ Pull-basierte Architektur.....	15
<b>4</b>	<b>Prozessbeschreibung</b> .....	<b>16</b>
4.1	Ist-Analyse .....	16
4.1.1	Störungsursachen.....	18
4.1.2	Zuweisung der Dächer zu den Arbeitsbühnen.....	19
4.1.3	Ausstattung der Dächer in den Arbeitsbühnen.....	19
4.1.4	Umsetzen der Dächer auf das Gurtband .....	20
4.1.5	Schwachstellenanalyse .....	20
4.2	Soll-Zustand.....	23
<b>5</b>	<b>Konzepte</b> .....	<b>25</b>
5.1	Anforderungen.....	25
5.2	SPS-gesteuertes Konzept .....	26
5.3	Datenbankbasiertes Konzept.....	27
<b>6</b>	<b>Operationalisierung der theoretischen Konzepte</b> .....	<b>28</b>
6.1	Implementierung der Komponenten.....	28
6.1.1	Master .....	30
6.1.2	Bühne.....	31
6.1.3	Gurtband .....	32
6.1.4	Daemon.....	34

6.2	Datenbankbasierte Client/Server-Architektur.....	36
6.2.1	Datenbankmodell.....	37
6.2.2	Datenbankzugriff auf IBM-Host.....	40
6.3	Kommunikation der Komponenten.....	42
6.4	Zuordnungsalgorithmus am Master.....	45
<b>7</b>	<b>Fazit.....</b>	<b>48</b>
7.1	Ergebnisse.....	48
7.2	Ausblick.....	49
<b>8</b>	<b>Literaturverzeichnis .....</b>	<b>52</b>
<b>9</b>	<b>Abbildungsverzeichnis.....</b>	<b>53</b>
<b>10</b>	<b>Tabellenverzeichnis.....</b>	<b>53</b>
<b>11</b>	<b>Anhang.....</b>	<b>54</b>
11.1	Begriffs- und Abkürzungsverzeichnis .....	54
11.2	Benutzerhandbuch für DMS .....	55

# 1 EINLEITUNG

Die Finanzkrise ab 2007, die als eine Banken- und Immobilienkrise begann, erreichte Ende 2008, mit dem Zusammenbruch der amerikanischen Bank "Lehman Brothers", die Realwirtschaft. Was als eine Kombination aus niedrigen Zinssätzen, Zahlungsausfällen und der Ausweitung der Kreditvergabe in den USA begann, breitete sich in Zeiten global zusammenhängender Finanzsysteme weltweit aus. Wie viele andere Großunternehmen, musste auch der börsennotierte Fahrzeug- und Maschinenbaukonzern "MAN" massive Auftragsrückgänge hinnehmen. Darauf reagierte das Unternehmen unter anderem mit Entlassungen und Kurzarbeit, die bis in das Jahr 2010 anhielt [1].

Ein Vorteil der Kurzarbeit bestand jedoch darin, dass Zeit für das Überdenken und Optimieren produktionsspezifischer Prozesse gefunden werden konnte. Zur Effizienzsteigerung wurde folglich die Prozessflussoptimierung mithilfe eines softwarebasierten Systems bei der Einsteuerung von LKW-Dächern in der Fahrerhausmontage beschlossen und zum Thema dieser Masterarbeit.

Da die Einsteuerung der Dächer bisher manuell durch einen Mitarbeiter geregelt wurde und kein genauer Prozessablauf definiert war, traten viele unerwünschte Wartezeiten auf, die die Effizienz der Abteilung stark beeinträchtigte. Diese Wartezeiten waren auf dem ersten Blick gering, jedoch summierten sie sich über die verschiedenen Arbeitsstationen und über den Tag hinweg auf. Somit musste durch eine Ist-Analyse Schwachstellen identifiziert und theoretische Gegenmaßnahmen entwickelt werden. Die gesammelten Ergebnisse halfen bei der Konzeptionierung einer Anwendung, die den Prozessfluss regulieren sollte. Da der Prozessablauf stark an ein Mehrprozessorsystem erinnert, konnte ein bewährter Zuteilungsalgorithmus als Basis für die Anwendung eingesetzt werden. Nach mehreren Iterationsstufen mit den verantwortlichen Vorgesetzten wurde schließlich die Software in der Produktion eingeführt. Dadurch wurde das Primärziel der Wartezeitminimierung erreicht, sodass die Mitarbeiter nun ruhiger und auch fehlerfreier arbeiten.

## 2 ZIELSETZUNG

Ziel der vorliegenden Masterarbeit ist die Entwicklung eines Systems zur Optimierung der Einsteuerung von Dächern in der MAN-Fahrerhausmontage und dadurch eine Minimierung der Wartezeiten.

Folgende Aufgaben sollen im Rahmen der Arbeit bearbeitet werden:

- Ist-/Soll-Analyse des Dachausstattungsprozesses
- Erarbeitung theoretischer Gegenmaßnahmen
- Erstellung eines Softwarelastenhefts
- Implementierung einer Anwendung mit verschiedenen grafischen Oberflächen (GUI) zur gleichmäßigen Auslastung der Mitarbeiter, basierend auf Montagezeiten
- Ermöglichen der Auswertung von Daten
  - Rückschluss von Fahrzeugnummer auf Montageuhrzeit und -bühne
  - Ansicht der Montagezeiten pro Tag/Woche/Monat je Bühne
  - Ansicht der Fehler pro Tag/Woche/Monat je Bühne
- Testen, Re-Design und Roll-Out der Applikation

## 3 THEORETISCHE GRUNDLAGEN

Bevor auf die praktische Umsetzung eingegangen wird, sollen zuerst die zugrunde liegenden theoretischen Grundlagen erläutert werden. Hierzu werden die Funktionsweise zweier Scheduling-Strategien wie auch Push- bzw. Pull-basierte Architekturen genauer erklärt.

### 3.1 SCHEDULING-STRATEGIEN

Scheduling-Strategien werden, wie der Name es sagt, zum Erstellen von schedules, also Ablaufplänen, verwendet. Diese werden benötigt, um Prozessen zeitlich begrenzte Ressourcen zuzuweisen. Anwendung finden Scheduling-Strategien in der Informatik, beispielsweise in der Zuweisung von Prozessen an Prozessoren.

Für nachstehende Erklärungen und Beispiele soll folgende Konvention eingeführt werden:

$P_i$  = Prozess  $i$

$r_i$  = Zeitpunkt, an dem der Prozess  $i$  bereit steht (ready)

$d_i$  = Frist des Prozesses  $i$  (deadline)

$t$  = verbleibende Berechnungszeit

$e_i$  = Dauer des Prozesses  $i$  (execution)

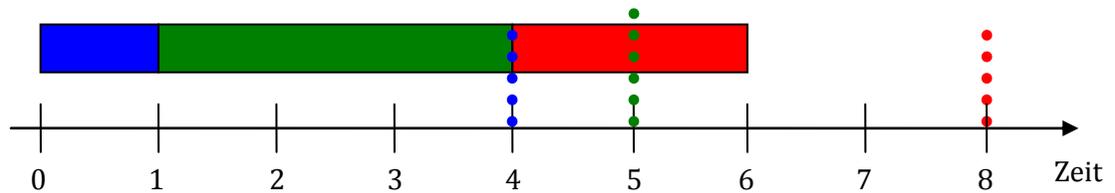
$sl_i$  = Spielraum des Prozesses  $i$  zu dessen Deadline (slack time)

Die erste untersuchte Scheduling-Strategie heißt Earliest Deadline First (EDF). Dabei handelt es sich um eine online und nicht präemptive Strategie, die einen Plan nach Fristen erstellt. Nicht präemptiv bedeutet, dass der Prozess nicht unterbrochen werden kann. Bei dem Algorithmus wird dem Prozess, der laut Frist als erster fertig sein muss, die Ressource zugewiesen. Bezogen auf nachstehendes Szenario würde sich folgende EDF-Strategie ergeben:

$P_1: r_1=0; e_1=2; d_1=8;$

$P_2: r_2=0; e_2=3; d_2=5;$

$P_3: r_3=0; e_3=1; d_3=4;$



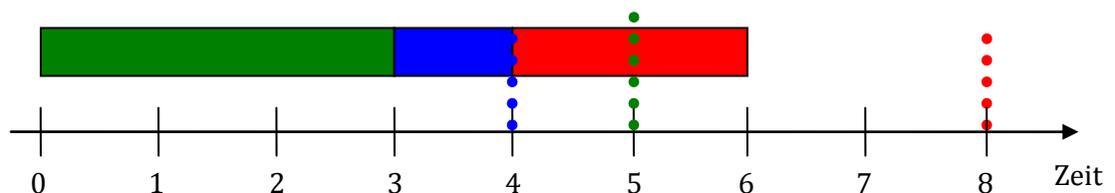
$P_3$  wird zuerst ausgeführt, da dessen Deadline  $d_3$  die nächste ist. Anschließend wird  $P_2$  ausgeführt und zum Schluss  $P_1$ .

Falls alle Prozesse die gleiche Bereitzeit besitzen und es sich um ein Einprozessorsystem handelt, ist EDF optimal. Optimal bedeutet, dass ein zulässiger Plan gefunden wird, falls einer existiert. Haben die Prozesse unterschiedliche Bereitzeiten  $r_i$ , so kann EDF versagen. Handelt es sich um ein Mehrprozessorsystem, so ist EDF nicht optimal, egal ob eine präemptive oder nicht präemptive Strategie verwendet wird (siehe 3.1.1 *Beispiel 1*).

Die zweite untersuchte Scheduling-Strategie heißt Least Slack Time (LST). Dabei handelt es sich ebenfalls um eine online und nicht präemptive Strategie. Der Plan in dieser Strategie wird nach den Spielräumen erstellt, wobei der Prozess mit dem kleinsten Spielraum die Ressource erhält. Der Spielraum ist die verbleibende Zeit, die einem Prozess bis zu seiner Frist übrig bleibt, nachdem er ausgeführt wurde. Berechnen lässt sich der Spielraum eines Prozesses mit folgender Gleichung:

$$sl_i = d_i - (t + e_i)$$

Dabei bleibt der Spielraum für den aktuell ausgeführten Prozess konstant, wohingegen alle Spielräume der anderen Prozesse abnehmen. Angewandt auf obiges Szenario würde sich folgende LST-Strategie ergeben:



$P_3$  wird zuerst ausgeführt, da dessen Spielraum  $sl_3$  mit 2 Einheiten am geringsten ist. Anschließend wird  $P_2$  ausgeführt, da der Spielraum  $sl_2$  0 Einheiten beträgt. Zum Schluss wird  $P_1$  ausgeführt.

Auf ein Einprozessorsystem kann im nicht präemptiven Fall LST selbst bei gleichen Bereitzeiten versagen. Handelt es sich um ein Mehrprozessorsystem, so ist LST nur dann optimal, falls alle Bereitzeitpunkte  $r_i$  gleich sind.

Vergleicht man beide Strategien, so erkennt man, dass LST Fristverletzung früher erkennt als EDF. Dafür müssen jedoch die Dauer der Prozesse bekannt sein.

Die beiden vorgestellten Scheduling-Strategien lassen sich auf die prinzipielle Aufgabenstellung dieser Masterarbeit übertragen. Hierzu können zu montierende Dächer als Prozesse und Arbeitsstationen als Prozessoren interpretiert werden. Montageprozesse können nicht unterbrochen werden, sodass eine nicht präemptive Situation vorliegt. Da mehrere Arbeitsstationen als Mehrprozessorsystem verstanden werden können, scheidet EDF aus, da dessen Strategie nicht optimal ist. LST kann prinzipiell im nicht präemptiven Fall ebenfalls versagen, jedoch stehen im Normalfall immer genügend weitere freie Arbeitsstationen zur Verfügung, sodass die Frist nicht überschritten wird.

Im folgenden werden ein paar Beispiele zu EDF und LST beschrieben, in denen die Strategien versagen. Die Beispiele werden mit zwei Prozessoren aufgeführt, jedoch lassen sich die gleichen Szenarien auf ein System mit mehr als zwei Prozessoren replizieren. Außerdem wird der Vergleich zu der gegebenen Situation in der MAN gezogen.

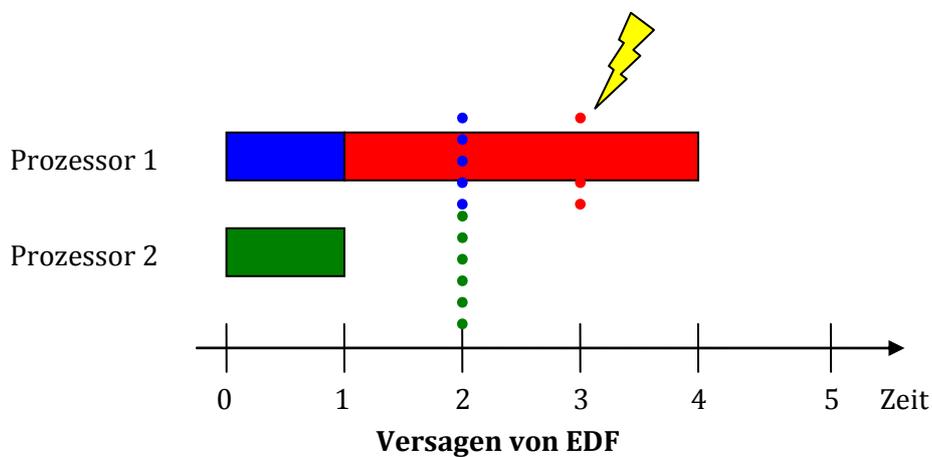
### 3.1.1 BEISPIEL 1

Im ersten Beispiel sollen drei Prozesse  $P_1$  mit gleichen Bereitzeiten  $r_i$  auf zwei Prozessoren ausgeführt werden. Obwohl ein Plan existiert, der von der LST-Strategie erkannt wird, versagt die EDF-Strategie.

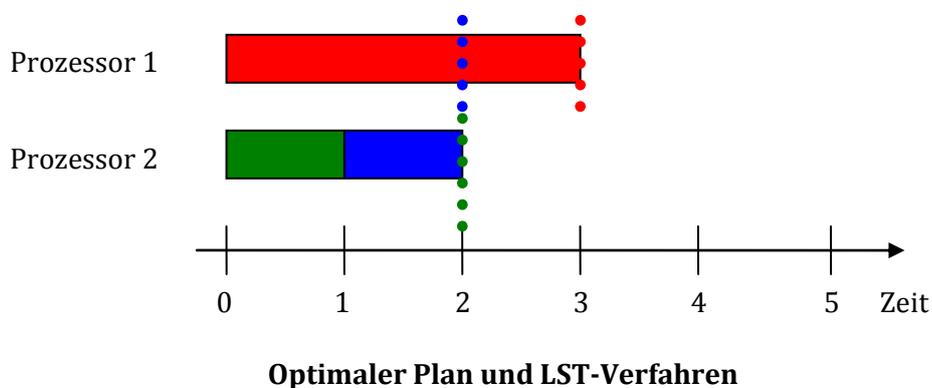
$P_1$ :  $r_1=0$ ;  $e_1=3$ ;  $d_1=3$ ;

$P_2$ :  $r_2=0$ ;  $e_2=1$ ;  $d_2=2$ ;

$P_3$ :  $r_3=0$ ;  $e_3=1$ ;  $d_3=2$ ;



Da  $d_2$  und  $d_3$  die nächsten Fristen sind, werden  $P_2$  und  $P_3$  zuerst auf den beiden Prozessoren ausgeführt. Erst im Anschluss kann  $P_1$  ausgeführt werden, was jedoch zu eine Überschreitung von  $d_1$  führt.



Die Berechnung der Spielräume ergibt  $sl_1=0$ , sodass  $P_1$  auf dem ersten Prozessor zuerst ausgeführt wird.  $P_2$  und  $P_3$  werden auf dem zweiten Prozessor nacheinander ausgeführt.

Dieses Beispiel ließe sich auf den Fall übertragen in dem drei Dächer gleichzeitig angeliefert werden. Die Konvention müsste wie folgt angepasst werden:

$P_i$  = Dach  $i$

$r_i$  = Zeitpunkt, an dem das Dach  $i$  geliefert steht (ready)

$d_i$  = Zeitpunkt, an dem das Dach  $i$  spätestens montiert sein sollte (deadline)

$t$  = verbleibende Montagezeit einer Arbeitsstation

$e_i$  = Montagedauer für Dach  $i$  (execution)

$sl_i$  = Spielraum des Dachs  $i$  zu dessen Deadline (slack time)

Ein solcher Fall ist jedoch technisch nicht möglich und wird folglich nicht näher betrachtet.

### 3.1.2 BEISPIEL 2

In diesem Beispiel sollen fünf Prozesse  $P_i$  mit unterschiedlichen Bereitzeiten  $r_i$  auf zwei Prozessoren ausgeführt werden. Sowohl die EDF- wie auch die LST-Strategie versagen, obwohl ein gültiger Plan existiert.

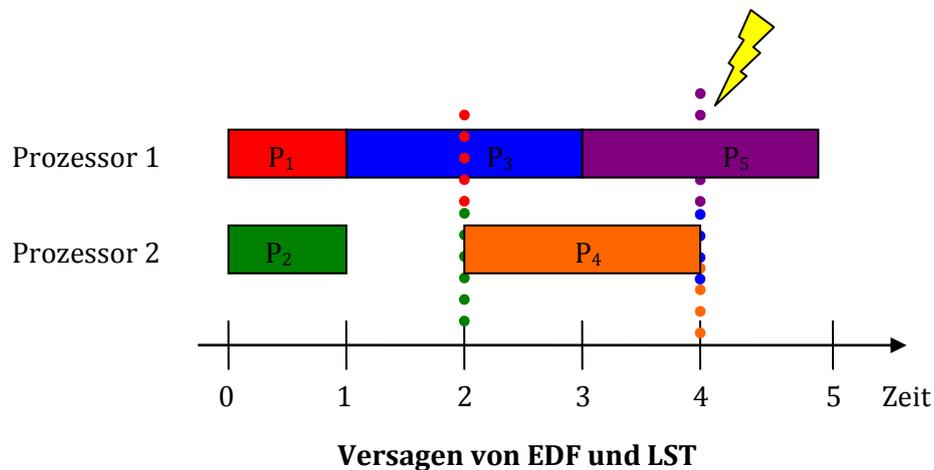
$P_1$ :  $r_1=0$ ;  $e_1=1$ ;  $d_1=2$ ;

$P_2$ :  $r_2=0$ ;  $e_2=1$ ;  $d_2=2$ ;

$P_3$ :  $r_3=0$ ;  $e_3=2$ ;  $d_3=4$ ;

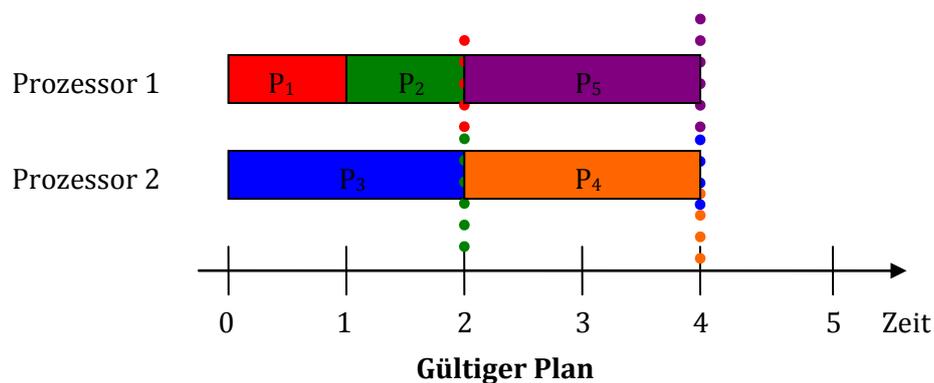
$P_4$ :  $r_4=2$ ;  $e_4=2$ ;  $d_4=4$ ;

$P_5$ :  $r_5=2$ ,  $e_5=2$ ;  $d_5=4$ ;



*EDF-Beschreibung:*  $P_1$  wird auf Prozessor 1 zuerst ausgeführt, da  $d_1=2$ .  $P_2$  wird auf Prozessor 2 ausgeführt, da  $d_2=2$ . Zum Zeitpunkt 1 wird  $P_3$  auf Prozessor 1 ausgeführt. Prozessor 2 wird nicht genutzt, da kein weiterer Prozess bereit steht. Zum Zeitpunkt 2 wird  $P_4$  mit  $d_4=4$  auf Prozessor 2 ausgeführt. Nachdem  $P_3$  auf Prozessor 1 ausgeführt wurde, wird zum Zeitpunkt 3  $P_5$  ausgeführt, jedoch wird die Deadline  $d_5=4$  verpasst.

*LST-Beschreibung:*  $P_1$  wird auf Prozessor 1 zuerst ausgeführt, da  $sl_1=1$ .  $P_2$  wird auf Prozessor 2 ausgeführt, da  $sl_2=1$ . Zum Zeitpunkt 0 ist  $sl_3=2$ . Zum Zeitpunkt 1 wird  $P_3$  auf Prozessor 1 ausgeführt. Prozessor 2 wird nicht genutzt, da kein weiterer Prozess bereit steht. Zum Zeitpunkt 2 wird  $P_4$  mit  $sl_4=0$  auf Prozessor 2 ausgeführt. Bereits zu diesem Zeitpunkt wird eine Fristverletzung deutlich, da  $sl_5=0$ . Nachdem  $P_3$  auf Prozessor 1 ausgeführt wurde, wird zum Zeitpunkt 3  $P_5$  ausgeführt, jedoch wird die Deadline  $d_5$  verpasst.



Es existiert ein gültiger Plan. Zum Finden des gültigen Plans müsste der Suchbaum mit NP-Aufwand abgearbeitet oder eine geeignete Heuristik gefunden werden, was sehr aufwendig ist [2].

Dieses Beispiel ist als Szenario mit fünf Dächern und zwei Arbeitsstationen durchaus denkbar. Jedoch stehen, wie bereits erwähnt, im Normalfall immer genügend weitere freie Arbeitsstationen zur Verfügung, sodass die Frist nicht überschritten wird.

### 3.2 PUSH-/ PULL-BASIERTE ARCHITEKTUR

Push und Pull beschreiben zwei verschiedenen Arten, wie Informationen in einer Systemarchitektur übertragen werden. In einer Push-basierten Architektur werden Änderungen von der Quelle an die Senken geschickt, wohingegen in einer Pull-basierten Architektur die Senken bei den Quellen Änderungen abfragen und laden. Beide Herangehensweisen haben Vor- und Nachteile und können miteinander kombiniert werden. Der Vergleich beider Architekturen wird im folgenden auf ein datenbankbasiertes Client/Server-System geführt.

In Push-basierten Architekturen werden Benutzereingaben der dezentralen Komponenten auf dem zentralen Server ausgeführt. Der Server ist dafür zuständig, die Business-Logik der Applikation abzuwickeln. Somit werden die Ressourcen des Servers und nicht der Clients beansprucht. Außerdem bleiben dem Client Tabellen- und Spaltennamen verborgen, was die Sicherheit deutlich erhöht. Hinzu kommt, dass durch den Einsatz einer solchen Architektur der Client/Server-Traffic immens gesenkt wird, da die Clients eine Nachricht bei Änderung von Daten erhalten. Weniger Daten werden somit zwischen dem Client und Server ausgetauscht weil nur bei Bedarf gesendet wird. Nachteile einer Push-basierten Architektur sind der erhöhte Aufwand zum Einrichten einer Relation und die fehlende Übertragungssicherheit. Des Weiteren würde ein Client mit fehlerhafter Verbindung Nachrichten nicht empfangen, ohne dass es der Server, der in diesem Fall der Sender ist, bemerkt. Als Beispiele, in denen Push verwendet wird, können RPC bzw. .Net Remoting für Interprozesskommunikation genannt werden. Außerdem verfolgen Stored Procedures in Datenbankmanagementsystemen dieselbe Idee und ermöglichen das Abarbeiten mehrerer zuvor hinterlegten SQL-Anweisungen.

In Pull-basierten Architekturen hingegen werden Informationen von den Clients aktiv am Server angefragt. Werden die Daten beispielsweise in Form einer Datenbank angeboten, so kann jeder Client in beliebigen Intervallen und ohne vorherige Einrichtung einer Relation darauf zugreifen. Dies führt aber zugleich zu einem Skalierungsproblem, da ausgiebiges Polling die Serverressourcen stark beanspruchen kann. Zur regelmäßigen Aktualisierung von Daten muss deswegen das Polling-Intervall adäquat angepasst werden, um nicht ein busy-waiting-ähnliches Verhalten zu produzieren. Falls die Verbindung zum Client unterbrochen wird, bevor er die Antwort erhalten hat, kann die Anfrage einfach neu verschickt werden. Somit wird eine Form der Übertragungssicherheit gewährleistet.

## 4 PROZESSBESCHREIBUNG

Zu Beginn muss erwähnt werden, dass die Dachmontage vom Fahrerhausband abgekoppelt ist, d.h. die Dachausstattung befindet sich auf einem Podest über dem Fahrerhausband und richtet sich nicht nach den Taktvorgaben. Ein Takt ist definiert mit 03:45 Minuten was bedeutet, dass ein Fahrerhaus sich in diesem Zeitintervall auf dem Gurtband (=Stauband) von einer Station zur nächsten bewegt. Die Dachausstattung ist eine Standmontage, die aus zehn Arbeitstationen, sogenannten Arbeitsbühnen besteht. Jede Bühne kann außerdem in einen Puffer- und Montageplatz unterteilt und je nach Mitarbeiterplan mit einem oder zwei Werkern betrieben werden. Die Perlenkette des Fahrerhausbandes wird bei der Dachausstattung erst am Gurtband wieder hergestellt. Als Perlenkette kann eine festgelegte Auftrags- und Fertigungsreihenfolge verstanden werden, die ab dem Zeitpunkt der Planung bis zur Produktfertigung unverändert bleiben sollte.

### 4.1 IST-ANALYSE

Eine Analyse der Ist-Situation bildet die Basis für das Aufdecken von Schwachstellen und die Ermittlung von Verbesserungspotentialen. Die Ist-Analyse dient zur Erarbeitung des Soll-Zustandes als Ausgangsbasis und kann ebenfalls als Checkliste für die Abarbeitung von Mängeln benutzt werden. Außerdem gibt sie Mitarbeitern und Managern einen Überblick über den bestehenden Zustand. Von großer Bedeutung ist die Einbeziehung der Mitarbeiter, da nur sie die tatsächlichen Gegebenheiten des Prozesses kennen. Die durch die Ist-Analyse entdeckten Schwachstellen können als Argumentationsgrundlage gegenüber Mitarbeitern in Führungspositionen für künftige Änderungen eingesetzt werden.

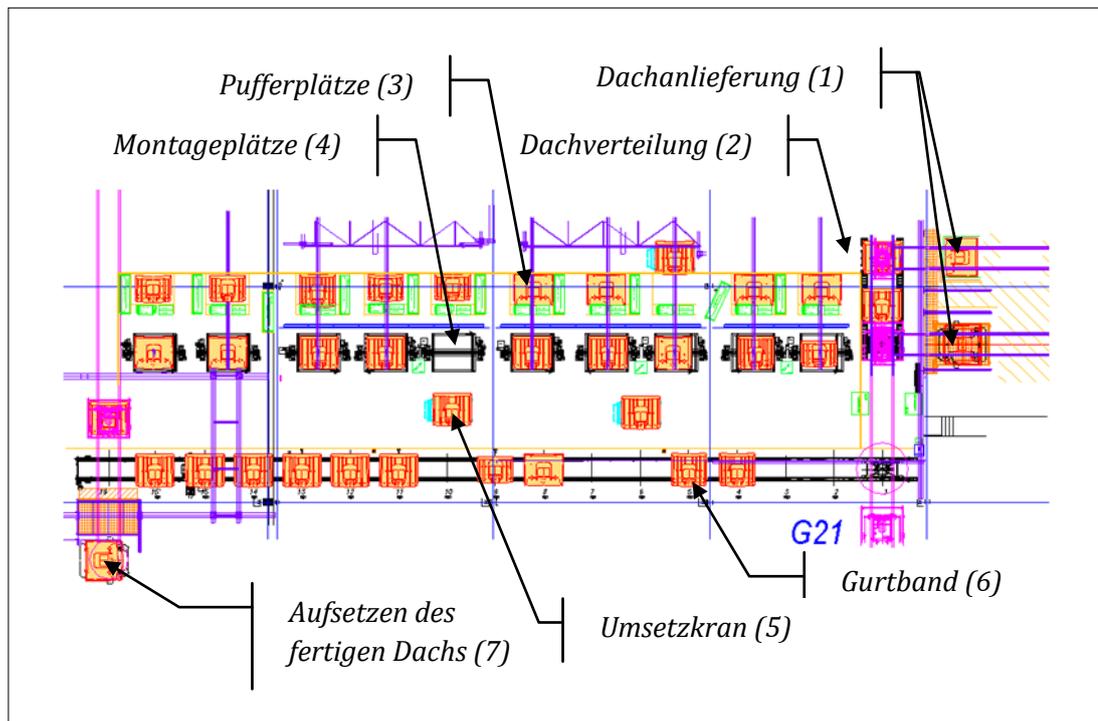
Zur Bewertung der Ist-Erfassung müssen Ziele festgehalten werden. Im Rahmen dieser Masterarbeit wurden nur funktionale Ziele betrachtet wie:

- Reduzierung der Durchlaufzeit
- Reduzierung von Stillstandzeiten
- Minimierung von Montagefehlern
- Erhöhung der Produktqualität

Diese Ziele können mithilfe einer Ablaufanalyse erreicht werden, indem hauptsächlich folgende Maßnahmen unternommen werden:

- Optimierung von inner- und zwischenbetrieblichen Prozessschnittstellen
- Identifikation und womöglich Rationalisierung überflüssiger Prozesse [3]

Der im Rahmen dieser Masterarbeit zu analysierende aktuelle Prozessfluss kann wie folgt beschrieben werden:



**Abbildung 1 - Layout für die Dachausstattung**

An Position 1 in *Abbildung 1* werden sowohl Blechdächer wie auch Montagedächer angeliefert, die sich prozesstechnisch durch unterschiedliche Ausstattungsmerkmale und dementsprechend durch unterschiedliche Montagezeiten voneinander differenzieren.

Der Verteiler (Position 2) schiebt das Dach auf einen Wagen und bestimmt mittels Abschätzung des Zeitaufwands für die Montage und eines manuell geführten Protokolls die Zuordnung des Dachs zu einer Bühne. Nachdem er die zu beliefernde Bühne ermittelt hat, wird das Dach in den dazugehörigen Pufferplatz (Position 3) gehängt.

Sobald der Montageplatz (Position 4) frei ist, wird das auszustattende Dach vom Pufferplatz in den Montageplatz befördert. Außerdem wird der dazugehörige Kommissionierwagen, der die in das Dach zu verbauenden Elemente beinhaltet, geholt, sodass mit der Montage des Dachs begonnen werden kann. Je nach Ausstattung, Mitarbeiterbelegung und Dachtyp kann die Montage eines Dachs zwischen 20 und 40

Minuten dauern [4]. Die Kommissionierwägen werden just-in-time von anderen Mitarbeitern mithilfe eines weiteren datenbankbasierten Systems (*Pick-To-Light*) gefüllt und, mit der Fahrzeugnummer gekennzeichnet, zum Abholen bereitgestellt.

Nach der vollendeten Montage wird das Dach mithilfe eines Umsetzkran (Position 5), der von einem Kranfahrer bedient wird, aus dem Montageplatz entnommen und auf dem Gurtband (Position 6) abgelegt. Beim Ablegen ist darauf zu achten, dass die Perlenkette wieder aufgebaut wird, was durch LED-Anzeigen mit den jeweiligen Fahrzeugnummern realisiert wird. Das sich in der Taktzeit bewegende Gurtband befördert die fertig montierten Dächer zum Aufsetzkran (Position 7), der diese auf die jeweiligen Fahrerhäuser auf der Ringbahn zum Kleben aufsetzt.

#### 4.1.1 STÖRUNGSURSACHEN

Dieser Prozessfluss ist in sich stimmig, jedoch hängt er von vielen Faktoren ab, die einen reibungslosen Ablauf stören können. Zu diesen kann z.B. die Sperrung eines Fahrerhauses in der Lackierung, die der Dachausstattung zeitlich vorgeschaltet ist, führen. Hierüber werden die Mitarbeiter in der Dachausstattung nicht informiert, sodass sie beispielsweise ein Hochdach ausstatten, dieses jedoch anschließend nicht auf, sondern vor dem Gurtband am Boden zwischenlagern müssen, da sich das zugehörige Fahrerhaus noch nicht auf der Ringbahn befindet. Erst nachdem das Fahrerhaus die Lackierung verlassen hat und vormontiert wurde, kann der Umsetzkranfahrer das bereits fertig montierte Dach vom Boden aufnehmen und auf die entsprechende Stelle auf dem Gurtband absetzen. Dieses mehrfache Auf- und Ablegen ist sehr zeitintensiv, da der Umsetzkran aus sicherheitstechnischen Gründen nur mit einer geringen Maximalgeschwindigkeit bedient werden kann.

Als weitere Störungsursache kann die Bühnenuzuweisung durch den Verteiler gesehen werden. Dieser schätzt die Montagezeit anhand des Modells und der Montierlöcher grob ab und versucht die Arbeitsbühnen abwechselnd mit Blech- und Hochdächern zu beliefern. Hintergrund dieser Alternation ist die Vorgabe, dass jede Arbeitsbühne in Abhängigkeit der eingesetzten Werker am Schichtende eine gewisse Soll-Stückzahl erreichen soll. Da jedoch, bedingt durch eine höhere Ausstattung, die Montage eines Hochdachs deutlich zeitintensiver ist, sind die Werker darauf bedacht, so viele Blechdächer wie möglich zu montieren. Diesem Wunsch verleihen sie nicht selten lautstark Ausdruck.

Generell ist der menschliche Faktor nicht zu vernachlässigen: bedingt durch persönliche Antipathien die oft auf ethnische Zugehörigkeiten zurückzuführen sind, konnten immer wieder nicht nachvollziehbare Zuordnungen des Verteilers beobachtet werden. Auf Nachfrage beim Verteiler wurde das Argument gebracht, dass dieser den Mitarbeiter "nicht so sehr möge" und vor ein paar Tagen ebenfalls mehrere

Hochdächer hintereinander erhalten hätte. Hierdurch kann die unterschiedlich hohe Kollegialität und Teamzugehörigkeit zwischen den Werkern erklärt werden. Außerdem wurden im Laufe der Analyse zwei Arbeitsweisen deutlich. Zum einen konnten kontinuierlich arbeitende Werker beobachtet werden, zum anderen solche, die teilweise sehr schnell und teilweise sehr langsam arbeiteten. Ein teilweise eingeführtes Springerprinzip, bei dem an zwei Bühnen je ein Arbeiter arbeitet und ein dritter ja nach Bedarf an einer Bühne aushilft, konnte die Effektivität ebenfalls nicht steigern, da aus oben genannten Gründen immer wieder zwischenmenschliche Spannungen auftraten.

#### **4.1.2 ZUWEISUNG DER DÄCHER ZU DEN ARBEITSBÜHNEN**

An Position 2 in *Abbildung 1* auf Seite 17 werden die Hochdächer wie auch Blechdächer nach keiner definierten Reihenfolge auf den Verteilerwagen geladen. Das bedeutet, dass bei gleichzeitiger Anlieferung zweier Dächer verschiedenen Typs der Verteiler nicht ahnen kann, welches der beiden Dächer zuerst montiert werden soll. Erst nachdem ein Dach auf dem Verteilerwagen gelegt wurde, ermittelt der Verteiler die nächste zu beliefernde Arbeitsbühne. Wie zuvor beschrieben, schätzt er hierzu den Arbeitsaufwand anhand des Dachtyps und der Montierlöcher ab und versucht zusätzlich eine abwechselnde Zuteilung von Blech- und Hochdächern zu gewährleisten. Die Gewährleistung der Alternation wird durch ein manuell geführtes Protokoll unterstützt. Steht die zu beliefernde Arbeitsbühne fest, schiebt der Verteiler das Dach bis zu dem zugehörigem Pufferplatz und fixiert das Dach mithilfe eines Vakuumsaugers in eine Hebevorrichtung. Nach der Zuteilung notiert er die Fahrzeugnummer des Dachs in einem manuell geführten Protokoll und fängt von neuem mit der Zuweisung an.

#### **4.1.3 AUSSTATTUNG DER DÄCHER IN DEN ARBEITSBÜHNEN**

Die Bühnenmitarbeiter werden vor Schichtbeginn durch den Schichtleiter auf ihre Arbeitsbühnen zugewiesen wobei je nach Zuteilung Zweier- und Einzelteams entstehen. Hierbei ist zu beachten, dass eine Doppelbühne nicht so produktiv ist wie zwei Einzelbühnen. Dies wird am Schichtende besonders in der Stückzahl deutlich: eine Einzelbühne wickelt in einer Schicht durchschnittlich zehn Dächer ab, wohingegen eine Doppelbühne nur 17 Dächer montiert. Dieser deutliche Unterschied ist darauf zurückzuführen, dass sich zwei Mitarbeiter bei der Montage bei bestimmten Handgriffen gegenseitig behindern und somit ausbremsen.

Den ersten Schritt, den die Bühnenmitarbeiter eines Zweiertams zur Ausstattung der Dächer machen müssen, ist den zugehörigen Kommissionierwagen zu holen. Dieser

beinhaltet alle zu verbauende Elemente wie Kabel, Antennen, Blenden usw. Währenddessen befördert der an der Arbeitsbühne bleibende Mitarbeiter das zu montierende Dach vom Pufferplatz mithilfe der vakuumbetriebenen Hebevorrichtung in den Montageplatz, um es dort mit zwei Gurten zu fixieren. Die Fixierung ist von Nöten, da das Dach um die Querachse gekippt und angehoben werden kann, um eine ergonomische Arbeitsweise zu ermöglichen. Nach adäquater Positionierung des Dachs beginnt der Einbau der Ausstattung, der je nach Dachtyp und Mitarbeiteranzahl zwischen ca. 20 Minuten (2 Mitarbeiter, Blechdach) bis zu ca. 80 Minuten (1 Mitarbeiter, Hochdach mit Vollausrüstung) dauern kann. Nach Vollendung der Ausstattung wird das Dach in die Ausgangsposition zurückgebracht, die Gurte gelöst und darauf gewartet, dass das Dach vom Umsetzkran aus dem Montageplatz entnommen wird. Steht bereits das nächste auszustattende Dach im Pufferplatz, kann der leere Kommissionierwagen währenddessen zurückgebracht und zugleich der neue zur Arbeitsbühne mitgebracht werden. Daraufhin wiederholt sich der Ausstattungsprozess an der Arbeitsbühne.

#### **4.1.4 UMSETZEN DER DÄCHER AUF DAS GURT BAND**

Der Umsetzkran fungiert als Bindeglied zwischen den Arbeitsbühnen und dem Gurtband. Jedes Mal, wenn ein fertig montiertes Dach am Montageplatz in die Ausgangsposition gebracht wird, kann der Kranfahrer den Umsetzkran über das Dach befördern und jenes mithilfe eines Vakuumsystems heraus heben. Dies passiert im Durchschnitt bei Hochdächern alle 4:03 min und bei Blechdächern alle 3:09 min [4]. Anschließend muss überprüft werden, ob die Fahrzeugnummer des Dachs an den LED-Anzeigen über dem Gurtband angezeigt wird. Ist dies der Fall, so setzt der Kranfahrer das Dach an die entsprechende Position auf dem Gurtband ab. Falls die Fahrzeugnummer noch nicht angezeigt wird, weil beispielsweise ein Hochdach zu früh ausgestattet wurde, so wird das Dach vor dem Gurtband am Boden abgelassen, um es zu einem späteren Zeitpunkt an die richtige Position zu befördern. Nach diesem Schritt widmet sich der Kranfahrer dem nächsten aus dem Montageplatz zu entnehmendem Dach.

#### **4.1.5 SCHWACHSTELLENANALYSE**

In einer Expertenrunde, an der Bühnen-Mitarbeiter sowie der Sparten-, Schicht- und Abteilungsleiter teilnahmen, wurde versucht die Ursachen für die langen Wartezeiten zu finden.

Die Wartezeiten lassen sich in drei Hauptkategorien unterteilen:

- Wartezeiten bedingt durch den Umsetzkran
- Wartezeiten bedingt durch den Verteiler
- Wartezeiten bedingt durch die Vorkommissionierung

Die für diese Arbeit relevanten Wartezeiten beschränken sich jedoch auf jene, die durch den Umsetzkran, dem Prozess und vor allem durch den Verteiler verursacht werden (siehe *Abbildung 2 - Ishikawa-Diagramm für Schwachstellenursachen*).

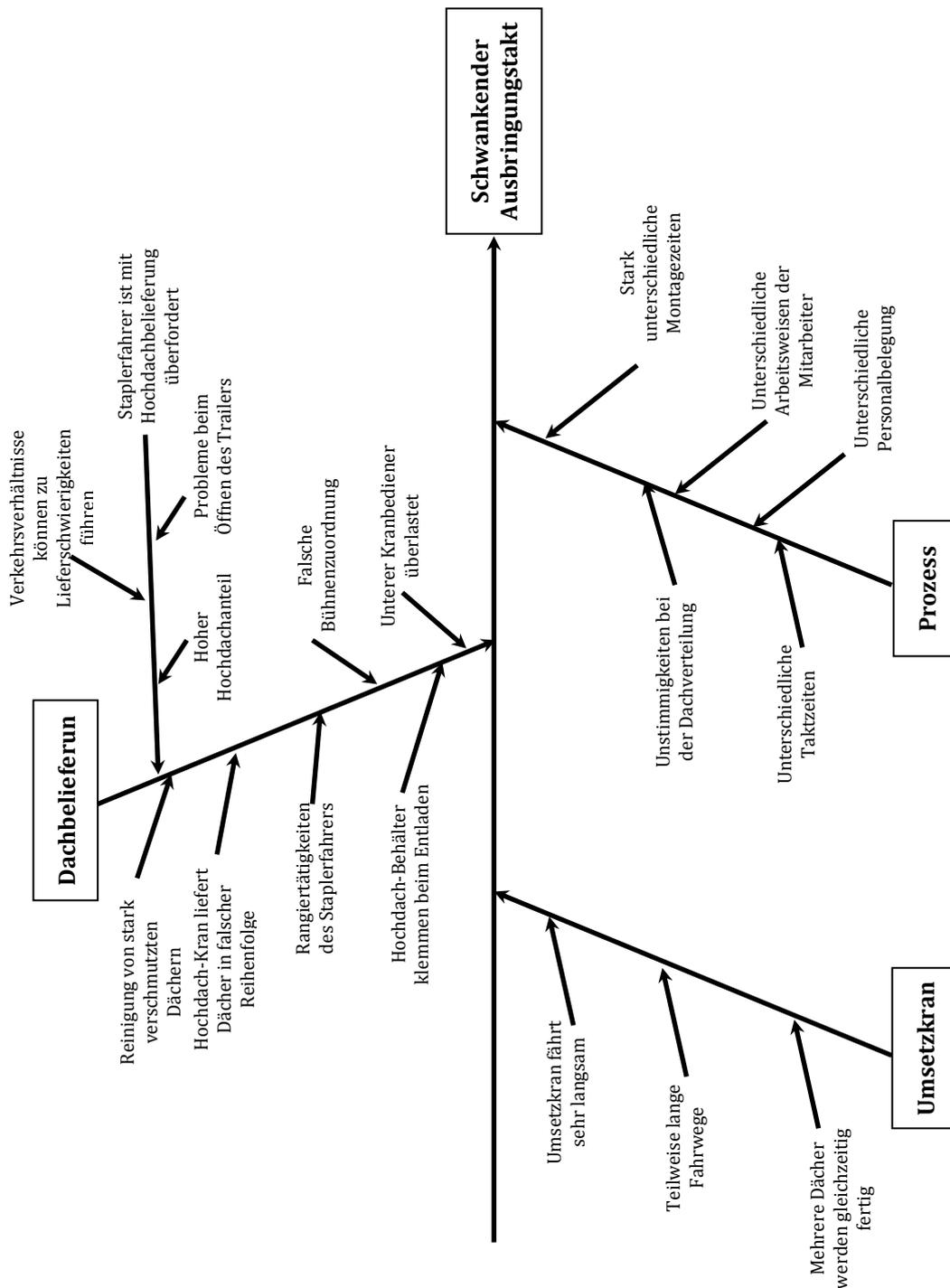


Abbildung 2 - Ishikawa-Diagramm für Schwachstellenursachen [4]

Erstere variieren zwischen null und maximal zwölf Minuten, was eine durchschnittliche Wartezeit von 1:52 min ergibt. Etwa 26% der Wartezeiten bedingt durch den Umsetzkran liegen über der Taktzeit. Insgesamt wartet ein Bühnen-Mitarbeiter während 4:45 h Produktivzeit 18,4 Minuten auf die Entnahme des fertig montierten Dachs aus dem Montageplatz. Als Hauptursachen für die durch den Umsetzkran bedingte Wartezeiten sind aufzuführen:

- Die Beendigung der Montage mehrerer Dächer fallen zeitlich nah aneinander, sodass der Kranfahrer mit dem Umsetzen der Dächer temporär überfordert ist
- Die langen Fahrwege
- Die geringe Geschwindigkeit des Umsetzkranes

Ähnlich wie beim Umsetzkran, verhält es sich mit der Wartezeit auf ein neues Dach. Diese Wartezeit ist definiert als die Zeit, die ein Bühnen-Mitarbeiter untätig ist, da die Montage des vorhandenen Dachs beendet ist und kein neues, zu montierendes Dach auf dem Pufferplatz zur Verfügung steht. Bezüglich dieser Wartezeiten kann man erwähnen, dass sie sich zwischen null Minuten (Dach ist im Pufferplatz vorhanden) und 22 Minuten verteilen. 70% der Wartezeiten befinden sich unter der Taktzeit von 3:45 min, was noch schlechter ist als die durch den Umsetzkran bedingte Wartezeiten. Mit einer durchschnittlichen Wartezeit von 2:27 min, erhält man während einem Produktivzeitraum von 4:45h eine aufsummierte Wartezeit bedingt durch den Verteiler von 24,5 Minuten. Hierzu können unter anderem folgende Ursachen genannt werden [4]:

- Die falsche Zuordnung des auszustattenden Dachs durch den Verteiler. Dieser kann als Zuordnungshinweis nur das manuell geführtes Protokoll, den Dachtyp, wie auch die Anzahl der Montagelöcher im Dach verwenden. Außerdem führen zwischenmenschliche Spannungen ebenfalls zu ineffizienten Bühnenzuweisungen.
- Die Anlieferung der Hochdächer verspätet sich beispielsweise aufgrund von schlechten Verkehrsverhältnissen oder Wetterbedingungen, aber auch aufgrund von Problemen beim Öffnen des Trailers.
- Stark verschmutzte Hochdächer müssen vor der Montage manuell gereinigt werden.
- Transportbehälter der Hochdächer klemmen beim Entnehmen des Dachs.

In Summe kann man in dem Beobachtungszeitraum von einer Wartezeit von mindestens 48 Minuten pro Bühne ausgehen, wobei die durchschnittliche Wartezeiten bedingt durch die Vorkommissionierung von 3:53 min pro Bühne noch nicht berücksichtigt sind. Diese summierte Wartezeit gilt es durch eine softwaregestützte Prozessoptimierung zu minimieren.

## 4.2 SOLL-ZUSTAND

Die Erfassung der Sollsituation basiert auf der Analyse des Ist-Zustands. Hierbei sind die erkannten Mängel aus der Ist-Analyse beseitigt und der Prozess optimiert dargestellt. Um zum Soll-Zustand gelangen zu können, muss eine Akzeptanz für die Veränderungen bei den Mitarbeitern wie auch Führungskräften durch diverse Argumente erreicht werden:

- Verbesserung der Arbeitsabläufe
- Reduktion der Bearbeitungszeiten
- Reduktion der Stillstandzeiten
- Einsparung von Kosten

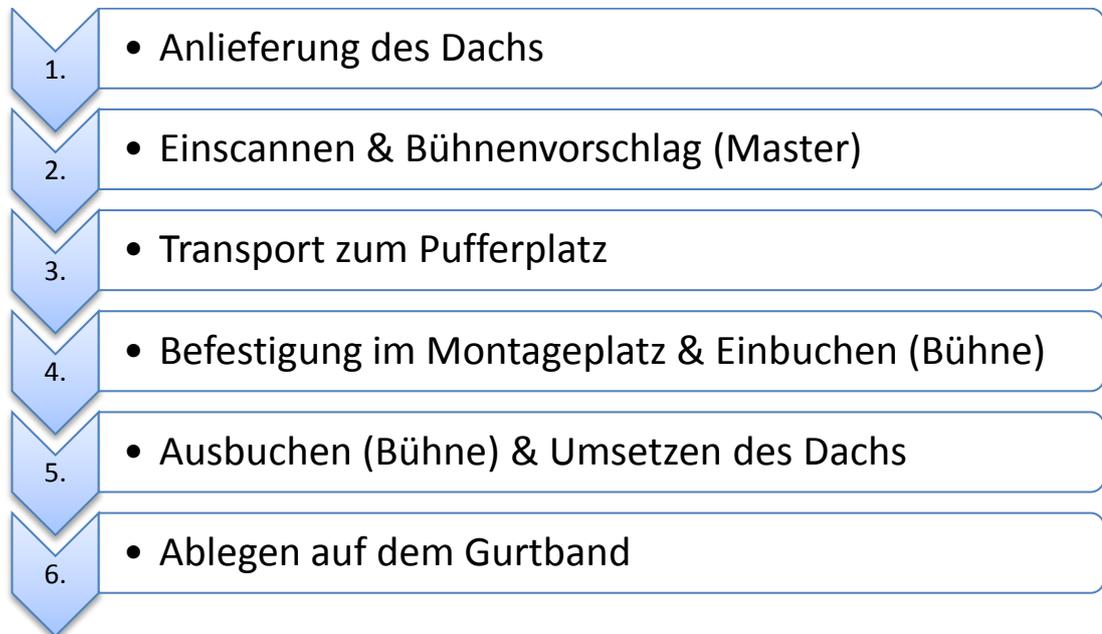
Vor der Umsetzung der Lösungen müssen zunächst kontrollierte Testläufe durchgeführt werden, sodass Optimierungsfehler bzw. Implementierungsfehler auf den Produktionsbetrieb nicht komplett durchschlagen. Außerdem ist darauf zu achten, dass die Mitarbeiter, die direkt am Prozess beteiligt sind, adäquat auf die Softwareoptimierung vorbereitet werden (beispielsweise durch Schulungen). Das Involvieren der Mitarbeiter ist ein wesentlicher Bestandteil der Verbesserungsmaßnahmen, da die Werker bei Nichtbeachtung oftmals negativ auf Veränderungen reagieren [4]. Die Einführung der neuen Software wurde folglich nacheinander in kleinen Schritten durchgeführt.

Im erwünschten Soll-Prozess soll die Zuweisung eines Dachs zu einer Arbeitsbühne mit Hilfe einer Software geschehen. Diese Software bestimmt auf Basis von Montagezeiten, die aus sogenannten Arbeitsplänen entspringen, welche Arbeitsbühne theoretisch als nächste frei wird. Der Bühne wird daraufhin das angelieferte Dach zugewiesen. Die Software soll also dem Verteiler die Entscheidung abnehmen, welche Bühne als nächstes beliefert werden soll. Damit ist der menschliche Sympathie/Antipathie-Faktor aus dem Prozess entfernt, was zur positiveren Einstellung der Werker führt.

Prinzipiell bleibt der Prozessfluss, wie in *4.1 Ist-Analyse* beschrieben, bestehen, wird jedoch an einigen Stellen erweitert. So muss der Mitarbeiter, der die Dächer bei der Anlieferung entgegennimmt und auf die Arbeitsbühnen verteilt, zusätzlich einen zum Dach zugehörigen Barcode mit der Fahrzeugnummer scannen. Hiermit werden die dachspezifischen Informationen geladen, eine optimale Arbeitsbühne berechnet und am Bildschirm ausgegeben, auf die das Dach transportiert werden soll. Die entsprechende Software läuft auf dem sogenannten Master-PC. Außerdem müssen die Montagemitarbeiter im Soll-Zustand eine grafische Oberfläche mit zwei Schaltflächen bedienen. Sobald sie ein auszustattendes Dach aus dem Pufferplatz in den Montageplatz ablegen, sollen die Werker diesen Schritt an der GUI dokumentieren. Die

Beendigung der Dachmontage soll ebenfalls an der GUI durch die Werker quittiert werden.

In *Abbildung 3* ist der Prozessfluss des Soll-Zustands im Überblick zu sehen:



**Abbildung 3 - Prozessfluss Soll-Zustand**

Die langfristige Zielsetzung besteht nicht in einer Mitarbeiter-rationalisierung bzw. Stückzahlerhöhung, sondern in einem ruhigeren Arbeitsablauf und einer daraus resultierenden geringeren Fehlerquote. Eine gerechte und regulierte Arbeitsverteilung soll im Vordergrund stehen, nicht die Erfüllung einer bestimmten Stückzahl.

## 5 KONZEPTE

Wie bei jeder Neukonzeptionierung existieren auch bei dieser Masterarbeit unterschiedliche Herangehensweisen. Jedes Konzept zieht seine eigenen Vor- und Nachteile nach sich, wobei diese in den folgenden Abschnitten genauer erläutert werden sollen. Zuvor werden jedoch die Anforderungen des Systems analysiert.

### 5.1 ANFORDERUNGEN

Um den in 4.2 *Soll-Zustand* geschilderten Prozessfluss realisieren zu können, müssen diverse Voraussetzungen geschaffen werden. Da es sich um ein verteiltes System handelt, müssen die einzelnen Systemabschnitte miteinander vernetzt sein, um kommunizieren zu können. Um aufwendige Verkabelungsarbeiten zu vermeiden, sollte dies über das bereits vorhandene MAN-Netzwerk via Ethernet realisiert werden. Fehlende Anbindungen können außerdem ohne großen Aufwand nachgerüstet werden. Die verwendete Hardware muss MAN-Richtlinien entsprechen und sollte kostengünstig sein.

Abgesehen von der Hardware, muss ebenfalls eine logische Anbindung an diversen Quellen geschaffen werden. Hierzu muss ein Zugang zu den dachspezifischen Daten wie Fahrzeugnummer, Montagezeiten, Dachtyp usw. vorliegen. Dies wird über den Zugriff auf eine MAN-Datenbank gewährleistet, die all diese Informationen zur Verfügung stellt. Eine adäquate Aufbereitung und Bereitstellung dieser Daten im DMS ist ebenfalls erforderlich.

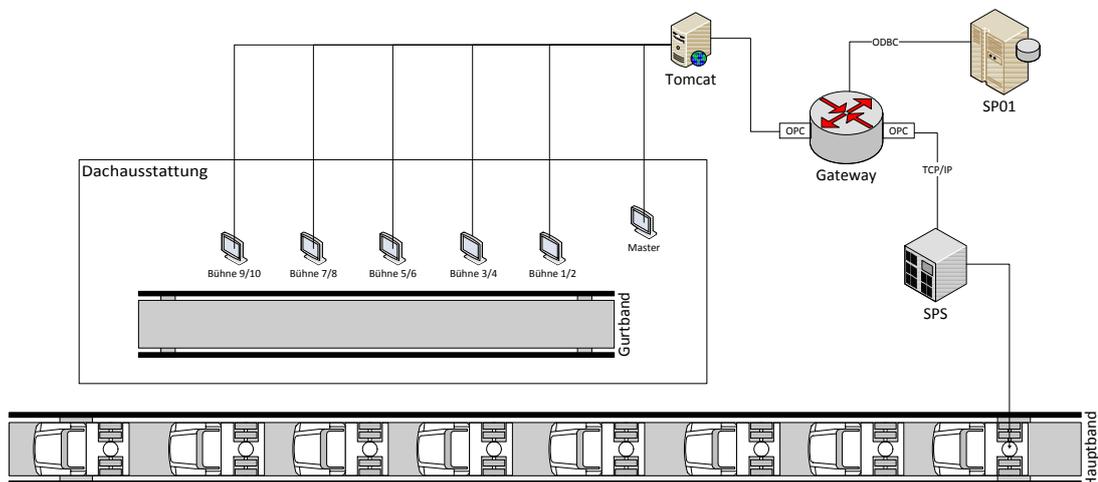
Neben den technischen Anforderungen, müssen auch die Mitarbeiter Anforderungen erfüllen. So muss eine gewisse Kontinuität in deren Arbeitsweise sichergestellt sein, egal ob es sich um Schichtbeginn oder Schichtende handelt und egal ob mehrere Hochdächer nacheinander an die selbe Arbeitsbühne zugewiesen werden. Außerdem muss Akzeptanz für das DMS bei den Werkern geweckt werden. Dies kann beispielsweise durch eine Schulung erreicht werden.

## 5.2 SPS-GESTEUERTES KONZEPT

Eine Speicherprogrammierbare Steuerung (SPS) ist ein Gerät, das zur Anlagensteuerung oder -regelung verwendet wird. Neben diversen Ein- und Ausgängen besitzt eine SPS außerdem eine Schnittstelle, über die sie programmiert werden kann.

Im MAN-Werk werden zahlreiche SPS eingesetzt, die die Fertigung automatisiert steuern. So wird beispielsweise die Position eines Dachs beim Durchlauf eines bestimmten Bandabschnitts in der Perlenkette aktualisiert. Dies wird über einen sogenannten Betriebsstellenfilter realisiert. Ein Betriebsstellenfilter kann als logischer Ausdruck verstanden werden, der bei bestimmten Ergebniswerten eine Aktion unternimmt. Als weiteres Beispiel kann die eingesetzte SPS erwähnt werden, die den Mitarbeitern in der Kommissionierung die Fahrzeugnummer mitteilen, deren Kommissionierwagen befüllt werden muss. Hierzu wurde auf dem Hauptband, einige Takte vor der Dachausstattung, ein Betriebsstellenfilter definiert, der einen Druckauftrag mit der Fahrzeugnummer an den Drucker in der Kommissionierung schickt.

In dem in *Abbildung 4 - SPS-gesteuerter Systemaufbau* gezeigtem SPS-gesteuertem Konzept soll ein Betriebsstellenfilter ähnlich wie für die Kommissionierung definiert werden. Zusätzlich zur SPS muss ein sogenanntes Gateway eingesetzt werden.



**Abbildung 4 - SPS-gesteuerter Systemaufbau**

Das Gateway ist ein Rechner auf dem ein OPC Server installiert ist. Der OPC Server stellt die Verbindung zur SPS über TCP/IP her und übersetzt die Daten in einem standardisiertem OPC-Format. Diese Daten können anschließend von einem Tomcat-

Server ausgelesen werden. Damit das Gateway über das Vorliegen neuer Daten in der SPS informiert wird, muss von dieser ein externes Impulssignal gesendet werden.

Neben den SPS-Informationen kann das Gateway außerdem alle weiteren fahrzeugspezifischen Daten wie Montagezeiten über eine ODBC-Verbindung aus der MAN-Datenbank auslesen. Auch diese Informationen werden für den Tomcat-Server im OPC-Format bereitgestellt. Letzterer ist für eine adäquate Datenaufbereitung zuständig, da die grafischen Oberflächen der Arbeitsstationen webbasiert sind. In diesem Konzept konzentriert sich die Systemlogik auf das Gateway sodass eine Konzeptionierung nach dem MVC-Modell gut umsetzbar ist. Dabei könnten die SPS und die Oracle-DB als Model, die einzelnen Weboberflächen an den Arbeitsstationen als Views und das Gateway und der Tomcat-Server als Controller verstanden werden.

Dieses SPS-gesteuerte Konzept ist für die Grundanforderung an das zu entwickelnde System ausreichend. Jedoch sind die Ausbaumöglichkeiten stark beschränkt. So müsste neben dem Gateway noch eine weitere Datenbank laufen, um den weiteren Anforderungen an das System (Log-Auswertung, Schnittstelle zu weiteren MAN-Komponenten...) gerecht zu werden. In diesem Fall kann jedoch das komplette System datenbankbasiert konzipiert werden, worauf im nächsten Abschnitt genauer eingegangen werden soll.

### **5.3 DATENBANKBASIERTES KONZEPT**

Im datenbankbasierten Konzept soll eine Datenbank als Grundlage der Konzeptionierung dienen. Darauf wird eine Client/Server-Architektur aufgesetzt. Die Clients können entweder webbasiert oder auch als Applikation entworfen sein.

Neben den Standardfunktionen einer Datenbank wie Datensicherheit, Datenintegrität, Anwendungsunterstützung und Mehrbenutzerfähigkeit besteht ein Vorteil des datenbankbasierten Konzepts vor allem in der freien Modellierung des Datenbankschemas [5]. Hierdurch lassen sich auch komplexe Anforderungen adäquat im Schema abbilden. Außerdem kann in diesem Konzept das MVC-Modell eingesetzt werden, indem die Datenbank als Model, die verteilten Applikationen als View und ein Hauptprogramm, das alle Aktionen durchführt, als Controller interpretiert werden. Weiterhin kann in diesem Konzept auf standardisierte Software und Literatur zurückgegriffen werden. Die Softwaresammlung XAMPP bietet unter anderem einen Web- und Datenbankserver, welche sich leicht installieren und konfigurieren lassen.

Im Gegensatz zum SPS-gesteuerten Konzept, ist es im datenbankbasierten Konzept möglich, Systemerweiterungen einfach umzusetzen. Hierzu müssen zwar das Schema der Datenbank und Abhängigkeiten darauf angepasst werden, jedoch können solche Änderungen von vornherein durch eine durchdachte Modellierung umgangen werden.

## 6 OPERATIONALISIERUNG DER THEORETISCHEN KONZEPTE

Nachdem die theoretischen Konzepte erläutert wurden, wird in diesem Abschnitt deren praktische Umsetzung beschrieben. Hierzu wird auf die Architektur des Systems, die Implementierung der einzelnen Komponenten und deren Interaktion wie auch auf den eingesetzten Algorithmus näher eingegangen.

Als Konzept wurde eine datenbankbasierte Client/Server-Architektur gewählt, um v.a. die Konsistenz der Daten sicherzustellen. Dadurch werden alle Informationen in einer MySQL-Datenbank abgelegt, die auf dem Master-PC läuft. Diese dient ebenfalls zur Kommunikation der einzelnen Komponenten untereinander die durch zyklisches Polling Änderungen am Datenbestand registrieren.

### 6.1 IMPLEMENTIERUNG DER KOMPONENTEN

Das Dach Management System (DMS) besteht aus vier Komponenten, welche über TCP/IP miteinander kommunizieren. Die Teilprogramme sind über einen Master-PC, fünf Bühnen-PCs und dem Schichtleiter-PC verteilt. Dabei ist hervorzuheben, dass es sich beim Master-PC um einen Industrie-Rechner mit gespiegelten Festplatten und redundantem Netzteil handelt, um die Ausfallsicherheit zu erhöhen. Zusätzlich wird in regelmäßigen Abständen ein komplettes Festplatten-Image automatisch erstellt, sodass im Notfall ein baugleicher Rechner die vom Master getätigten Funktionen zeitnah übernehmen könnte, da es sich hier um einen Single Point of Failure handelt. Diese Funktionen sind zum Einen das Bereitstellen einer grafischen Oberfläche für die Mitarbeiter (Master-GUI), zum Anderen die Interaktion mit der restlichen MAN-IT mittels eines Dienstes (Daemon) und v.a. die Bereitstellung der Daten mithilfe der MySQL-Datenbank.

Aufgrund des breiten Einsatzspektrums wurde *C#* als Programmiersprache gewählt und die einzelnen Komponenten in Visual Studio 2010 entwickelt. *C#* wird als „C-sharp“ ausgesprochen und entspricht damit im Englischen der Musiknote „Cis“, also der Erhöhung des Tons C um einen Halbton. Damit versinnbildlicht der Name „C#“ in ähnlicher Weise wie „C++“ eine Weiterentwicklung der Programmiersprache C [6]. Als Kernelement von *C#* ist wie in C der Block anzusehen, der mit geschweiften Klammern gekennzeichnet wird. Blöcke können beispielsweise Methoden oder auch bedingte Anweisungen nach einer *if*-Anweisung sein. Diese Blöcke bestimmen den Gültigkeitsbereich von Variablen, was bedeutet, dass Variablen, die innerhalb des Blocks definiert wurden auch nur innerhalb dieses bestimmten Blocks gültig sind. Folglich können gleiche Variablenbezeichnung in verschiedenen Blöcken problemlos

eingesetzt werden. Wie in *C* wird auch in *C#* eine Anweisung mithilfe eines Semikolons beendet und eine freie Formatierung des Codes ermöglicht. Referenzen hingegen werden mit dem Schlüsselwort *ref* gekennzeichnet, sodass keine Zeiger benötigt werden [6].

Da bereits an weiteren Bandabschnitten mit SPS-Konzepten gearbeitet wird (siehe 5.2 *SPS-gesteuert*), liegt die Frage nahe, weshalb das System nicht SPS-gesteuert und webbasiert aufgebaut werden sollte. Hierzu könnte als Skriptsprache Perl, PHP oder auch JavaScript eingesetzt werden. Dem ist zu entgegnen, dass es von vornherein absehbar war, dass die Werker im restriktivsten Fall ausschließlich deren zugewiesene grafische Oberfläche nutzen dürfen. Weitere Anwendungen sollten nicht zugänglich gemacht und unnötige Tastatureingaben ignoriert werden. Als Hauptargument ist jedoch aufzuführen, dass ursprünglich ein RPC-Konzept verfolgt wurde welches mit einer webbasierten Oberfläche technisch nicht umsetzbar gewesen wäre. Außerdem ist die Flexibilität und Erweiterbarkeit des SPS-gesteuerten Konzepts stark eingeschränkt. Folglich eignet sich eine datenbankbasierte .Net-Applikation hervorragend, da soeben genannte Restriktionen im Gegensatz zu einem webbasierten System sehr leicht umzusetzen sind.

Das RPC-Konzept konnte aus bürokratischen Gründen nicht umgesetzt werden; damit scheiterte auch die strikte Einhaltung des MVC-Modells was eine dezentrale Logik nach sich zog.

Der Einsatz eines vorgefertigten Frameworks war leider nicht möglich. Deshalb wurde ein Basisprojekt mit den in allen Komponenten benötigten Klassen als Framework erstellt. Hierzu zählen unter anderem die Klasse mit den SQL-Methoden, wie auch die Klasse für den Zugriff auf eine externe config.ini-Datei. Diese Datei ist im selben Ausführungspfad wie die Anwendung abgelegt und ermöglicht einfache Änderungen an lokalen Parametern des DMS. Dafür ist nur rudimentäres Wissen notwendig, sodass beispielsweise bei einer Neuinstallation eines Bühnen-PCs die Bühnennummer und der Zugriff auf die MySQL-Datenbank sehr einfach durch einen MAN-Mitarbeiter angepasst werden kann.

```
[DB]
; Zugriff auf MySQL-DB
database = mydatabase
username = myuser
password = mypa$$w0rd
ip = 192.168.100.123

[SYSTEM]
BuehnenNr = 3
```

**Abbildung 5 - config.ini einer Bühnen-GUI**

Eine detaillierte Beschreibung aller GUIs, die im folgenden beschrieben werden, finden sich im Anhang 11.2 *Benutzerhandbuch für DMS*.

### 6.1.1 MASTER

Die im Prozessfluss als erste genutzte GUI wird bei der Dachanlieferung und -verteilung genutzt (siehe *Abbildung 6 - Master-GUI*). Diese zeigt eine Übersicht der zehn Puffer- und Montagplätze an, wie auch deren Mitarbeiterbelegung. Außerdem kann im oberen Abschnitt mithilfe eines Barcodescanners die zum Dach zugehörige Fahrzeugnummer eingegeben werden, sodass daraufhin die optimale Arbeitsbühne berechnet wird. Der Barcode der Fahrzeugnummer liegt dem Mitarbeiter in Papierform vor und wird automatisch durch einen SPS-gesteuerten Betriebsstellenfilter ausgedruckt.

	Montageplatz	Pufferplatz	Anzahl MA
Bühne 1:			1
Bühne 2:			3
Bühne 3:		06X-1235	2
Bühne 4:		06S-5074	1
Bühne 5:	23U-2137		0
Bühne 6:	06X-T479		0
Bühne 7:	35K-3498		0
Bühne 8:	98D-2439		0
Bühne 9:			0
Bühne 10:		74L-4423	0

**Abbildung 6 - Master-GUI**

Nach dem Scannen der Fahrzeugnummer und der Berechnung der optimalen Arbeitsbühne wird das Dach automatisch auf dem vorgeschlagenen Pufferplatz angezeigt, sodass im Optimalfall keine weitere Interaktion seitens des Mitarbeiters benötigt wird.

Falls die Systemempfehlung jedoch revidiert werden sollte, kann man eine passwortgeschützte Maske öffnen in der die Zuteilung der Dächer manuell per Drag&Drop geändert werden kann. Ein Grund hierfür ist beispielsweise der Ausfall einer Arbeitsbühne. In dieser Maske besteht außerdem die Möglichkeit Dächer vom

Pufferplatz in den Montageplatz einzubuchen bzw. vom Montageplatz auszubuchen. Diese Funktion ist nötig, wenn ein Arbeitsbühnen-PC ausfallen sollte und somit das aktuell montierte Dach durch den Bühnenmitarbeiter nicht mehr ausgebucht werden kann. Außerdem besteht die Möglichkeit im passwortgeschütztem Bereich die Mitarbeiterbelegung der Arbeitsbühnen anzupassen, da diese von Schicht zu Schicht alterniert.

### 6.1.2 BÜHNE

Zwei Arbeitsbühnen teilen sich einen zwischen ihnen aufgestellten Rechner. Folglich wurde die Bühnen-GUI als Split-Screen realisiert, wobei sich die beiden Seitenabschnitte ausschließlich durch die angezeigte Bühnennummer unterscheiden, die in der Konfigurationsdatei definiert werden kann.

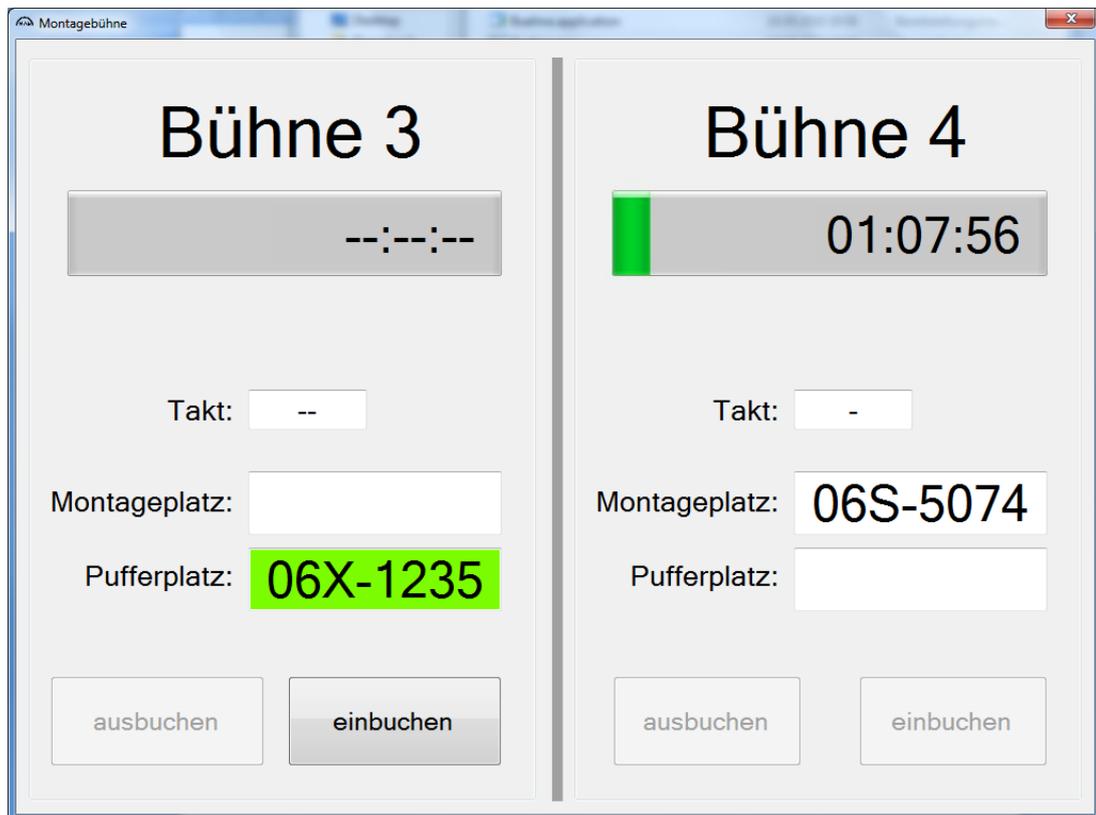


Abbildung 7 - Bühnen-GUI

Wird ein Dach am Master auf den Pufferplatz einer Bühne zugewiesen, so taucht dieses unmittelbar im entsprechenden Pufferplatz-Feld auf. Zugleich wird in der Datenbank geprüft, ob der zugehörige Kommissionierwagen bereits gefüllt wurde. Ist dies der Fall, so wird das Pufferplatz-Feld grün hinterlegt. Außerdem wird bei der Zuweisung durch den Master der *einbuchen*-Button aktiviert (siehe *Abbildung 7 - Bühnen-GUI*,

linke Seite). Durch Betätigen dieser Schaltfläche kann der Bühnenmitarbeiter den Prozessschritt des Dachumsetzens vom Pufferplatz auf dem Montageplatz im System angeben. Der *einbuchen*-Button wird deaktiviert, die Fahrzeugnummer des Dachs wechselt vom Pufferplatz auf dem Montageplatz und der Werker kann mit dem Ausstatten des Dachs beginnen. Als Hilfestellung wurde ein Ladebalken mit abnehmender Zeitanzeige eingefügt, der dem Werker die verbleibende Montagezeit des zu montierenden Dachs angibt aktiviert (siehe *Abbildung 7 - Bühnen-GUI*, rechte Seite). Die Montagezeit eines Dachs lässt sich aus diversen Quellen ableiten: zuerst wird die Stückliste zu der Fahrzeugnummer abgefragt, die alle zu verbauende Teile beinhaltet. Anschließend wird aus den Arbeitsplänen zu jedem zu verbauendem Teil die Dauer des Einbaus ausgelesen und aufsummiert. Die so erhaltene summierte Montagezeit wird zum Schluss adäquat auf die Anzahl der Werker an der Arbeitsbühne umgerechnet.

Damit der Werker den Überblick über die Position des zugehörigen Fahrerhauses auf dem Hauptband nicht verliert, wird in einem weiteren Feld die entsprechende Taktzahl ausgegeben. Dieses Feature ist besonders wichtig, da bei Montagestörungen schnell erkannt werden kann, wie zeitkritisch die Fertigstellung werden könnte. Zur optischen Unterstützung fängt die Taktanzeige ab einer definierten Taktunterschreitung an rot zu blinken.

Nach Ablauf der Montagezeit sollte der Werker die Montage des Dachs beendet haben sollen. Zugleich wird der *ausbuchen*-Button aktiviert mit dessen Betätigung die Fahrzeugnummer des Dachs aus dem Pufferplatzfeld entfernt wird. Systemintern befindet sich das Dach nun auf dem Gurtband.

### 6.1.3 GURTBAND

Damit sich der Schichtleiter, der sich hauptsächlich in seinem Büro aufhält, einen Überblick über die aktuelle Situation seiner Abteilung verschaffen kann, wurde die Gurtband-GUI realisiert (siehe *Abbildung 8 - Gurtband-GUI*). Diese zeigt die Belegung des Gurtbands in Echtzeit an.

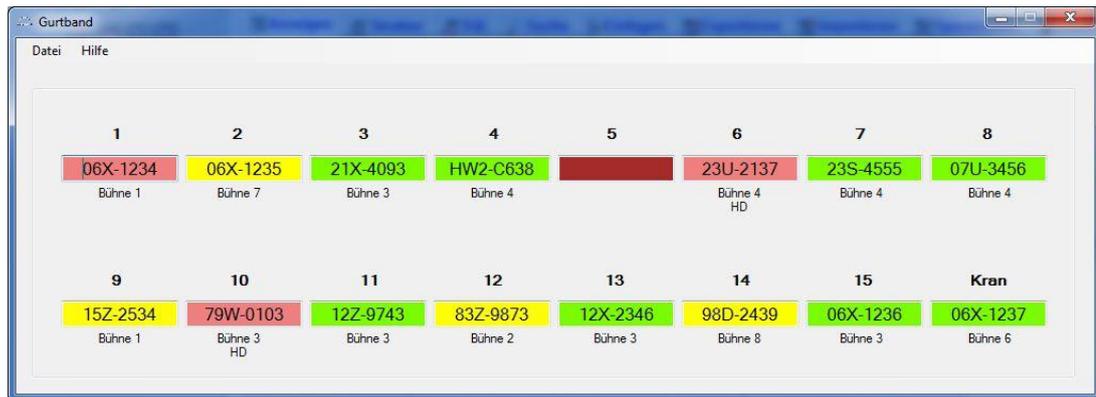


Abbildung 8 - Gurtband-GUI

Die verschiedenen Farben deuten auf unterschiedlich weite Fertigungszustände hin. So bedeutet eine grün hinterlegte Fahrzeugnummer, dass das Dach fertig montiert auf dem Gurtband abgelegt wurde. Gelbe Felder weisen auf ein Dach hin, das sich zur Zeit noch auf dem Montageplatz befindet. Rot hinterlegte Fahrzeugnummern kennzeichnen Dächer, die sich noch im Pufferplatz befinden, wohingegen braun hinterlegte Felder das komplette Ausbleiben eines Dachs im DMS angeben. Unter den Fahrzeugnummern werden die Arbeitsbühnen, an denen das Dach montiert wird, wie auch der Dachtyp (Hochdach, Blechdach) ausgegeben.

Neben dieser Maske hat der Benutzer außerdem die Möglichkeit Einstellungen am DMS vorzunehmen. Hierzu können diverse Untermasken über das Menü geöffnet werden. Einstellmöglichkeiten sind beispielsweise die Eingabe der Schichtinformationen, die für jede Schicht definiert werden müssen. Diese sind notwendig, um die Zeitanzeigen der Bühnen-GUIs in den Schichtpausen und nach Schichtende anzuhalten. Weitere Einstellmöglichkeiten bestehen in der Anpassung der Systemparameter mit denen das gesamte DMS justiert werden kann. Hierzu zählen beispielsweise die Angabe des Taktes, bei dessen Unterschreitung das Taktfeld der Bühnen-GUI farblich zu blinken anfängt. Ein weiteres Beispiel ist die Regulierung von Warnmeldungen für die Gurtband-GUI: Überschreitet beispielsweise ein sich auf dem Pufferplatz befindliches Dach die zehnte Position des Gurtbands, so erhält der Schichtleiter eine Meldung mit der Fahrzeugnummer, dem Dachtyp und der Arbeitsbühne. Für die verschiedenen Prozessfortschritte (Master, Pufferplatz, Montageplatz, Gurtband) können diverse Taktschwellen definiert werden.

Da jede Transaktion (Einscannen, Einbuchen, Ausbuchen, ...) mitgeloggt wird, wurde ein Statistikmodul zur Datenauswertung realisiert. Neben dem Nachschlagen des Prozessfortschritts eines Dachs mithilfe dessen Fahrzeugnummer, kann ebenfalls die aufsummierte Wartezeit, die es im Rahmen dieser Masterarbeit zu minimieren gilt, pro Bühne oder pro Schicht berechnet werden.

Genauere Informationen zu den Optionen und Statistiken können dem Anhang 11.2 *Benutzerhandbuch für DMS* entnommen werden.

#### 6.1.4 DAEMON

Die Daemon-Komponente (siehe *Abbildung 9 - Daemon-GUI*) ist hauptsächlich für die Datenakquisition aus der MAN-IT-Infrastruktur zuständig. Das bedeutet, dass alle dachbezogenen Informationen, wie z.B. Montagezeiten, Ausstattung und Position des zugehörigen Fahrerhauses, auf dem Hauptband in diversen Tabellen bzw. Views auf einem Großrechner in einer Oracle-Datenbank im MAN-Netzwerk abgelegt sind. Diese gilt es mit dem DMS zu verknüpfen.

Hierzu wurden lesende Datenbankzugriffe auf die Oracle-Datenbank mithilfe eines Oracle-Connectors implementiert, die je nach Konfiguration automatisch in gewissen Intervallen oder auch manuell durchgeführt werden können. Die Synchronisationsintervalle lassen sich in den Systemparametern anpassen (siehe *11.2 Benutzerhandbuch für DMS*). Um das Datenaufkommen so gering wie möglich zu halten, werden die Intervalle je nach Nutzung konfiguriert: so ist es beispielsweise ausreichend, die Montagezeiten alle 24 Stunden zu aktualisieren, wohingegen die Position des zugehörigen Fahrerhauses auf dem Hauptband mindestens alle 3:45 min abgeglichen werden sollte, da dies der Taktzeit entspricht. Zur besseren Übersicht und Debugging-Zwecken wird der Zeitpunkt der letzten Synchronisation wie auch dessen Dauer mitgeloggt und in der GUI ausgegeben.

Das Datenaufkommen wird durch den Einsatz des Daemons ebenfalls gering gehalten, indem nur dieser mit der Oracle-Datenbank kommuniziert und nicht alle weiteren oben genannten Komponenten. Die Zugriffsraten werden von den MAN-Systemadministratoren streng überwacht und bei zu starker Belastung kann es zu einer Sperrung des Datenbank-Users kommen. Ein weiterer Vorteil des Daemons als Single Point of Access (SPA) ist, die daraus resultierende einfachere System-Architektur. Es reicht aus, wenn die restlichen Komponenten ausschließlich auf die systeminterne Datenbank zugreifen und die Kommunikation mit der restlichen MAN-IT-Infrastruktur dem Daemon überlassen wird.

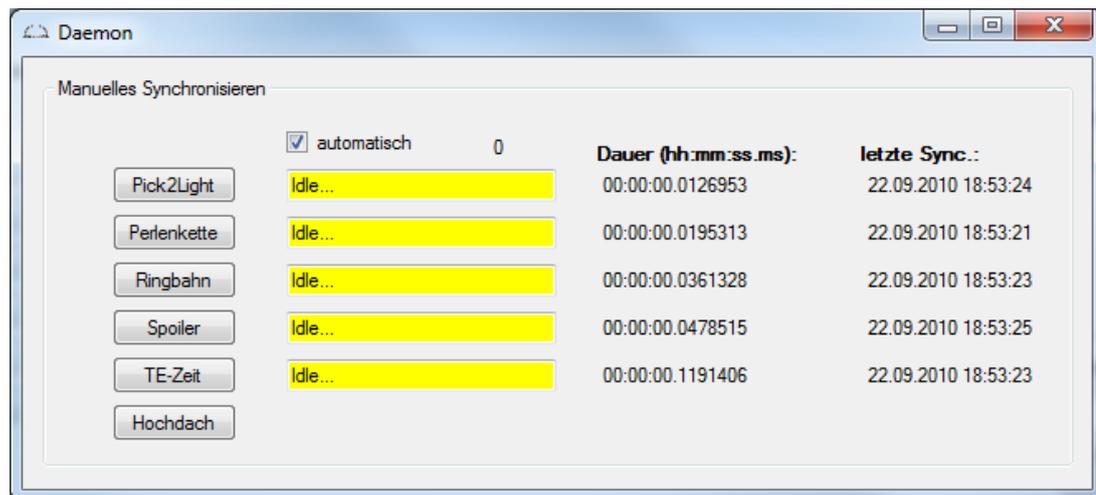


Abbildung 9 - Daemon-GUI

Neben der Datensynchronisation ist diese Komponente ebenfalls für das zyklische Prüfen des Schichtstatus zuständig. Durch Abgleich der systeminternen Uhrzeit mit den zuvor hinterlegten Schichtplänen kann entschieden werden, ob das DMS sich im Betriebsmodus oder auch im Pausenmodus befinden soll. Hierzu wird ein Flag in der Datenbank gesetzt, das von allen anderen Komponenten in regelmäßigen Abständen abgefragt wird.

## 6.2 DATENBANKBASIERTE CLIENT/SERVER-ARCHITEKTUR

Aus den in *5 Konzepte* erarbeiteten Konzepten wurde die datenbankbasierte Lösung gewählt. Grund dafür ist vor allem die hohe Flexibilität und Modellierbarkeit einer Datenbank für die gegebenen Anforderungen. Weiterhin muss auch die Gewährleistung des gleichzeitigen, jederzeit möglichen Zugriffs auf die Daten erwähnt werden. Zwar wäre ein SPS-gesteuertes System ebenfalls denkbar gewesen, jedoch hätte sich der Schwerpunkt dieser Masterarbeit auf die Konfiguration und Programmierung von SPS und Betriebsstellenfilter verlagert. Außerdem bietet letzteres Konzept keine so guten Möglichkeiten zum Auswerten von Log-Daten und damit zur Erstellung von Statistiken.

Als zentralisiertes Datenbanksystem kommt wie bereits erwähnt eine relationale MySQL-Datenbank zum Einsatz. Dabei lag beim Entwurf des Schemas der Schwerpunkt auf einer adäquaten und anforderungsorientierten Modellierung, da eine enge Kopplung zwischen dem Datenbankdesign und den entwickelten GUIs besteht. Konkurrierende Zugriffe wurden durchgängig mit einer Vermeidungsstrategie gelöst. Dadurch müssen keine kritischen Bereiche definiert und Transaktionen eingesetzt werden. Damit bei Datenänderungen in Tabellen die darauf verweisenden Felder anderer Tabellen nicht in den Methoden der Komponenten angepasst werden müssen, wurden außerdem diverse Primär- und Sekundärschlüssel definiert. Auf ein Acknowledge von Aktionen (z.B. Einbuchen, Ausbuchen...) wurde bei der Konzeptionierung verzichtet, da es sich um kein kritisches System handelt. Der Schwerpunkt der Arbeit liegt auf dem Roll-Out eines funktionierenden Systems, sodass der Einsatz von (Log-)Views, beispielsweise für die Statistiken, erst in einer Erweiterung des Systems denkbar wäre.

### 6.2.1 DATENBANKMODELL

Das in *Abbildung 10* auf Seite 39 gezeigte Datenbankmodell bildet das Schema der eingesetzten Datenbank ab. Aus Gründen der Übersichtlichkeit wurde auf die Angabe der Typen verzichtet. Die linken weißen Kästen symbolisieren Tabellen bzw. Views der Oracle-Datenbank auf dem MAN-Großrechner. Die ausgegrauten Bereiche deuten auf eine Ausbaumöglichkeit des Systems mit diversen Komponenten hin (siehe 7.2 Ausblick). Der mit "Procedures" betitelte Kasten weist diverse stored procedures aus. Da das RPC-Konzept nicht umgesetzt werden konnte, wurde kurzfristig der Einsatz von stored procedures als Alternative überdacht. Stored Procedures sind Folgen von SQL-Anweisungen kombiniert mit weiteren Sprachelementen zur Ablaufsteuerung und bieten sich als Schnittstelle zwischen der Applikation und dem Datenbank-Server an [7]. Es kann nur eine Stored Procedure gleichzeitig ausgeführt werden sodass konkurrierende Zugriffe von vornherein vermieden werden. Des Weiteren wird die Unabhängigkeit der Plattform und der Programmiersprache gewährleistet. Aus zeitlichen Gründen konnte diese Alternative leider nicht weiter verfolgt werden.

Beim Entwurf des Datenbankmodells wurde die Fahrzeugnummer oftmals als Primärschlüssel (PK) verwendet, da sie eindeutig ist. Folglich wurde die *DachStatus*-Tabelle mit PK *Fahrzeugnummer* als zentrale Tabelle mit einigen Fremdschlüsseln (FK) konzipiert. Somit enthält diese Tabelle nahezu alle Informationen, die die verschiedenen Komponenten benötigen. Das Pufferplatzfeld der Bühnen-GUI korreliert beispielsweise direkt mit dem Eintrag der Spalte *Fahrzeugnummer*, wenn die Spalte *Position* den Eintrag *Pufferplatz* aufweist. Ähnlich verhält es sich mit dem Montageplatzfeld. *KW\_ready* deutet auf den Fortschritt der Befüllung des Kommissionierwagens hin. Je nach hinterlegtem Wert ändert sich die Hintergrundfarbe des Pufferplatzfeldes in der Bühnen-GUI. *Takt* gibt an, an welcher Position sich das zugehörige Fahrerhaus auf dem Ausstattungsband befindet. Auch hier bestimmt der hinterlegte Wert, wann das Taktfeld der Bühnen-GUI rot zu blinken anfängt. *BuehnenNr* weist auf die Bühne hin, auf die das Dach montiert wird bzw. wurde und *Prozent* gibt an, zu wie viel Prozent ein Dach ausgestattet werden soll. Diese Funktion wäre besonders bei Schichtwechseln wichtig, wenn die Folgeschicht die Montage eines bereits angefangenen Dachs vollenden muss. Somit würden die Werker die Mitteilung erhalten, bei welchem Arbeitsschritt sie ihre Arbeit unterbrechen sollen.

Jede Aktion (Einbuchen, Ausbuchen) wird im Normalfall in der Tabelle *Log\_Dachhistorie* mitgeloggt. Hier werden die Timestamps zu der Fahrzeugnummer (PK) bei Änderung der Position des Dachs (z.B. von Pufferplatz auf Montageplatz) hinterlegt. Falls ein Dach in der Master-GUI von einem Pufferplatz auf einen anderen Pufferplatz verschoben wird, taucht diese Aktion in der Tabelle *Log\_DachHistorie\_extended* auf. In *Log\_BuehnenHistorie* können zu einer bestimmten Schicht die aufsummierte Wartezeit (*Idle\_Zeit*), die Mitarbeiterbelegung (*MA\_Anzahl*),

die geleistete Montagezeit (*TE\_Zeit\_Buehne*) und die summierte Zeit, die der Bühnenmitarbeiter länger zum Montieren benötigt hat (*AusEinbuchen\_Zeit*), ausgelesen werden. Aus diesen Log-Tabellen lassen sich unter anderem über Kreuzprodukte die Statistiken erstellen [7]. Auf eine einzige Logtabelle, die alle Informationen beinhaltet, wurde aus Performanz- und Übersichtlichkeitsgründen verzichtet und dafür Redundanz in Kauf genommen.

In der Tabelle *Schichtinformationen* definiert der Schichtleiter zu jeder Schicht deren Beginn und Ende, wie auch die Pausen und deren Längen.

In *Optionen* werden alle Systemparameter hinterlegt. Diese unterscheiden sich durch deren in *Opt\_Bezeichnung* hinterlegten Bezeichnung voneinander. Je nach Parametertyp wird der Wert des Parameters in die Spalte *int\_wert*, *text\_wert* oder *date\_wert* abgelegt.

*GUI\_online* wird für die Untermaske der Gurtband-GUI benötigt. In dieser Maske wird angezeigt, welche GUIs zur Zeit laufen und welche nicht. Dieser Status ist in der Spalte *online* hinterlegt. Die Mitteilung der einzelnen Komponenten an die Datenbank über ihren Online-Status wurde über ein Heartbeat realisiert, der in der Spalte *last\_signal* permanent aktualisiert wird. Dieser Wert wird ebenfalls durch den Daemon regelmäßig überprüft, der ihn mit seiner Systemzeit abgleicht. Weicht ein Wert in *last\_signal* zu stark von der aktuellen Systemzeit ab, so kann davon ausgegangen werden, dass die entsprechende Komponente nicht mehr online ist und der Master setzt die zugehörige Spalte *online* auf 0.

Die Tabelle *BuehnenStatus* bildet den aktuellen Zustand der Bühnen ab. Hier kann die Mitarbeiterbelegung (*MA\_Anzahl*), die bisher geleistete Montagezeit (*bisherige\_TE\_ZeitSum*) und die Fahrzeugnummer des Dachs, das sich zum aktuellen Zeitpunkt auf dem Montageplatz befindet (*akt\_fahrzeugnummer*) ausgelesen werden. Auch hier wurde Redundanz (siehe Tabelle *DachStatus*) zugunsten von Performanz- und Übersichtlichkeitsgründen in Kauf genommen. In der Spalte *akt\_verbl\_sekzeit* wird die verbleibende Montagezeit eines derzeit auszustattendes Dachs zyklisch durch die Bühnen-GUI aktualisiert. Hintergrund hierfür ist, dass beim Schließen und erneutem Öffnen der Bühnen-GUI die Fahrzeugnummern zwar weiterhin in den richtigen Feldern angezeigt wurden, der Countdown jedoch von neuem mit der kompletten Montagezeit gestartet wurde. Dies haben die Werker zu Ihrem Vorteil ausgenutzt. Ein weiterer Vorteil des Zwischenspeicherns der Montagezeit ist eine sehr einfache Entwicklung eine Übersichtsmaske mit den verbleibenden Montagezeiten (siehe 7.2 *Ausblick*, Abschnitt Kommissionierungs-GUI und Umsetzkran-GUI).

Die restlichen Tabellen beinhalten Informationen der MAN-Datenbank, die im nächsten Kapitel genauer beschrieben werden.

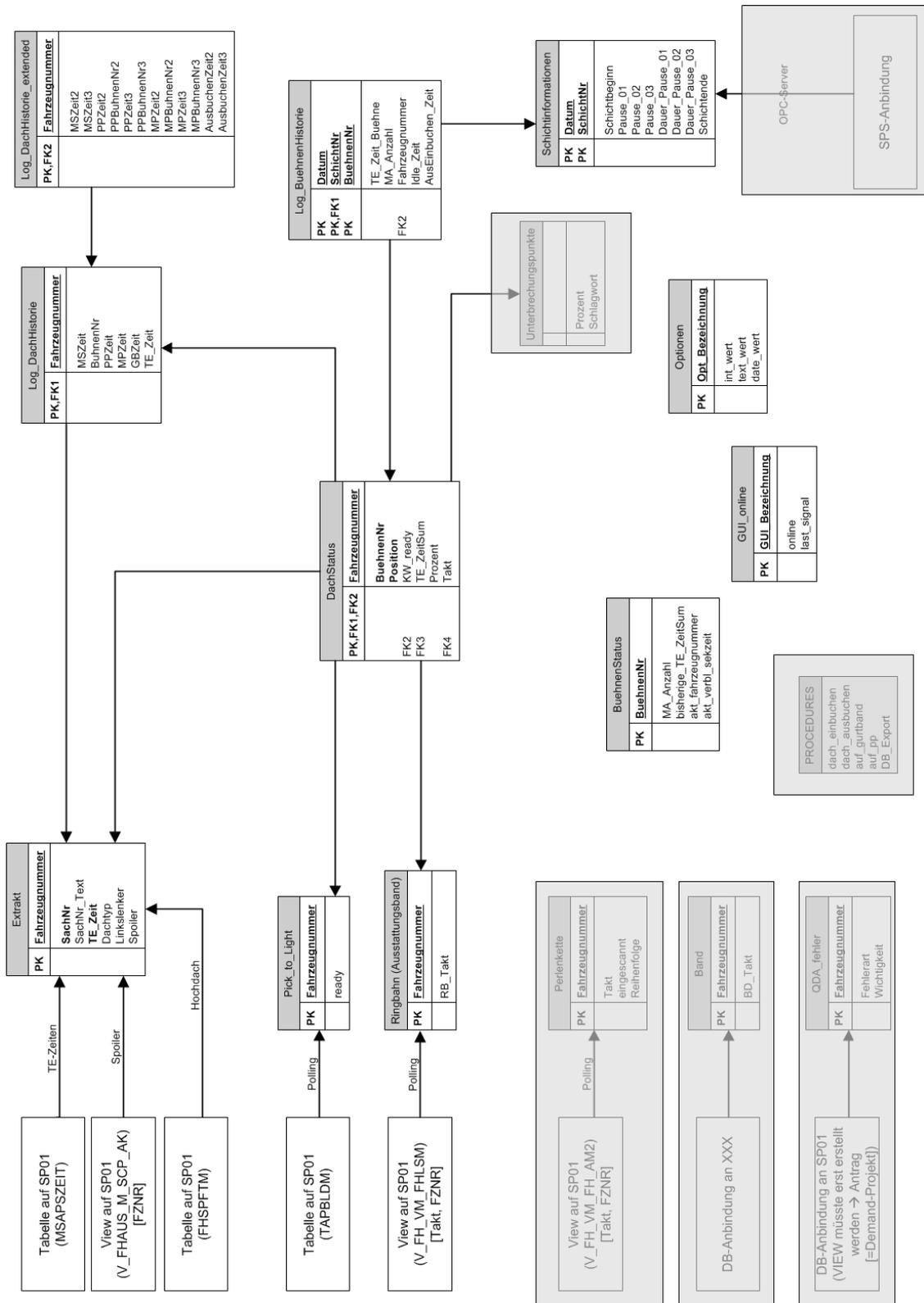
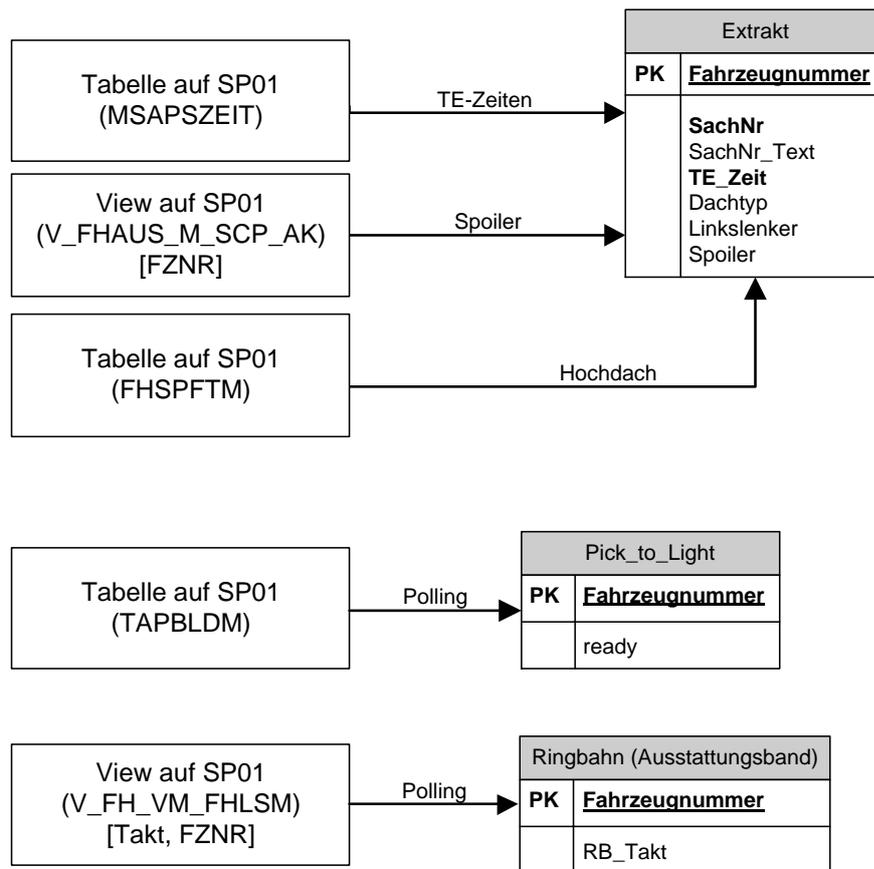


Abbildung 10 - Datenbankschema

## 6.2.2 DATENBANKZUGRIFF AUF IBM-HOST

Alle fahrzeugspezifischen Datensätze liegen auf einem Großrechner mit der Bezeichnung SP01, auf dem eine Oracle-Datenbank läuft. Diese Daten gilt es mit dem DMS zu verknüpfen, was mit einem zyklischen Polling durch den Daemon realisiert wurde (siehe *Abbildung 11 - Polling von Oracle-DB in MySQL-DB*).



**Abbildung 11 - Polling von Oracle-DB in MySQL-DB**

Fahrzeugspezifische Daten sind beispielsweise die Montagezeit (Tabelle *MSAPSZEIT*), der Status des Kommissionierwagens (Tabelle *TAPBLDM*) oder auch die Position auf dem Ausstattungsband (View *V\_FH\_VM\_FHLSM*).

Wie in 6.1.4 *Daemon* beschrieben, greift aus Performanzgründen ausschließlich der *Daemon-Service* auf die Oracle-Datenbank zu. Hierzu setzt er beispielsweise folgende SQL-Query ab:

```
SELECT BTRF, FZNR, MAX(LFD_NR) AS 'MAX_LFD_NR', BEARB_KZ
      FROM MONTAUF.TAPBLDM
      WHERE BTRF          = '4551PDT'
      GROUP BY BTRF, FZNR, BEARB_KZ
;
```

Die erhaltene Ergebnismenge wird sofort in die entsprechende MySQL-Tabelle geschrieben und wird nur in Form eines Tupels in einer Variable zwischengespeichert. Mit einer *while*-Schleife wird durch alle Tupeln iteriert. Je nach Bedarf werden die Attribute der Tupeln in der entsprechenden MySQL-Tabelle neu erstellt oder aktualisiert:

```
INSERT INTO Pick_to_Light ('Fahrzeugnummer ', 'ready ')
      VALUES
      (Tupel.BEARB_KZ, Tupel.FZNR)
;
```

Bzw.:

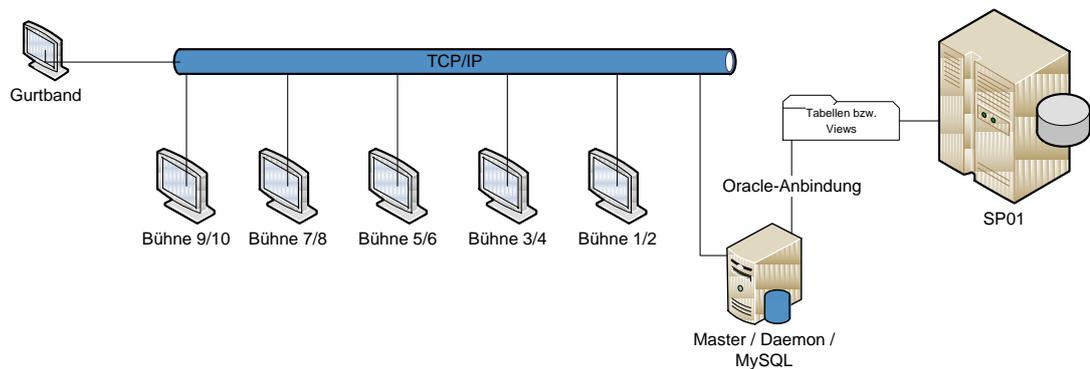
```
UPDATE Pick_to_Light
      SET
      ready          = Tupel.BEARB_KZ
      WHERE
      Fahrzeugnummer = Tupel.FZNR
;
```

Nach diesem Prinzip lassen sich in unterschiedlichen Synchronisationsintervallen alle benötigten fahrzeugspezifischen Daten der Oracle-Datenbank aus- und in die MySQL-Datenbank einlesen. Sobald ein Wert in letzterer Datenbank aktualisiert wird, wird diese Änderung per Cascade an weitere Tabellen weitergegeben. Bewegt sich beispielsweise das *Ausstattungsband*, so wird diese Änderung in der View *V\_FH\_VM\_FHLSM* in der Oracle-Datenbank sichtbar. Beim nächsten Polling wird die Tabelle *Ringbahn* der MySQL-Datenbank die Änderungen übernehmen. Diese Änderungen werden aber ebenfalls automatisch in der Tabelle *DachStatus*

übernommen, da beim Erstellen der FK-Constraints in der Tabelle *DachStatus* der Zusatz "ON UPDATE CASCADE" verwendet wurde.

### 6.3 KOMMUNIKATION DER KOMPONENTEN

Die einzelnen Komponenten, wie auch die Datenbank, müssen miteinander kommunizieren können. Hierzu wurde folgende physikalische Anbindung über Ethernet realisiert:



**Abbildung 12 - Anbindung der Komponenten an die hausinterne IT-Infrastruktur**

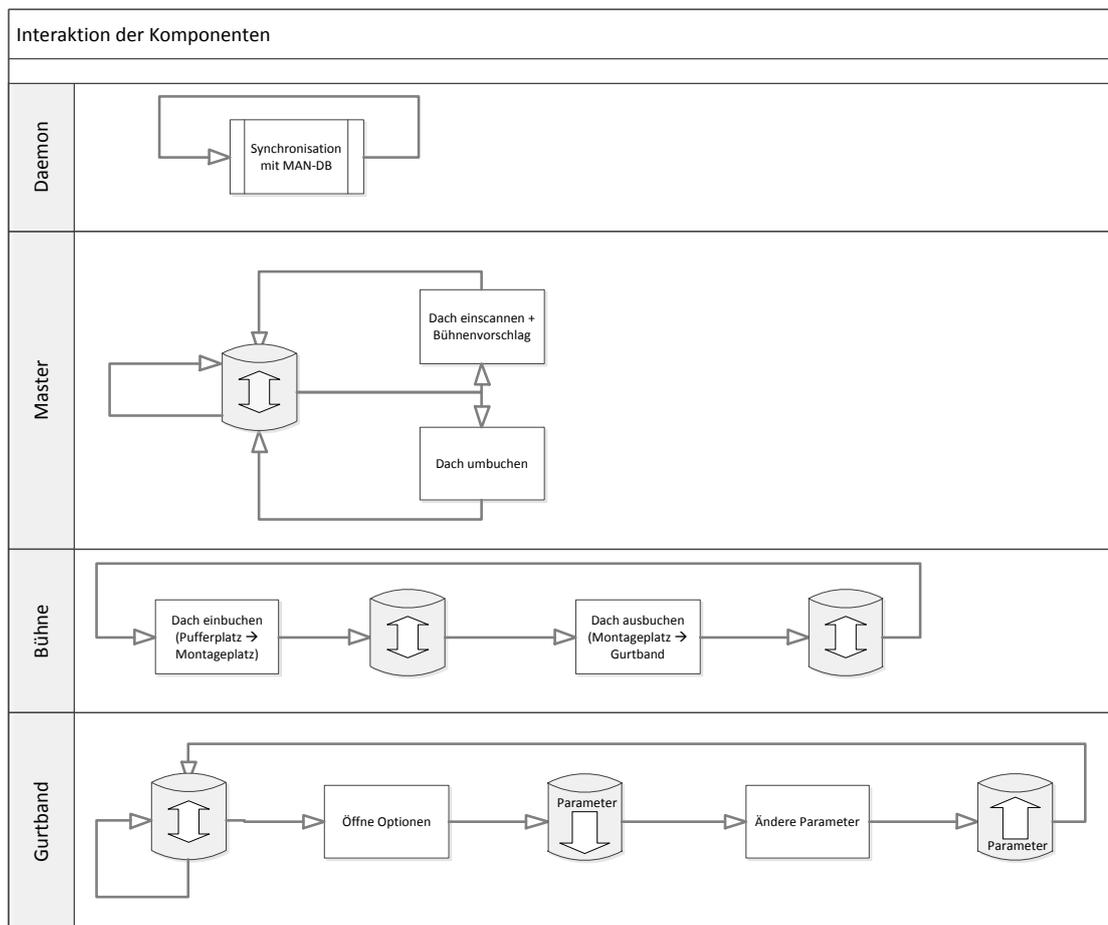
Die MySQL-Datenbank, die neben dem Daemon auf dem Master-PC untergebracht ist, kann über das MAN-LAN mittels TCP/IP von den verschiedenen Komponenten angesprochen werden. Der Daemon fungiert quasi als Cache-Refresher, da er als einziger auf den MAN-Großrechner zugreifen kann, auf dem die Oracle-Datenbank läuft und die Einträge der MySQL-Datenbank aktualisiert. Somit wird sichergestellt, dass die Komponenten mit einem aktuellen Datenbestand arbeiten.

Diese zyklische Aktualisierung wird auch in *Abbildung 13 - Interaktion der Komponenten* deutlich. Zwar differieren die Synchronisationsintervalle der verschiedenen Tabellen deutlich voneinander, jedoch bleibt die prinzipielle Funktionsweise die selbe.

Neben dem Daemon läuft die Master-GUI. Diese befindet sich hauptsächlich in einem Wartezustand, in dem die zehn Felder der Puffer- wie auch Montageplätze im Zweisekudentakt aktualisiert werden. Wird ein Dach eingescannt, so wird der Zuordnungsalgorithmus gestartet. Das neue Dach wird daraufhin im System aufgenommen und die Master-GUI wechselt wieder in den Wartezustand. Änderungen an der Zuordnung der Dächer zu den Arbeitsbühnen bzw. an der Mitarbeiterbelegung werden beim Speichern und Schließen der Optionen-Maske propagiert.

Nachdem in der Bühnen-GUI nur die Aktionen *Einbuchen* und *Ausbuchen* möglich sind, ist eine Benachrichtigung der Datenbank über Änderungen nur nach deren Ausführung notwendig. Ansonsten wird auch in dieser Maske regelmäßig lesend auf die Datenbank zugegriffen, um womögliche Änderungen an relevanten Daten zu erfahren.

Bei der Gurtband-GUI verhält es sich ähnlich wie bei der Master-GUI: die Hauptanzeige wird zyklisch durch Polling aktualisiert. Will der Schichtleiter Änderungen an den Parametern vornehmen, so werden diese aus der Datenbank in die Maske geladen. Nach erfolgreichem Speichern und Schließen werden die kompletten Parameter in die Datenbank zurückgeschrieben.



**Abbildung 13 - Interaktion der Komponenten**

Die Interaktion der einzelnen Komponenten untereinander wird durch das zyklische Polling sichergestellt. So werden beispielsweise die Bühnen-GUIs in den Pause-Modus versetzt, indem der Daemon ein Flag in der Datenbank setzt, das von den Bühnen-GUIs beim nächsten zyklischen Polling gelesen wird. Ähnlich verhält es sich bei der Master-GUI, wenn ein Dach an der Bühne vom Pufferplatz in den Montageplatz eingebucht

wurde. Der Versatz von wenigen Sekunden (Polling-Intervall) ist beim DMS in der Praxis keinesfalls kritisch.

Abbildung 14 - Prozesszyklus eines Dachs zeigt den Prozessablauf der Bühnenzuordnung von der Anlieferung eines Dachs bis zum Ablegen auf dem Gurtband. Der Zuordnungsalgorithmus beim Master wird im nächsten Kapitel genauer erläutert.

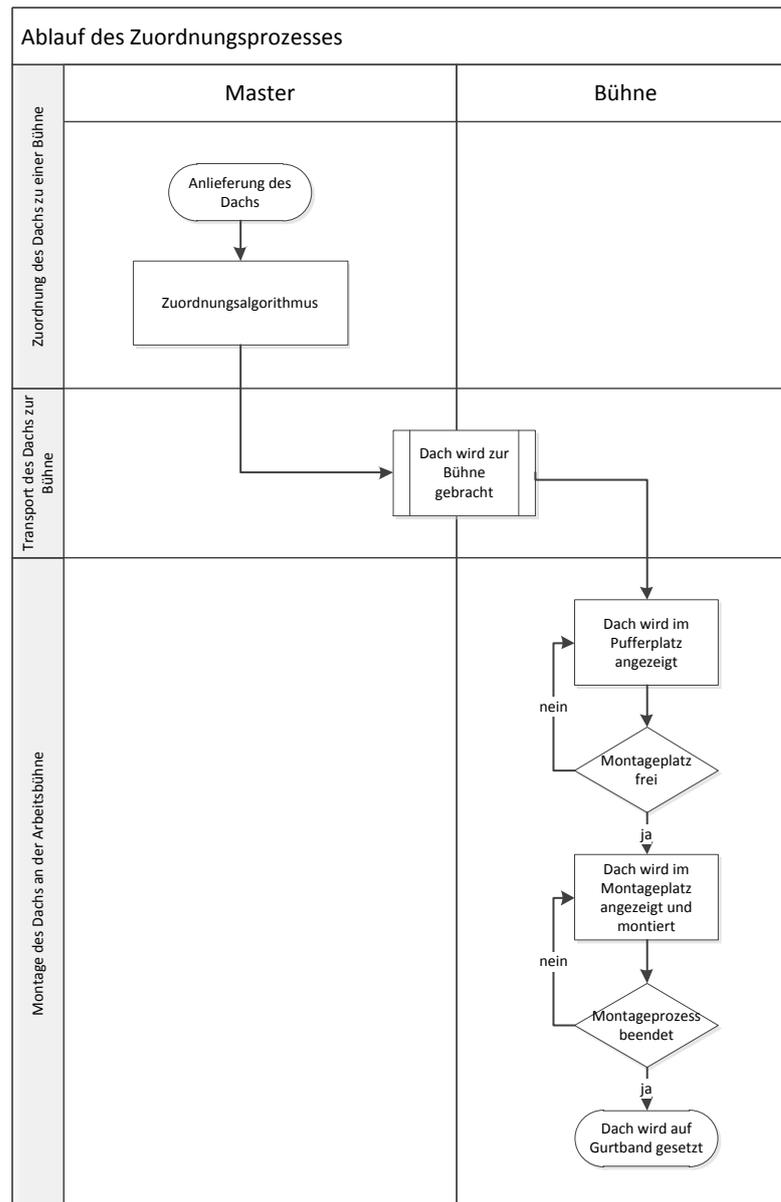


Abbildung 14 - Prozesszyklus eines Dachs

## 6.4 ZUORDNUNGsalGORITHMUS AM MASTER

Welche Bühne das soeben eingescannte Dach als nächstes montieren soll, hängt primär von der verbleibenden Montagezeit der Bühnen ab. Bei genauerer Betrachtung des Prozesses erkennt man aber, dass weitere Faktoren wie die Mitarbeiteranzahl und auch der späteste Fertigungszeitpunkt in die Entscheidungsfindung einfließen. Prinzipiell liegt aber die Idee der Scheduling Strategie LST zu Grunde, da die zehn Arbeitsbühnen als Multicore-Prozessor und die Montagezeiten der Dächer als Threads interpretiert werden können.

In *Abbildung 15* wird der Zuordnungsalgorithmus in einem Prozessflussdiagramm aufgeschlüsselt. Der Algorithmus fängt immer mit dem Einlesen der Fahrzeugnummer eines Dachs am Master mithilfe des Scanners. Daraus lassen sich Informationen wie Montagezeit, Dachtyp und Verbau eines Spoilers ableiten. Anschließend prüft der Algorithmus, ob aktivierte Arbeitsbühnen leer stehen, d.h., dass sowohl der Pufferplatz wie auch der Montageplatz kein Dach beinhalten. Dieser Bühnenzustand muss unter allen Umständen vermieden werden, da hierdurch die Leerzeiten in die Höhe getrieben werden. Falls eine solche Bühne erkannt wird, wird das zu montierende Dach auf dessen Pufferplatz eingebucht und der Algorithmus ist beendet.

Falls keine der Arbeitsbühnen leer steht, werden im nächsten Schritt alle deaktivierten Bühnen aus der Auswahl der potentiell nächsten Bühnen entfernt. Eine deaktivierte Bühne ist jene, deren Mitarbeiteranzahl vom Schichtleiter auf 0 gesetzt wurde. Die übrig gebliebene Menge an Bühnen wird mit einem Bubblesort-Algorithmus nach deren aktuell verbleibenden Montagezeiten sortiert. Die Bühne mit der kürzesten verbleibenden Montagezeit wird anschließend darauf geprüft, ob deren Pufferplatz leer steht. Falls der Pufferplatz der Bühne bereits belegt sein sollte, wird die nach Bubblesort nächste Bühne auf einen freien Pufferplatz geprüft. Anderenfalls berechnet der Zuordnungsalgorithmus die Montagezeit des soeben eingescannten Dachs im Verhältnis zu der Mitarbeiteranzahl an der Bühne und summiert das Ergebnis mit der aktuell noch verbleibenden Montagezeit der Bühne. Falls die Summe einen gewissen Schwellwert, der wiederum in den Parametern des DMS spezifiziert werden kann, nicht überschreitet, wird das Dach auf diese Arbeitsbühne eingebucht und der Algorithmus ist beendet. Eine Überschreitung dieses Schwellwertes würde ein sogenanntes Notdach nach sich ziehen. Dies bedeutet, dass die Ausstattung eines Dachs noch nicht vollendet ist, obwohl dieses in kurzer Zeit auf das Fahrerhaus geklebt werden soll. Eine solche Verzögerung kann im schlimmsten Fall ein Bandstopp nach sich ziehen.

Tritt jedoch eine Überschreitung des Schwellwerts auf, so wird die Höhe der Überschreitung (in Minuten) zwischengespeichert und alle weiteren, nach Bubblesort potentiell möglichen, Arbeitsbühnen geprüft. Dies geschieht solange, bis eine Arbeitsbühne ohne Montagezeitüberschreitung errechnet wurde. Ergibt sich bei allen

verbleibenden Bühnen eine Montagezeitüberschreitung, so wird das zu montierende Dach auf die Bühne mit der geringsten Überschreitung eingebucht. Außerdem gibt das System eine Warnung an den Verteiler aus, sodass dieser die Mitarbeiter an der entsprechenden Bühne vor dem Notdach warnen kann.

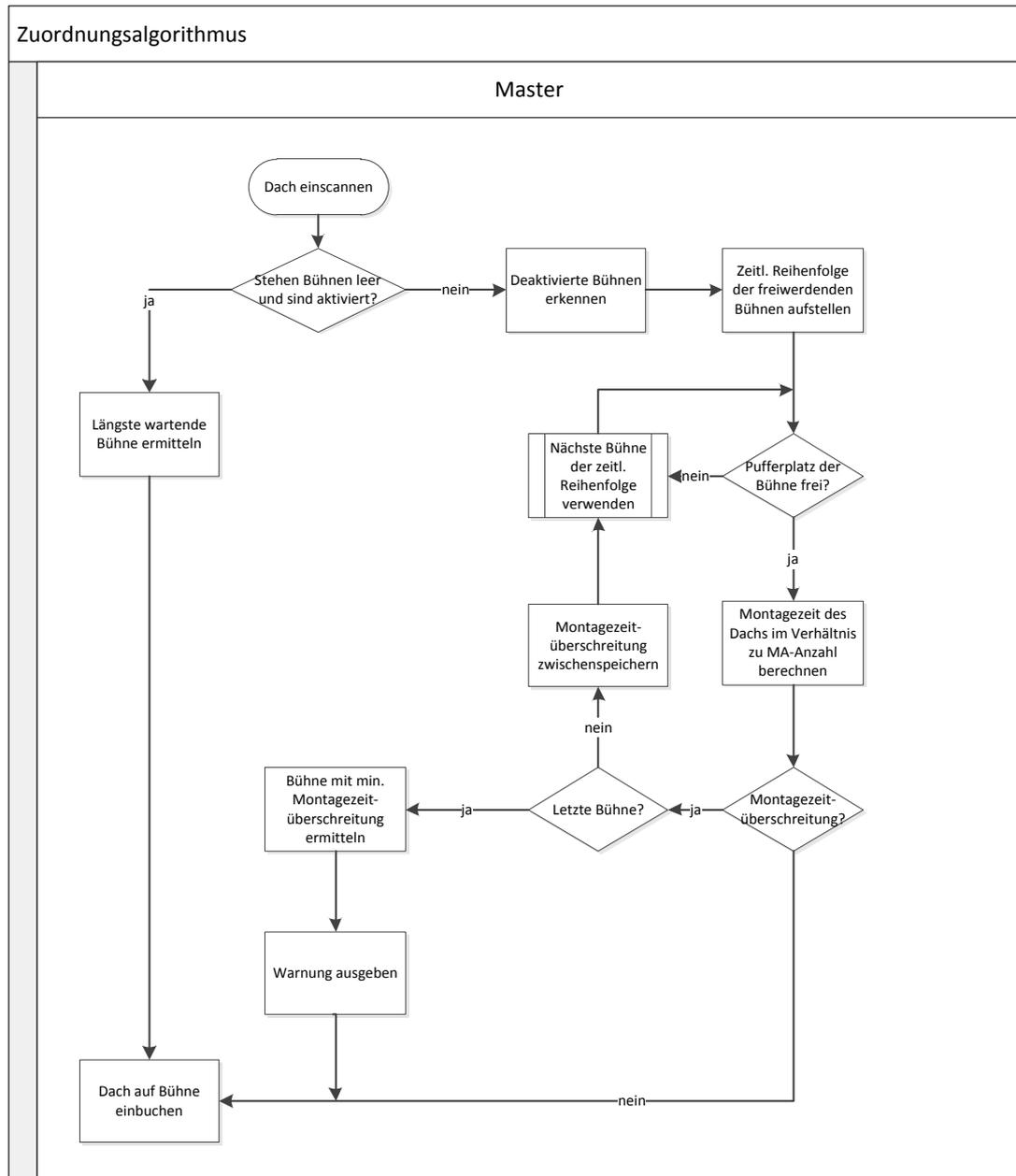


Abbildung 15 - Grafische Beschreibung des Zuordnungsalgorithmus

Bei der Erstellung der zeitlichen Reihenfolge der freiwerdenden Arbeitsbühnen und der Prüfung der Montagezeitüberschreitung erkennt man eine Korrelation zur LST Strategie. Bei LST soll der Prozess mit dem kleinsten Spielraum identifiziert werden. Ähnlich hierzu muss in dieser Anwendung die Arbeitsbühne mit der kleinsten verbleibenden Montagezeit identifiziert werden. Außerdem muss bei der Prüfung der Montagezeitüberschreitung - wie bei LST - der Spielraum wie folgt berechnet werden:

$$\text{Spielraum} = \text{Schwellwert} - (\text{verbleibende Montagezeit} + \text{Montagezeit des neuen Dachs})$$

Hieraus lässt sich je nach Vorzeichen des Spielraums ableiten, ob eine Montagezeitüberschreitung stattfinden wird (negatives Vorzeichen) oder nicht (positives Vorzeichen).

Durch die Prüfung auf die Montagezeitüberschreitung, kann der Algorithmus das Hauptziel der Wartezeitminimierung nicht mehr zu 100% erreichen. Als Beispiel könnte folgendes Szenario mit einem Schwellwert von 76 Minuten und zwei Arbeitsbühnen mit leeren Pufferplätzen dienen: Bühne 1 ist mit einem Werker belegt und hat eine Restzeit von 3 Minuten, wobei Bühne 2 mit zwei Werkern belegt ist und eine Restzeit von 8 Minuten hat. Ein Hochdach mit einer Montagezeit von 74 Minuten wird eingescannt. Fände keine Prüfung auf Montagezeitüberschreitung statt, so würde dieses Hochdach auf Bühne 1 zugewiesen werden und nach 77 Minuten ausgestattet sein. Da jedoch auf Montagezeitüberschreitung geprüft wird, wird Bühne 2 dieses Hochdach erhalten, sodass es nach 45 Minuten ( $8 \text{ min} + \lceil 75 \text{ min} / 2 \rceil$ ) ausgestattet wäre. Wird nach dem zugewiesenen Hochdach innerhalb der nächsten 3 Minuten (=Restzeit Bühne 1) kein neues Dach eingescannt, tritt der Fall ein, dass Bühne 1 nicht weiter arbeiten kann und somit Wartezeit entsteht.

## 7 FAZIT

Ziel der Masterarbeit war die Entwicklung eines Systems zur Optimierung der Einsteuerung der Dächer in der Fahrerhausmontage. Die Hauptursache für die oft auftretende Leerzeiten an den Arbeitsbühnen war das durch den Verteiler manuell geführte Protokoll, das keinen klaren Rückschluss lieferte welche Bühne als nächstes beliefert werden sollte.

Nach der Analyse des komplex ineinandergreifendes Systemablaufs konnte ein erstes Testsystem gestartet werden, welches mit der Zeit durch schwer vorhersehbare Szenarien erweitert wurde. Neben den programmier- wie auch prozesstechnischen Herausforderungen galt es ebenfalls bürokratische Hürden zu meistern und Akzeptanz bei den Mitarbeitern zu erreichen.

Mittlerweile wird das Dach Management System für die Produktion eingesetzt und auch weiterentwickelt.

### 7.1 ERGEBNISSE

Nach dem Roll-Out des DMS wurden aus den Statistiken eine summierte Wartezeit bedingt durch den Verteiler von 15 Minuten pro Schicht registriert. Dies entspricht einer Reduktion von 255 Minuten. Damit wurde die Zielsetzung dieser Masterarbeit erreicht.

Die Rentabilität und Zeitersparnis des DMS lassen sich anhand von *Tabelle 1 - Ersparnis durch Einsatz des DMS* belegen. Dabei ergeben sich 4,5 Stunden Wartezeit vor dem Roll-Out aus den zehn Arbeitsbühnen mit einer durchschnittlichen Wartezeit von 27 Minuten.

Wartezeit vor Roll-Out	4,5 Stunden pro Schicht
Wartezeiten nach Roll-Out	15 Minuten pro Schicht
Schichten pro Tag	2
Arbeitstage pro Jahr	245 d
Minutensatz in €/min	0,77 €/min
Einsparung	$255 \text{ min} * 2 \text{ Schichten} * 245 \text{ d} * 0,77 \text{ €/min}$
	<b>= 96.211€</b>

**Tabelle 1 - Ersparnis durch Einsatz des DMS**

Bei einer grob geschätzten Investition von 20.000€ für Hardware und Dienstleistung ergibt sich ein Return on Investment ( $\text{RoI} = 20.000\text{€} / 96.211\text{€}$ ) von 51 Tagen [8]. Damit ist die Investition nach 2,5 Monaten amortisiert.

## 7.2 AUSBLICK

Selbstverständlich sind im Laufe der Entstehung dieser Masterarbeit noch viele weitere Ideen und Verbesserungen aufgetaucht, die aber den zeitlichen Rahmen gesprengt hätten. Zu allererst sollte die Umstellung des Systems auf eine push-basierende Architektur vorgenommen werden. Mit dieser Methode wäre das (zugegebenermaßen geringe) Risiko ausgeschlossen, dass zwei Transaktionen an verschiedenen Komponenten - aber mit dem selben Dach - zu Fehlern führen könnte. Außerdem würde sich das Datenaufkommen immens verringern, da nur kommuniziert werden würde, wenn Änderungen im System auftreten. Ob jedoch die Investition für ein Re-Design des DMS vorgenommen wird, ist sehr fraglich, da sich die Wartezeiten dadurch in keinster Weise verbessern würden.

In *Abbildung 15 - Grafische Beschreibung des Zuordnungsalgorithmus* wird am rechten Pfad zweimal getestet, ob eine Bühne deaktiviert ist oder nicht. Diese doppelte Abfrage sollte durch Vorschaltung der Prüfung abgeschafft werden:

1. Prüfung auf deaktivierte Bühnen
2. Ergebnismenge prüfen auf leere Pufferplätze (= potenzielle Bühnen)
3. Ergebnismenge prüfen auf leere Montageplätze (= leere Bühnen)

Entstanden ist die doppelte Abfrage durch nachträgliches Hinzufügen von Anforderungen und mangelnder Zeit zum Re-Design. Eine diesbezügliche Optimierung des Zuordnungsalgorithmus ist auch hier fraglich, da für die MAN dadurch kein Mehrwert entsteht.

Denkbar wäre jedoch die Einführung eines sogenannten "Perlenkettenchecks". Hiermit ist gemeint, dass beim Einscannen der Dächer zugleich die Einhaltung der Perlenkette überprüft wird. Wird ein Dach übersprungen, so erhält der Verteiler an der Master-GUI eine Nachricht.

Eine weitere Verbesserung ist im Bereich der Vorkommissionierung möglich. In diesem Segment werden die Kommissionierwägen mit den Werkstücken befüllt, die anschließend in die Dächer montiert werden müssen. Ein datenbankbasiertes System namens Pick-to-Light unterstützt hierbei die Lagerarbeiter. Diese scannen vor dem Befüllen des Wagens die Fahrzeugnummer, sodass alle zu kommissionierenden Teile im Lager durch eine aufleuchtende Lampe gekennzeichnet werden. Beim Entnehmen

der Teile quittiert der Mitarbeiter den Handgriff durch Betätigung eines Sensors, der wiederum einen Flag zu der zugehörigen Sachnummer in einer Oracle-Tabelle setzt.

Dies wurde bisher so gehandhabt, dass die gefüllten Kommissionierwagen zur Abholung durch die Bühnenarbeiter in einem speziellen Bereich abgestellt wurden. Durch zyklische Synchronisation der soeben genannten Oracle-Tabelle mit dem DMS konnte somit eine Anzeige in der Bühnen-GUI realisiert werden, die dem Werker mitteilt, ob die Kommissionierung des sich auf dem Pufferplatz befindlichen Dachs, bereits beendet wurde (Pufferplatzfeld wird grün hinterlegt, sobald Kommissionierung beendet ist). Somit werden unnötig gelaufene Wege des Werkers vermieden und die Kommissionierung stärker in den Ausstattungsprozess eingebunden.

An dieser Stelle wäre der Entwurf einer neuen Kommissionierungs-GUI eine Möglichkeit zur Verbesserung. Sie ist ähnlich wie die Master-GUI aufgebaut, zeigt jedoch zusätzlich die verbleibende Montagezeiten der Arbeitsbühnen an. Der Prozessfluss müsste dahingehend verändert werden, dass nicht mehr die Bühnenarbeiter den Kommissionierwagen holen, sondern die Kommissionierer den Wagen zu den entsprechenden Arbeitsbühnen bringen. Somit entsteht eine Just-In-Time Lieferung der Kommissionierwagen, die den Prozessfluss abrundet. Diese Maßnahme wurde nach Beendigung der Masterarbeit in Auftrag gegeben, um die in *4.1.5 Schwachstellenanalyse* erwähnte Wartezeit bedingt durch die Vorkommissionierung zu verringern.

Beim Umsetzkran verhält es sich ähnlich wie bei der Kommissionierung. Nachdem die Ausstattung in das Dach verbaut wurde, stellen die Monteure Ihre Montageplätze in die Ausgangsposition zurück und lösen die Bandhalterungen. Daraufhin setzt der Kranfahrer des Umsetzkran die Dächer mithilfe einer Funkfernbedienung auf das Gurtband ab, von dem sie später für das robotergesteuerte Verkleben mit dem Fahrerhaus entnommen werden. Da der Kranfahrer nur auf die Arbeitsbühnen reagiert deren Montageplätze sich in der Ausgangsposition befinden, konnten mehrfach lange und ineffiziente Umsetzarbeiten beobachtet werden. Diese entstanden beispielsweise, wenn der Umsetzkran sich auf Höhe der Arbeitsbühne 1 befand und ein Dach aus dem Montageplatz 8 entnehmen musste, das jedoch auf Höhe der Arbeitsbühne 1 auf dem Gurtband abgelegt werden musste. Während des Umsetzvorgangs wurde jedoch die Montage auf dem Montageplatz 2 beendet, dessen Dach auf Höhe der Arbeitsbühne 9 abgesetzt werden sollte. Das Umsetzen wurde erst im Anschluss durchgeführt.

Abhilfe für diesen wenig förderlichen Prozess könnte wieder eine GUI schaffen, die der Kommissionierungs-GUI ähnelt. Durch eine Maske würde der Umsetzkranfahrer erfahren, welche Bühnen als nächstes ihre Montagearbeiten beenden, um daraus ideale Krantrajektorien abzuleiten. Beispielsweise müsste er bei oben genanntem Szenario ein paar Sekunden warten und mit dem Umsetzen des Dachs aus dem Montageplatz 2 beginnen. Auch diese Maßnahme wurde nach Beendigung der Masterarbeit in Auftrag gegeben.

Eine Regelung des Prozessflusses des unteren Kranfahrers, der die Hochdächer in die Dachausstattung im ersten Stock transportiert, wäre ebenfalls sinnvoll. Zur Zeit ist es nicht genau definiert, welches der zeitweilig beiden zur Verfügung stehenden Hochdächer der untere Kranfahrer zuerst abarbeiten soll. Dadurch kommt es immer wieder vor, dass ein noch nicht benötigtes Hochdach bereits ausgestattet wurde und vor dem Gurtband am Boden abgelegt werden muss. Das zweite Hochdach, das vernachlässigt wurde, muss jedoch rasch ausgestattet werden, um einen Bandstopp zu vermeiden. Wie bei den vorigen Problemstellungen, wäre auch hier eine grafische Oberfläche sehr hilfreich. Diese müsste die Perlenkette abfragen, nach Hochdächern filtern und den relevanten Abschnitt grafisch aufgearbeitet dem unteren Kranfahrer ausgeben. Somit wäre er immer informiert, welches Dach als nächstes abgewickelt werden soll.

Eine erwähnenswerte Erweiterung des Systems besteht in der Überarbeitung der Bühnen-GUI. Zum Einen sollen die fünf testweise eingeführten Bühnen-PCs ein paar Monate nach Beendigung dieser Masterarbeit durch zehn qualitativ hochwertigere Bildschirm-PCs ausgetauscht werden. Diese werden auf Kopfhöhe montiert, um die Werker nicht zu behindern. Außerdem soll die Bedienung ausschließlich über den Touchscreen erfolgen. Da die Bühnen-GUI zuvor als Split-Screen-Lösung realisiert wurde, ist eine optische Überarbeitung nötig. Weitere Funktionalitäten wie die Einblendung der aktuellen Uhrzeit, Echtzeit-Feedback aus der Qualitätssicherung über fehlerhaft montierte Dächer, die Berechnung der Ist-/Soll-Schichtzeit, wie auch Zugriff auf das Werker-Info-System (WIS) müssen ebenfalls umgesetzt werden. Die Schnittstelle zum WIS ist vor allem für die Qualitätssicherung wichtig, da in diesem System dem Werker Hinweise auf mögliche Fehler bei der Ausstattung des soeben montierten Dachs gemacht werden. Diese Erweiterung wurde ebenfalls nach Beendigung der Masterarbeit in Auftrag gegeben.

Die letzte denkbare Erweiterung des DMS liegt in der besseren Anbindung an die restliche MAN IT-Infrastruktur. In *Abbildung 10 - Datenbankschema* ist im rechten unteren Abschnitt eine SPS-Anbindung über einen OPC-Server angedeutet. Durch diese Schnittstelle wäre der Schichtleiter nicht mehr gezwungen, tagtäglich die Schichtinformationen in das System einzugeben. Diese Informationen könnten über TCP/IP regelmäßig und automatisch aus der zuständigen SPS ausgelesen und in das DMS eingepflegt werden.

## 8 LITERATURVERZEICHNIS

1. **Finanzkrisenspecial.** [Online] [Zitat vom: 25. Januar 2011.] <http://www.finanzkrisenspecial.de/finanzkrise-ab-2007/alle-seiten.html>.
2. **Knoll, Alois.** *Vorlesung Echtzeitsysteme.* s.l.: Lehrstuhl Informatik VI an der Technischen Universität München, 2010.
3. **Becker, J., Mathas, C. und Winkelmann, A.** *Geschäftsprozessmanagement.* Heidelberg: Springer, 2009.
4. **Franz, Nadja.** *MAN Diplomarbeit: Erstellung einer Schwachstellenanalyse, Ermittlung von Ursachen und Planung von Lösungsvorschlägen zur Optimierung der Prozesse des Montageablaufs, des Kommissionierbereichs und der Produktionssteuerung.* München: s.n., 2010.
5. **Kemper, Alfons.** *Datenbanksysteme - Eine Einführung.* Oldenburg: Oldenburg (6. Aufl.), 2002.
6. **Vasters, Clemens.** *.NET Crashkurs. C#, .NET Framework, ASP.NET, VB.NET, ADO.NET, Managed C#, XML.* s.l. : Microsoft Press Deutschland (2. Aufl.), 2002.
7. **Klug, Uwe.** *Datenbank-Anwendungen entwerfen & programmieren: Von der objektorientierten Analyse bis zur SQL-Implementierung.* s.l. : W3L GmbH (2. Aufl.), 2002.
8. **Dickmann, Philipp.** *Schlanker Materialfluss: mit Lean Production, Kanban und Innovationen.* s.l.: Springer (2. Aufl.), 2008.
9. **Meyer, Urs.** *Grafische Methoden der Prozessanalyse.* s.l.: Hanser Fachbuchverlag (1. Aufl.), 2005.
10. **Kuhrmann, Marco.** *Verteilte Systeme mit .NET Remoting, Grundlagen - Konzepte - Praxis.* s.l.: Spektrum Akademischer Verlag (2. Aufl.), 2004.
11. **MAN.** [Online] 2009. [Zitat vom: 24. Mai 2010.] <http://www.man.de>.

## 9 ABBILDUNGSVERZEICHNIS

Abbildung 1 – Layout für die Dachausstattung.....	17
Abbildung 2 - Ishikawa-Diagramm für Schwachstellenursachen [4] .....	21
Abbildung 3 - Prozessfluss Soll-Zustand .....	24
Abbildung 4 - SPS-gesteuerter Systemaufbau.....	26
Abbildung 5 - config.ini einer Bühnen-GUI.....	29
Abbildung 6 - Master-GUI .....	30
Abbildung 7 - Bühnen-GUI.....	31
Abbildung 8 - Gurtband-GUI .....	33
Abbildung 9 - Daemon-GUI.....	35
Abbildung 10 - Datenbankschema .....	39
Abbildung 11 - Polling von Oracle-DB in MySQL-DB.....	40
Abbildung 12 - Anbindung der Komponenten an die hausinterne IT-Infrastruktur .....	42
Abbildung 13 - Interaktion der Komponenten.....	43
Abbildung 14 - Prozesszyklus eines Dachs.....	44
Abbildung 15 - Grafische Beschreibung des Zuordnungsalgorithmus.....	46

## 10 TABELLENVERZEICHNIS

Tabelle 1 - Ersparnis durch Einsatz des DMS .....	48
---	----

# 11 ANHANG

## 11.1 BEGRIFFS- UND ABKÜRZUNGSVERZEICHNIS

Bezeichnung	Erklärung
Kostenstelle	Ort der Kostenerstehung (z.B. 55)
Stand	Ort der Leistungserbringung (Podestbereich: Stand 03B)
Segment	Kombination aus Kostenstelle und Stand (z.B. 5503B)
Werk	Produktionsstätte
Werker	Gewerblicher Mitarbeiter der Produktionsstätte
(Arbeits-)Bühne	Arbeitsplatz für 1-2 Werker
Taktzeit	Durchschnittlich benötigte Zeit für die Produktion eines Dachs (Soll: 3:45; Ist: 3:35) (HD: 4:03; BD: 3:09)
Band	Reihenfolgesynchrone Aufreihung von Produktionspositionen
Takt	Zeitintervall von 3:45 min für die Fortbewegung des Bandes um eine Position
Stand	Örtlichkeit, an der der Arbeitsvorgang stattfindet (Podestbereich: Stand 03B)
TE-Zeit	durchschnittlich benötigte Zeit für den Einbau einer bestimmten Komponente (Montagezeit nach REFA bestimmt)
Arbeitsplan	Anweisungen mit genau definierten Handgriffen an jeweiligen Ständen mit hinterlegten TE-Zeiten
Verkaufsgruppe	Gruppierung von einzelnen Montageteilen zu einem Paket (Tür besteht aus Scheibe, Rückspiegel...)
Sachnummer	kleinste hinterlegte Einheit von Montagestücken in Verkaufsgruppe
Stückliste	Liste der zu verbauenden Verkaufsgruppen und Sachnummern
Vorkommissionierung	Ort, an dem die Kommissionierwägen bestückt werden
Verteiler	Mitarbeiter, der für die Verteilung der Dächer auf einzelne Bühnen zuständig ist
Perlenkette	festgelegte Auftrags- bzw. Fertigungsreihenfolge

## 11.2 BENUTZERHANDBUCH FÜR DMS



### *Benutzerhandbuch für DMS (Dach Management System)*



**Benjamin Brandenbourger**  
[benjamin@brandenbourger.de](mailto:benjamin@brandenbourger.de)

**Version 2.0**

# Inhaltsverzeichnis des Benutzerhandbuchs

1.	Einleitung	57
2.	Installation	57
2.1.	Konfigurationsdatei anpassen	57
2.2.	HOW-TO für Neuinstallation	59
3.	Komponenten	59
3.1.	Master	59
3.1.1.	Einstellungen	60
3.2.	Bühne	62
3.3.	Daemon	63
3.4.	Gurtband	64
3.4.1.	Status des Systems	64
3.4.2.	Statistikmodul	65
3.4.2.1.	Dach Historie	66
3.4.2.2.	Bühne Historie	67
3.4.2.3.	Leerlauf der Bühnen	68
3.4.2.4.	Wartezeit Bühne	69
3.4.3.	Schichtinformationen	69
3.4.4.	Optionen	70
3.4.4.1.	System	71
3.4.4.2.	Gurtband	72
3.4.4.3.	Bühnen	73
3.4.4.4.	Schichtinformationen	73
4.	Kommunikation der Komponenten	74

# 1. Einleitung

Die Software zur Einsteuerung der Dächer in der Dachmontage soll dazu dienen, Leerlaufzeiten von Montagebühnen zu vermeiden. Die Anwendung besteht aus vier Haupt-Komponenten:

- Master
- Bühne
- Gurtband
- Daemon

Dabei läuft das Master- wie auch das Damon-Programm auf der selben physikalischen Maschine.

Die Kommunikation zwischen den einzelnen Komponenten läuft über das MAN-interne Netzwerk (siehe 4).

## 2. Installation

Zur Installation reicht es aus, den entsprechenden Ordner (Master/Buehne/Daemon/Gurtband) von der CD auf einen MAN-PC zu kopieren. Die in dem Ordner befindliche EXE-Datei wird durch einen Doppelklick gestartet. Hierzu empfiehlt es sich, eine Verknüpfung in den Autostart-Ordner zu erstellen.

Außerdem muss auf dem Rechner das Microsoft Framework 4.0 installiert sein.

### 2.1. Konfigurationsdatei anpassen

Bei den Bühnen-GUIs muss vor dem Ausführen des Programms die Datei namens *config.ini* bearbeitet werden (Öffnen beispielsweise mit dem Editor). In der Datei muss im Abschnitt *[SYSTEM]* der Schlüssel *BuehnenNr* adäquat eingetragen werden (z.B. Rechner für Montagebühne 5/6 erhält die *BuehnenNr=5*). Außerdem muss im Abschnitt *[DB]* die Informationen der MySQL-Datenbank eingetragen werden.

```
[DB]
; Zugriff auf MySQL-DB
database = mydatabase
username = myuser
password = mypa$$w0rd
ip = 192.168.100.123

[SYSTEM]
BuehnenNr = 3
```

**Abbildung 1: Beispiel für config.ini der buhne.exe**

Da die Daemon-Komponente zusätzliche Funktionen ausführt, wie das Synchronisieren der MAN-Datenbank mit der MySQL-Datenbank, sind in der zugehörigen *config.ini* mehr Informationen hinterlegt:

```
[DB]
; Zugriff auf MySQL-DB
database = mydatabase
username = myuser
password = mypa$$w0rd
ip = 127.0.0.1

[ORADB]
; Zugriff auf Oracle-DB
Database = XE
Username = ora-user
Password = ora-pw

[SYNCHRONISATION]
; Letzte Synchronisationszeitpunkte
Pick2light=02.11.2010 08:08:39
Perlenkette=24.10.2010 12:48:18
Ringbahn=24.10.2010 12:48:46
Spoiler=24.10.2010 12:44:18
Tezeit=22.09.2010 18:53:23

[KONFIGURATION]
Betriebsstellenfilter=4551PDT
```

**Abbildung 2: config.ini der Daemon-Komponente**

Im Abschnitt *[DB]* ist erneut die Schnittstelle zur MySQL-Datenbank definiert. Dabei wird die IP-Adresse 127.0.0.1 (=localhost) benutzt, da die Daemon-Komponente im Normalfall auf dem Rechner installiert ist, auf dem auch die MySQL-Datenbank läuft. Falls dies nicht der Fall sein sollte, müsste die IP-Adresse an dieser Stelle angepasst werden.

Im Abschnitt *[ORADB]* wird der Zugang zur Oracle-Datenbank der MAN definiert. Dieser Zugang wurde über einen Workflow-Antrag erstellt.

Im Abschnitt *[SYNCHRONISATION]* werden zu informativen Zwecken der letzte Synchronisationszeitpunkt der Tabelle festgehalten.

## 2.2. HOW-TO für Neuinstallation

Bühnen-PC:

1. Microsoft Framework 4.0 installieren (Admin-Rechte benötigt)
2. Bühnen-Ordner auf den Rechner kopieren
3. Verknüpfung zur *buehne.exe* in den Autostartordner ablegen
4. *Config.ini* adäquat anpassen

Master-PC:

1. XAMPP installieren (beinhaltet MySQL-DB und Apache-Server)
2. In phpMyAdmin eine Datenbank namens „roof-db“ anlegen
3. Backup-SQL-Skript zum Aufbau der DB und Tabellen ausführen
4. Microsoft Framework 4.0 installieren (Admin-Rechte benötigt)
5. Master- und Daemon-Ordner auf den Rechner kopieren
6. Verknüpfungen zur *WindowsFormsApplication1.exe* (=Master) und *daemon.exe* in den Autostartordner ablegen
7. Ggf. *config.ini* adäquat anpassen (eigtl. nicht notwendig, da auf localhost zugegriffen wird)

## 3. Komponenten

### 3.1. Master

Die Master-Anwendung läuft auf dem Master-PC der neben dem Aufnahmekran positioniert ist. Am Master-PC ist ein Barcode-Scanner über USB angeschlossen mit dem die Ausdrücke der Dächer eingescannt werden. Dabei wird die Fahrzeugnummer in das Eingabefeld eingelesen (siehe Screenshot „06S-5074“ grün hinterlegt) und die optimale Bühnennummer ausgegeben. Das Dach wird sogleich auf dem jeweiligen Pufferplatz eingebucht.

Eine manuelle Eingabe der Fahrzeugnummer ist ebenfalls möglich. Dabei ist darauf zu achten, dass die Eingabe entweder durch Drücken der Enter-Taste oder durch ein 3-sekündiges Warten validiert wird. Danach wird die Eingabe automatisch ausgelesen.

	Montageplatz	Pufferplatz	Anzahl MA
Bühne 1:			1
Bühne 2:			3
Bühne 3:		06X-1235	2
Bühne 4:	06S-5074		1
Bühne 5:	23U-2137		0
Bühne 6:	06X-T479		0
Bühne 7:	35K-3498		0
Bühne 8:	98D-2439		0
Bühne 9:			0
Bühne 10:		74L-4423	0

### 3.1.1. Einstellungen

Durch Klicken auf den „Einstellungen“-Button gelangt man, nach der erfolgreichen Eingabe des Passworts, zu einer Oberfläche, in der diverse Einstellungen vorgenommen werden können.

	Montageplatz	Pufferplatz	Anzahl MA
Bühne 1:		06S-5074	1
Bühne 2:			3
Bühne 3:		06X-1235	2
Bühne 4:			1
Bühne 5:	23U-2137		0
Bühne 6:	06X-T479		0
Bühne 7:	35K-3498		0
Bühne 8:	98D-2439		0
Bühne 9:			0
Bühne 10:		74L-4423	0

Speichern und Schließen

Durch einen Rechtsklick auf einen Montageplatz, erscheint ein Kontextmenü, das die Vorgänge „Dach ausbuchen“ und „Dach auf dem Pufferplatz zurückbuchen“ zur Verfügung stellt. Die Funktionen sind selbsterklärend.

Durch einen Rechtsklick auf einen Pufferplatz, erscheint ein Kontextmenü, das die Vorgänge „Dach aus System löschen“, „Dach auf Montageplatz einbuchen“ und „Dach umbuchen nach Pufferplatz“ anzeigt. Letzteres ermöglicht das Umbuchen eines Dachs auf einen anderen leeren Pufferplatz.

Falls ein Dach versehentlich in das System eingebucht wurde, können alle dazugehörigen Einträge mit „Dach aus System löschen“ gelöscht werden.

Die Funktion „Dach auf Montageplatz einbuchen“ ist selbsterklärend.

Das Umbuchen eines Dachs auf einen anderen leeren Pufferplatz kann ebenfalls durch Drag&Drop durchgeführt werden.

Alle Kontextmenü-Vorgänge sind kontextabhängig, d.h. nicht durchführbare Vorgänge werden von vornherein ausgegraut (z.B. Dach auf Pufferplatz 1 umbuchen, obwohl sich dort bereits ein Dach befindet).

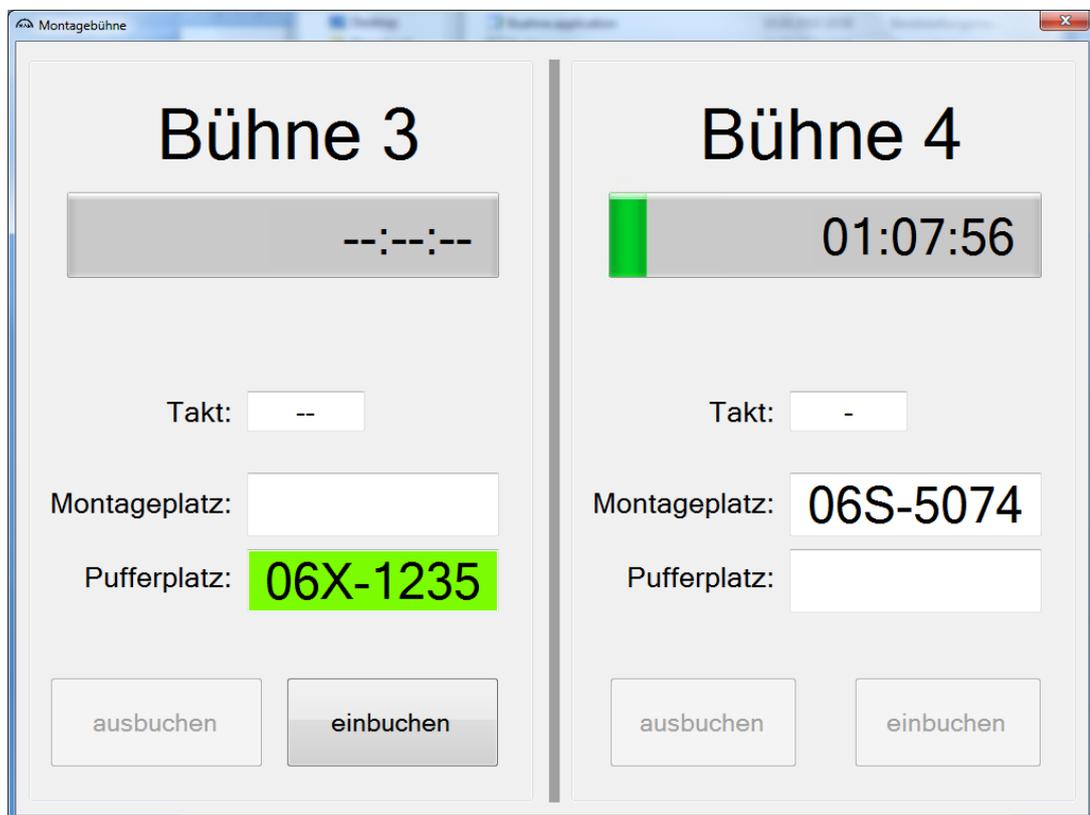
Änderungen werden sofort in der Datenbank gespeichert.

Auf der rechten Seite dieser grafischen Oberfläche wird dem Benutzer die Möglichkeit gegeben, die Anzahl der Mitarbeiter an den jeweiligen Bühnen einzustellen. Die Anzahl kann durch das Betätigen der jeweiligen Pfeiltasten rechts neben der Ziffer geändert werden. Zum Deaktivieren einer Bühne muss die Anzahl auf 0 gesetzt werden. Dieser Zustand wird zusätzlich durch einen grauen Hintergrund verdeutlicht.

Die Einstellungen der Mitarbeiteranzahl werden erst durch das Klicken auf „Speichern und Schließen“ wirksam.

## 3.2. Bühne

Die Bühnen-GUI ist immer zweigeteilt. Die linke Seite zeigt die in der *config.ini* angegebene *Buehennr* an, die rechte Seite die darauf folgende Bühne.



Unter der Bühnenummer ist ein Zeitbalken mit Zeitanzeige angebracht, der die Soll-Zeit für ein Dach herunter zählt. Dabei wurde diese Sollzeit nach den MAN-spezifischen Kriterien aus der TE-Zeit umgerechnet (siehe Optionen unter 3.4).

Falls sich ein Dach auf dem Pufferplatz befindet, wird dessen Fahrzeugnummer in dem entsprechendem Pufferplatz-Feld angezeigt. Die Hintergrundfarbe dieses Feldes kann

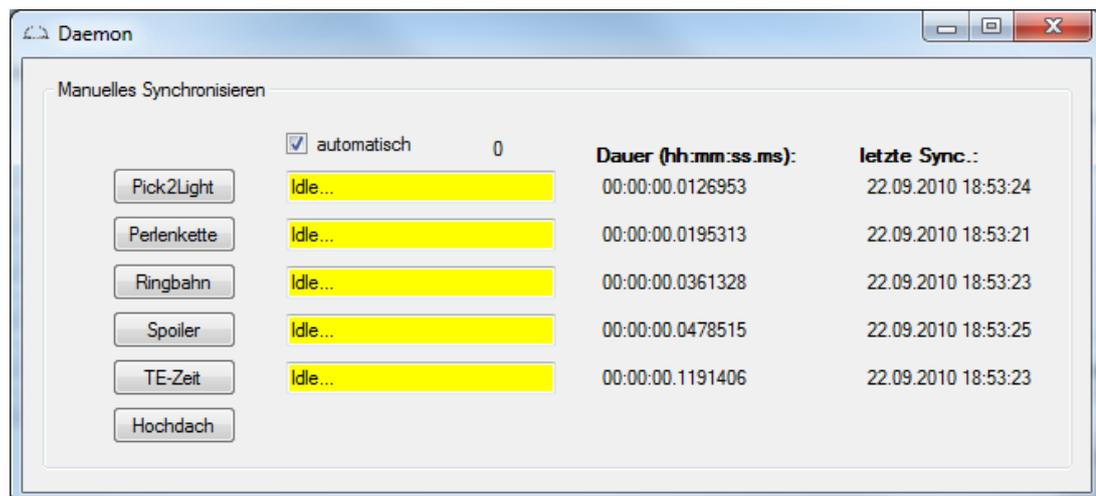
zwischen weiß (=Kommissionierwagen noch nicht angefangen), orange (=Kommissionierwagen wird zur Zeit befüllt) und grün (=Kommissionierwagen steht bereit zur Abholung) wechseln. Des Weiteren besteht die Möglichkeit, durch einen Klick auf den „einbuchen“-Button das Dach auf den Montageplatz einzubuchen. Dabei wird der Countdown automatisch gestartet.

Zusätzlich wird die Position des Fahrerhauses auf der Ringbahn im Takt-Fenster angezeigt. Dieses Fenster beginnt bei gewissen Taktunterschreitungen (siehe Optionen unter 3.4) zu Blinken und wechselt die Hintergrundfarbe in orange bzw. rot.

Nachdem die Montagezeit eines Dachs abgelaufen ist, wird der „ausbuchen“-Button aktiviert. Durch Betätigen dieses Buttons wird das Dach vom Montagplatz ausgebucht.

### 3.3. Daemon

Der Daemon ist ein Programm, das im Hintergrund auf dem Master-PC laufen muss. Er dient zur Synchronisierung der Daten zwischen dem MAN-Netzwerk mit der Oracle-Datenbank und der Software zur Einsteuerung der Dächer.



In der grafischen Oberfläche ist darauf zu achten, dass der Haken bei „automatisch“ gesetzt ist. Dadurch wird ein automatischer Datenabgleich in frei definierbaren Abständen gewährleistet. Die Abstände können unter Optionen in der Gurtband-GUI eingestellt werden (siehe 3.4.4). Zur Verdeutlichung der automatischen Synchronisierung läuft ein Timer im oberen Teil des Fensters mit.

Außerdem kann durch Betätigen der Buttons auf der linken Seite die Synchronisation manuell angestoßen werden. Während einer Synchronisation wechselt die Status-

Farbe der entsprechenden zu synchronisierenden Tabelle auf grün. Im Normalzustand ist der Hintergrund auf gelb gestellt.

Aus informativen Gründen wird die Dauer, wie auch der Zeitpunkt der letzten Synchronisierung mit angezeigt.

### 3.4. Gurtband

Die Gurtband-GUI soll die Belegung des Gurtbands in nahezu Echtzeit abbilden. Dabei sind die 16 Plätze des Bands erkennbar, die jeweils leer oder mit einer Fahrzeugnummer gekennzeichnet sind.



Die einzelnen Farben sind wie folgt zu interpretieren:

- Dunkelrot: Dach wurde noch nicht eingescannt.
- Rot: Dach wurde noch nicht angefangen.
- Gelb: Dach wird gerade montiert.
- Grün: Dach ist fertig montiert.

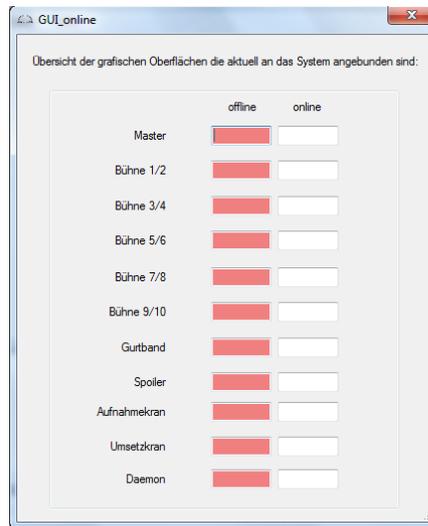
Diese Informationen können in der Gurtband-GUI ebenfalls unter *Hilfe* → *Legende* nachgelesen werden.

Unterhalb der Fahrzeugnummern wird mit angegeben, an welcher Bühne das Dach gerade montiert wird bzw. wurde und ob es sich um ein Hoch- oder Blechdach handelt.

#### 3.4.1. Status des Systems

Unter *Hilfe* → *Status des Systems* kann der Benutzer die Status der einzelnen Komponenten überprüfen. Je nachdem, ob eine Komponente eine Verbindung zum System hergestellt hat, wird der dazugehörige Status offline/online angepasst. Somit

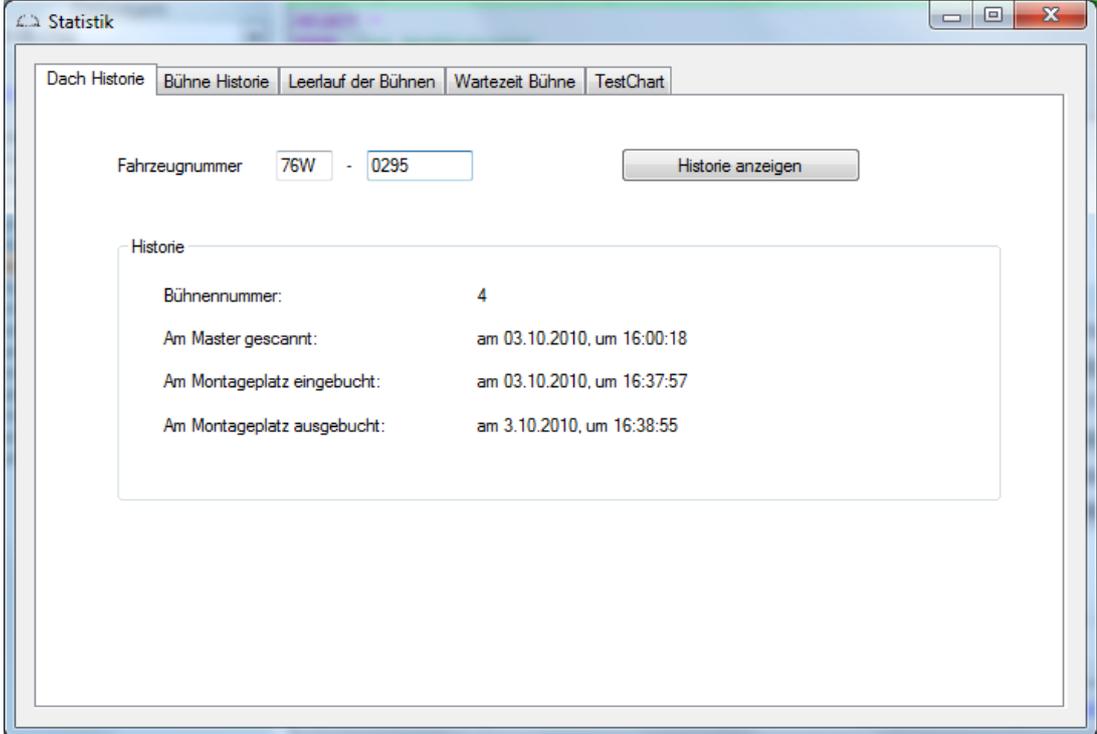
kann beispielsweise sichergestellt werden, dass die Komponente „Spoiler-GUI“ auch tatsächlich geöffnet ist und der Spoilermonteur die Montageaufforderung auch erhält.



### 3.4.2. Statistikmodul

Über *Datei* → *Statistikmodul öffnen* gelangt der Benutzer nach der erfolgreichen Passworteingabe in das Statistikmodul. Dieses ist in verschiedene Registerkarten unterteilt, die wie folgt genauer beschrieben werden.

### 3.4.2.1. Dach Historie



The screenshot shows a software window titled 'Statistik' with several tabs: 'Dach Historie', 'Bühne Historie', 'Leerlauf der Bühnen', 'Wartezeit Bühne', and 'TestChart'. The 'Dach Historie' tab is active. It features two input fields for 'Fahrzeugnummer' containing '76W' and '0295', separated by a hyphen. To the right is a button labeled 'Historie anzeigen'. Below this is a section titled 'Historie' containing a table with the following data:

Historie	
Bühnennummer:	4
Am Master gescannt:	am 03.10.2010, um 16:00:18
Am Montageplatz eingebucht:	am 03.10.2010, um 16:37:57
Am Montageplatz ausgebucht:	am 3.10.2010, um 16:38:55

In den Eingabefeldern kann die Fahrzeugnummer eines Dachs eingegeben werden. Durch Betätigen der Enter-Taste oder des „Historie anzeigen“-Buttons wird im System nach diesem Dach gesucht. Falls zu diesem Dach Informationen gefunden wurden, werden diese aufgegliedert nach Bühnennummer und Montage-Uhrzeit aufgelistet.

### 3.4.2.2. Bühne Historie

Statistik

Dach Historie Bühne Historie Leerlauf der Bühnen Wartezeit Bühne TestChart

**04.10.2010**

Oktober 2010

	Mo	Di	Mi	Do	Fr	Sa	So
39	27	28	29	30	1	2	3
40	4	5	6	7	8	9	10
41	11	12	13	14	15	16	17
42	18	19	20	21	22	23	24
43	25	26	27	28	29	30	31
44	1	2	3	4	5	6	7

Heute: 06.10.2010

Historie

Bühnennummer: 4

Schichtnummer: 2

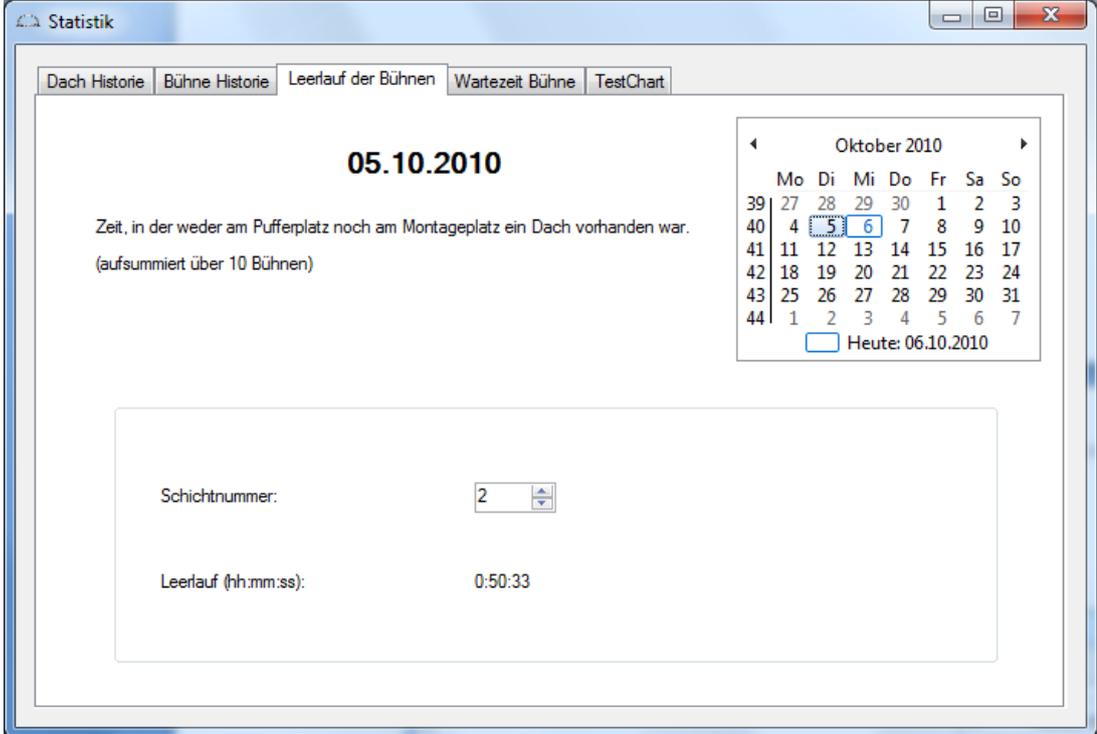
Anzahl der Mitarbeiter: 1

bisherige Arbeitszeit (min.): 66

Zeit, während der weder der Montageplatz noch der Pufferplat belegt waren (=Leerlauf): 13:9

Zuerst sollte im linken oberen Teil des Fensters das gewünschte Datum ermittelt werden. Anschließend kann die Bühnennummer wie auch Schichtnummer spezifiziert werden. Falls zu dieser Suchkombination keine Daten hinterlegt sind, wird dies im Fenster durch eine Nachricht angezeigt. Falls Daten hinterlegt sind, werden diese aufbereitet und angezeigt. Dabei entspricht die Minutenangabe („bisherige Arbeitszeit (min.)“) der Arbeitszeit, die an der zuvor spezifizierten Bühne bisher herunter gezählt wurde. Die „Leerlauf“-Zeit entspricht der Zeit in der weder der Montageplatz, noch der Pufferplatz mit einem Dach belegt waren. In dieser Zeit konnte der Mitarbeiter nicht produktiv sein.

### 3.4.2.3. Leerlauf der Bühnen



Statistik

Dach Historie Bühne Historie **Leerlauf der Bühnen** Wartezeit Bühne TestChart

**05.10.2010**

Zeit, in der weder am Pufferplatz noch am Montageplatz ein Dach vorhanden war.  
(aufsummiert über 10 Bühnen)

Oktober 2010

	Mo	Di	Mi	Do	Fr	Sa	So
39	27	28	29	30	1	2	3
40	4	5	6	7	8	9	10
41	11	12	13	14	15	16	17
42	18	19	20	21	22	23	24
43	25	26	27	28	29	30	31
44	1	2	3	4	5	6	7

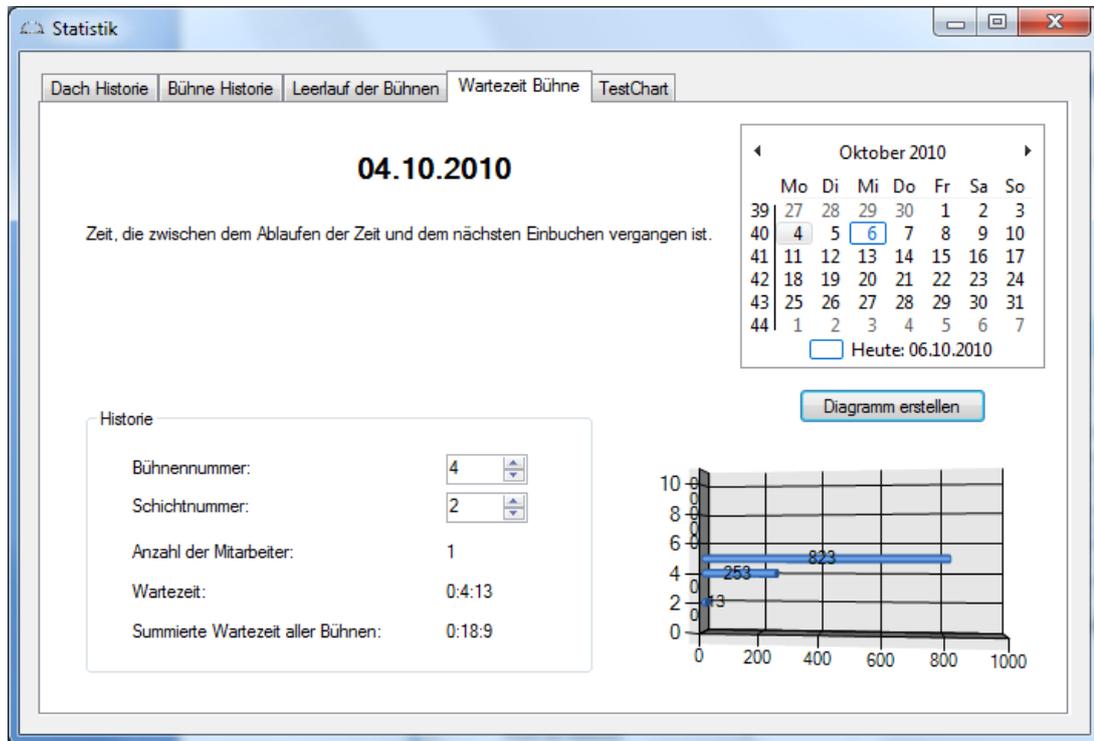
Heute: 06.10.2010

Schichtnummer: 2

Leerlauf (hh:mm:ss): 0:50:33

In dieser Maske kann, wie zuvor auch, ein Datum, wie auch eine Schichtnummer spezifiziert werden. Anschließend wird die Leerlauf-Zeit, summiert über alle 10 Bühnen, ermittelt.

### 3.4.2.4. Wartezeit Bühne



Unter der Registerkarte „Wartezeit Bühne“ kann nach Auswahl des Datums, der Bühnennummer und der Schichtnummer angezeigt werden, wie viel Zeit zwischen dem Ausbuchen und dem nächsten Einbuchen vergangen ist. Dies ist z.B. der Fall, wenn eine Bühne ständig länger als die vorgegebene Montagezeit braucht. Ein anderer Fall wäre die Abwesenheit der Mitarbeiter, sobald die Montagezeit abgelaufen ist. Neben der bühnen-spezifischen Wartezeit wird auch die summierte Wartezeit aller Bühnen angezeigt.

Unterhalb des Kalenders befindet sich ein Button mit der Aufschrift „Diagramm erstellen“. Durch Betätigen dieses Buttons wird ein Balkendiagramm erstellt, das die Wartezeiten der 10 Montagebühnen grafisch darstellt.

### 3.4.3. Schichtinformationen

Über *Datei* → *Schichtinformationen* gelangt man in eine Maske, in der die Schichtinformationen hinterlegt werden können.

**Schichtinformationen**

**06.10.2010**  
Informationen bereits hinterlegt

Schichtnummer: 1

Schichtbeginn: 6 : 16

1. Pause: 9 : 0

2. Pause: 11 : 30

3. Pause: 0 : 0

Schichtende: 15 : 0

*Dauer:* 15 Min.  
30 Min.  
0 Min.

Oktober 2010

	Mo	Di	Mi	Do	Fr	Sa	So
39	27	28	29	30	1	2	3
40	4	5	6	7	8	9	10
41	11	12	13	14	15	16	17
42	18	19	20	21	22	23	24
43	25	26	27	28	29	30	31
44	1	2	3	4	5	6	7

Heute: 06.10.2010

Speichern Abbrechen

Dabei wählt man zuerst rechts das jeweilige Datum und anschließend die Schichtnummer aus. Sobald alle Daten eingegeben wurden, kann man diese im System durch einen Klick auf dem „Speichern“-Button eingeben. Alternativ kann auch die Tastenkombination „Alt+S“ benutzt werden.

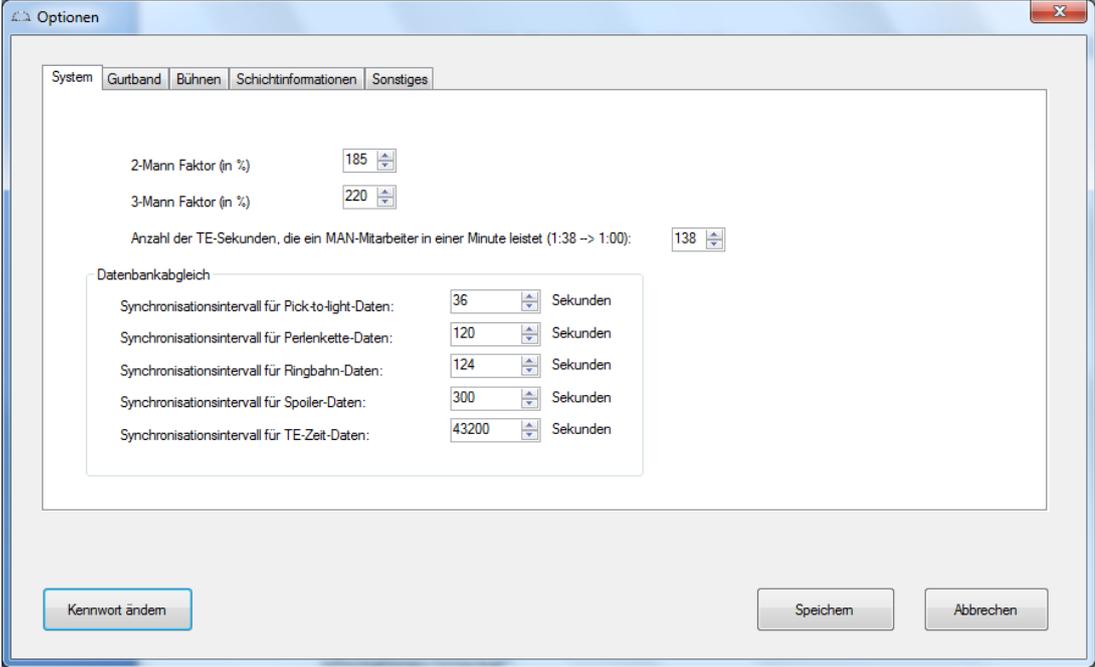
Falls zu einem Datum bereits Schichtinformationen hinterlegt wurden, wird dies unterhalb des Datums textuell mitgeteilt (siehe Screenshot: „Informationen bereits hinterlegt“). Falls noch keine Uhrzeiten zu einer Schicht abgespeichert wurden, lautet der Text „keine Informationen hinterlegt“.

#### 3.4.4. Optionen

Über *Datei*→*Optionen* gelangt man nach erfolgreicher Passworteingabe in die Optionen-Maske, die wiederum in mehrere Registerkarten unterteilt ist. Änderungen können bei jeder Registerkarte mit „Speichern“ quittiert werden. Dabei werden Änderungen in allen Registerkarten auf einmal gespeichert.

Durch Betätigen des Buttons „Kennwort ändern“ kann das Systemkennwort geändert werden. Dazu muss das aktuelle Kennwort, das neue Kennwort und eine Bestätigung des neuen Kennworts eingegeben werden. Das Kennwort gilt global; dies bedeutet, dass eine Änderung auch das Kennwort in der Master-GUI betrifft.

### 3.4.4.1. System



Optionen

System | Gurtband | Bühnen | Schichtinformationen | Sonstiges

2-Mann Faktor (in %) 185

3-Mann Faktor (in %) 220

Anzahl der TE-Sekunden, die ein MAN-Mitarbeiter in einer Minute leistet (1:38 -> 1:00): 138

Datenbankabgleich

Synchronisationsintervall für Pick-to-light-Daten: 36 Sekunden

Synchronisationsintervall für Perlenkette-Daten: 120 Sekunden

Synchronisationsintervall für Ringbahn-Daten: 124 Sekunden

Synchronisationsintervall für Spoiler-Daten: 300 Sekunden

Synchronisationsintervall für TE-Zeit-Daten: 43200 Sekunden

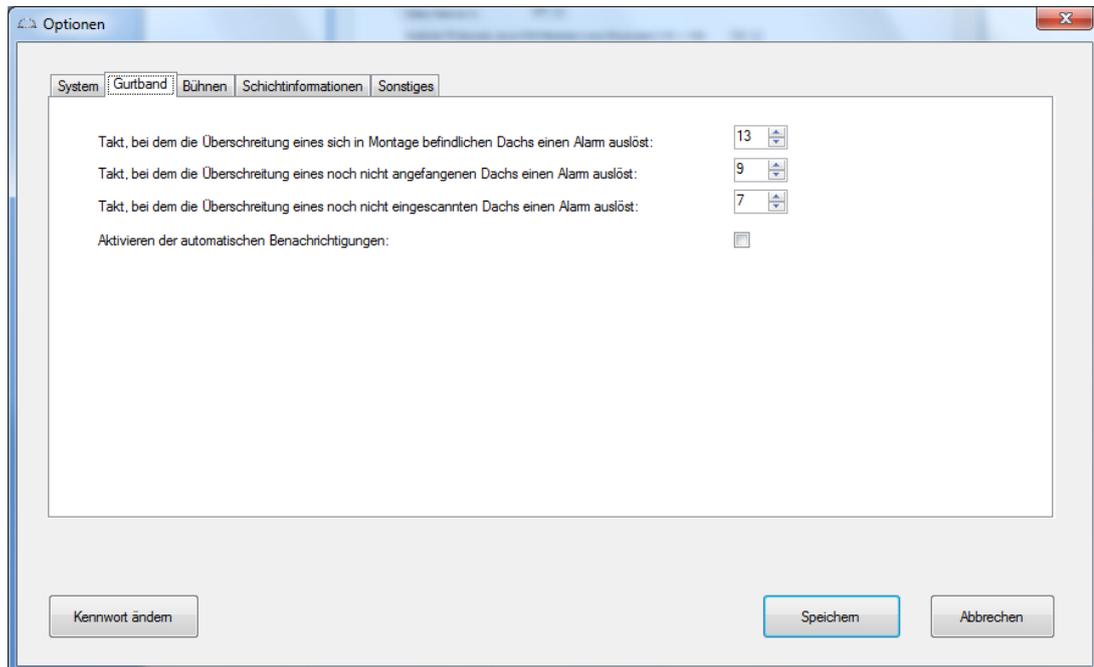
Kennwort ändern | Speichern | Abbrechen

In dieser Registerkarte können allgemeine Einstellungen zum System vorgenommen werden. Beispielsweise kann der 2- bzw. 3-Mann-Faktor und der MAN-Faktor definiert werden. Diese Faktoren werden bei der Zeitermittlung in folgender Weise einbezogen:

Montagezeit an Bühne = (summierte TE-Zeit / MAN-Faktor) / 2-Mann-Faktor

Unter *Datenbankabgleich* können die Synchronisationsintervalle zwischen der MAN-Oracle-Datenbank und dem System angepasst werden. Es empfiehlt sich eher größere Intervalle zu wählen, damit die MAN-Oracle-Datenbank nicht unnötigerweise belastet wird.

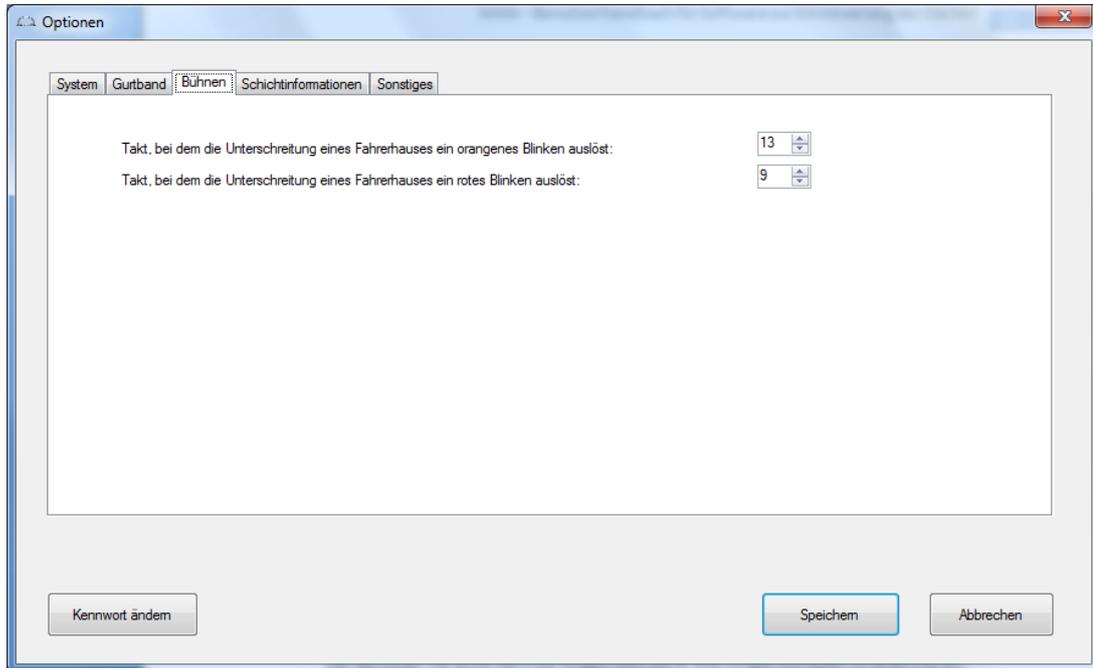
### 3.4.4.2. Gurtband



In dieser Registerkarte kann definiert werden, zu welchem Zeitpunkt die Gurtband-GUI Warnungen anzeigen soll. Diese Warnungen können durch Unterschreitung eines Taktes auf dem Ausstattungsband gesteuert werden. Zu den 3 Status (Dach nicht eingescannt, Dach nicht angefangen, Dach in Montage) können individuelle Warnzeitpunkte definiert werden.

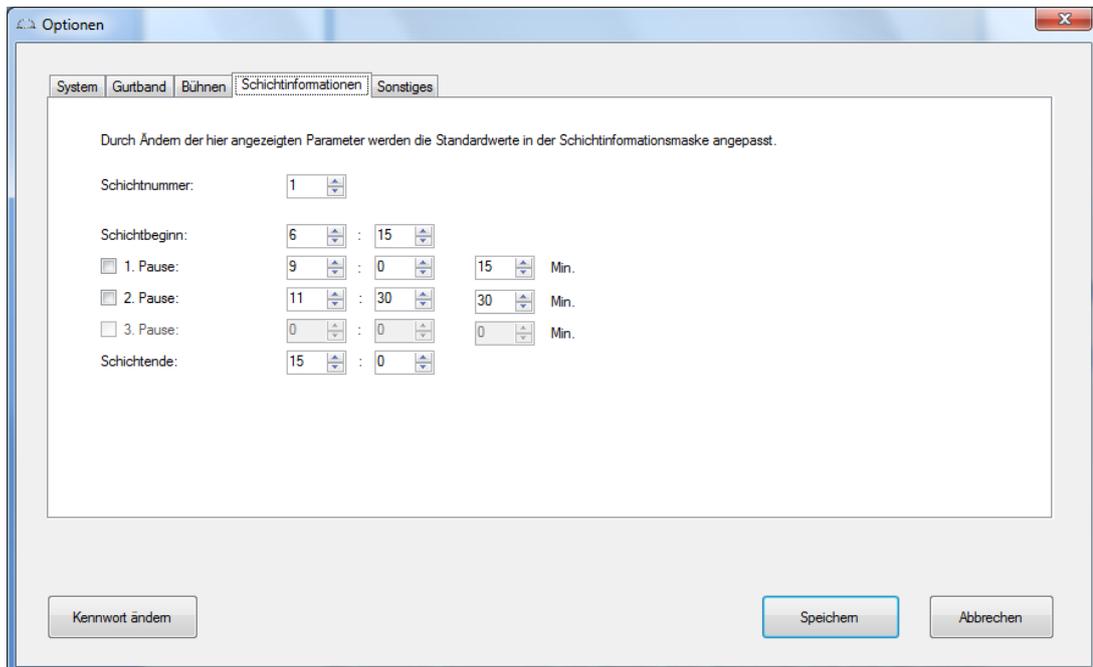
Der Benutzer hat außerdem die Möglichkeit diese Warnungen zu deaktivieren.

### 3.4.4.3. Bühnen



Ähnlich wie bei der Registerkarte „Gurtband“, kann auch für die Bühnen-GUI der Zeitpunkt definiert werden, an dem die Taktanzeige orange bzw. rot zu blinken beginnt.

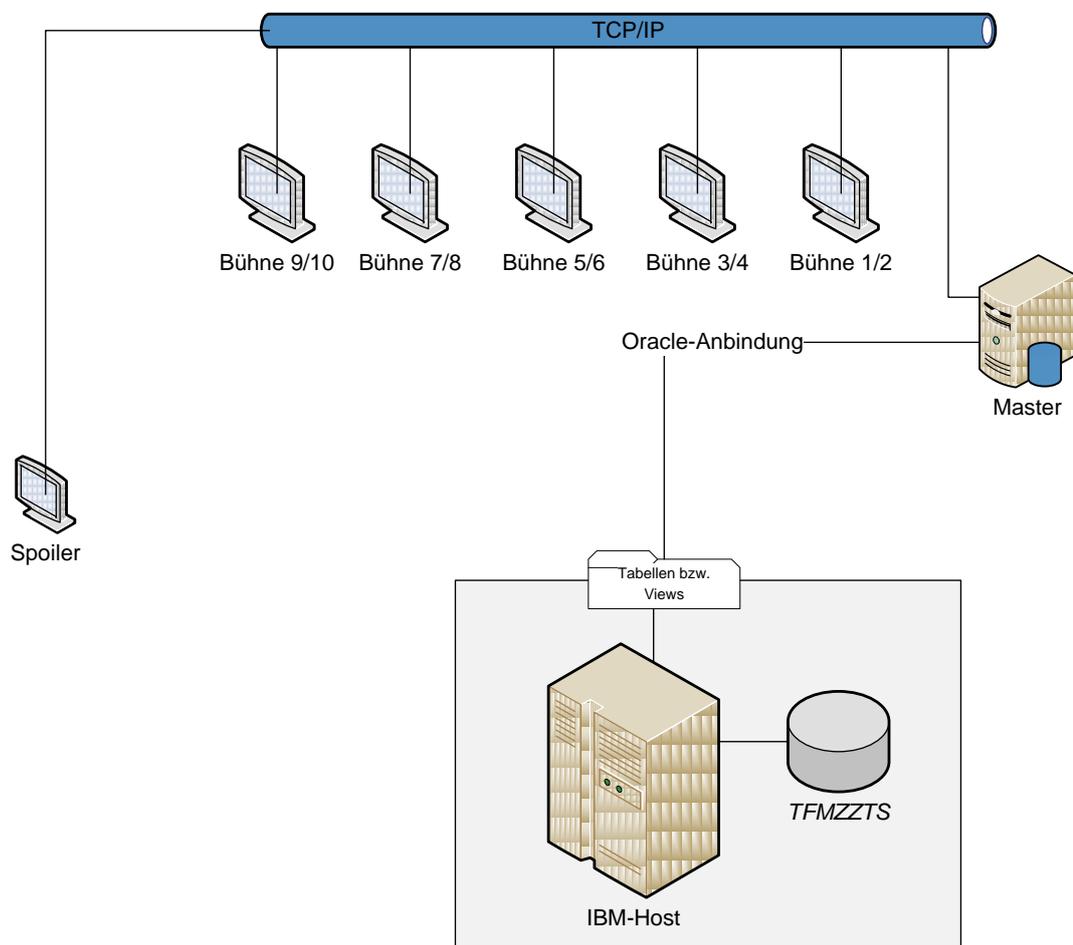
### 3.4.4.4. Schichtinformationen



In dieser Registerkarte können die Standardwerte der Schichtinformationen definiert werden. Diese Funktion soll eine höhere Benutzerfreundlichkeit sicherstellen, da der Benutzer bei der Eingabe der oft sehr ähnlichen Schichtinformationen (siehe 3.4.3. Schichtinformationen) die vorgegebenen Werte nur bestätigen muss.

## 4. Kommunikation der Komponenten

Die einzelnen Komponenten wie Master- bzw. Bühnen-PC und auch die erstellte Datenbank, die alle mit dynamischen IP-Adressen versehen sind, müssen miteinander kommunizieren. Hierzu wurde folgende Anbindung realisiert:



**Abbildung 3: Anbindung der Komponenten an das hausinterne MAN-LAN**

Die Anbindung der einzelnen Rechner läuft über das MAN-Hausnetz.