

Emulation und Optimierung
Testumgebung zur Inbetriebnahme
von Logistiksoftware

Dr.-Ing. Robert Schulz

Technologies - Systems - Solutions

Themen

A Dürr Automotion GmbH

- Organisation / Produktspektrum

B Ausgangssituation

- Motivation
- DVRS (Dynamic Vehicle Re-Sequencing)
- Lösungsansatz

C Der Ansatz für die Emulation

D Die Architektur

- Testumgebung Emulation / Leitrechner
- Aufgaben der Komponenten
- Ausschnitt aus der Emulation

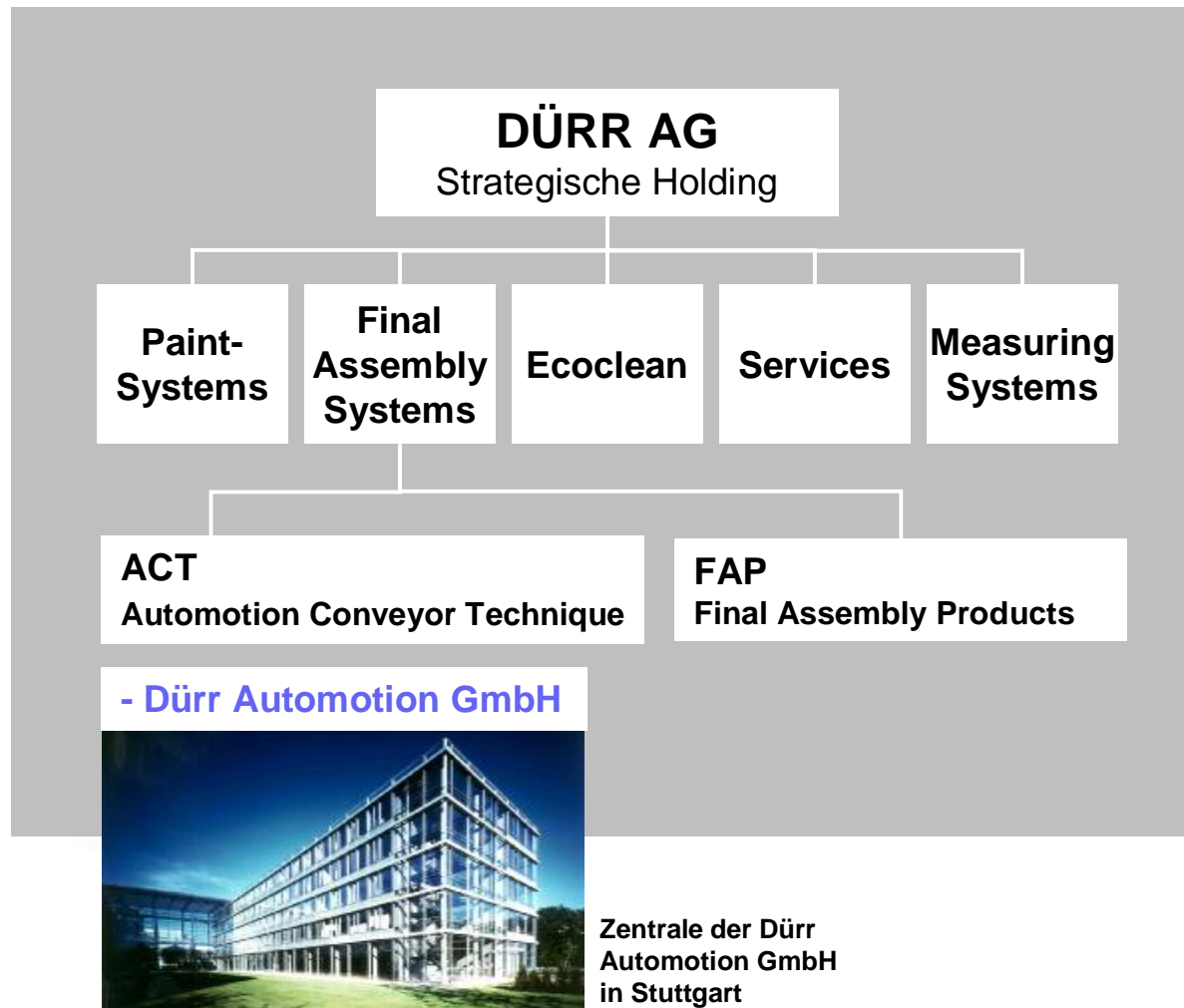
D Emulation

- Nutzen
- Wann lohnt Emulation?

A Dürr Automation GmbH



Die DÜRR AG



Produktline Fördertechnik

Flurförderer



- Skidförderer
- Plattformförderer
- Plattenbandförderer
- Gurtförderer
- Inverted Power&Free
- Inverted Twin-Trolley-System (TTS)

Hängeförderer



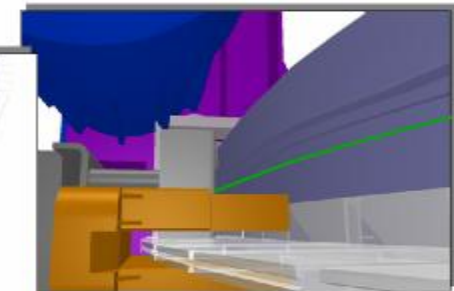
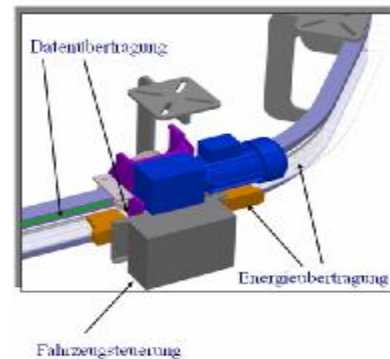
- Elektrohängebahn
- TTS
- Pendelförderer
- Power & Free
- Kettenförderer
- Plattform-TTS

Regalbediengeräte Flächenlager



Movitrans®

Berührungslose Informations- und Energieübertragung

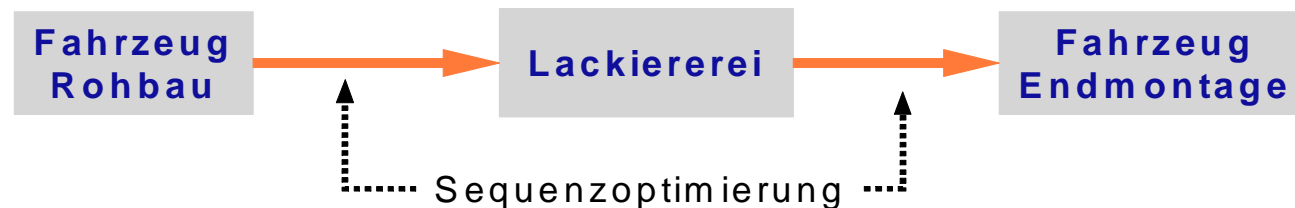


Einsatzfelder: EHB, FTF

B Ausgangssituation

DVRS – Dynamic Vehicle Re-Sequencing / Hyundai Alabama

Motivation:



- Durch Dynamic Vehicle Re-Sequencing sollen Sequenzbrüche, die durch Verwirbelung in der Lackiererei entstehen, ausgeglichen werden.
- Die Steuerung und Optimierung erfolgt im DVRS-Leitrechner.

Anforderung:

Sowohl die Inbetriebnahme der Steuerungssoftware des Leitrechners als auch die Anpassung des Optimierungsverfahren sollen unabhängig von der realen Anlage möglich sein.

B Ausgangssituation



DVRS – Dynamic Vehicle Re-Sequencing / Hyundai Alabama

Lösungsansatz:

- ; Mit Hilfe von Tracking-Punkten werden Karossen in der Anlage verfolgt, so dass ein Abbild des Anlagenzustandes entsteht. Das Abbild liefert die Emulation.
- ; Auf der Grundlage des Anlagenzustandes und den Randbedingungen zur Reihenfolgebildung trifft der DVRS-Leitrechner Re-Sequencing und Materialflussentscheidungen.
- ; Alle Materialfluß Entscheidungen des Leitrechners werden im Emulationsmodell umgesetzt. Damit ist eine Bewertung der Reihenfolgegüte und des Optimierungsverfahren nach jedem Re-Sequencing möglich.

B Ausgangssituation

Möglichkeiten zur Beeinflussung der Reihenfolgesequenz

Substitution

Zwei Karossendatensätze mit gleichen Varianten werden ausgetauscht.
Beispiel für eine Variante: mindestens Rohbautyp und Farbe sind gleich.

Eine Karosse, die z. B. durch Nacharbeit in der Sequenz weiter nach hinten gelangt ist, kann mit einer Karosse gleicher Variante, die in der Sequenz weiter vorne steht, ausgetauscht werden.

Auslagern aus dem Painted Body Storage

Welche Karosse wann aus dem PBS ausgelagert wird, entscheidet der DVRS-Leitrechner. Die hier gebildete Sequenz soll dabei der geplanten Montagesequenz entsprechen.

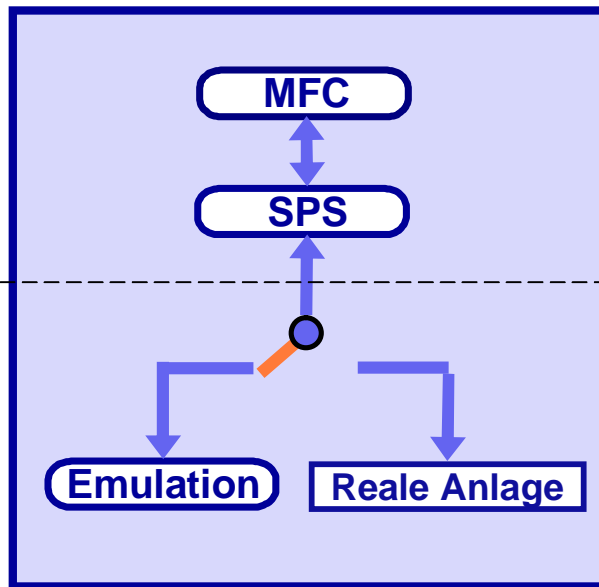
Update

Der Karossendatensatz einer Karosse wird überschrieben.
Dadurch kann z. B. das Attribut „urgent“ einer Karosse gesetzt werden. In diesem Fall überholt die Karosse via *Fast Track* physikalisch andere Karossen.

Der Ansatz für die Emulation

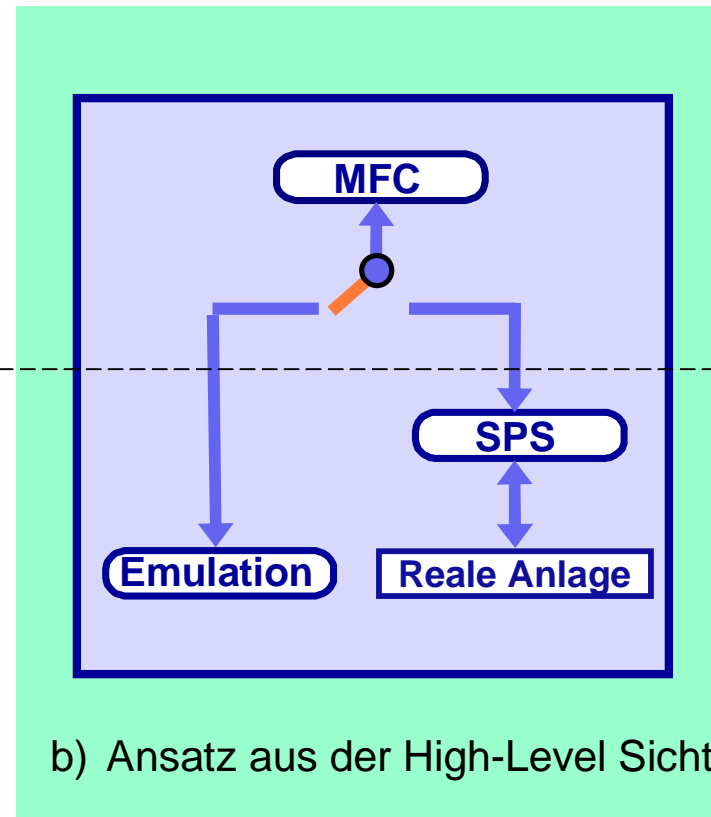


Low Level / High Level



a) Ansatz aus der Low-Level Sicht

Testen von SPS-Steuerungen,
oder kleine Anlagenbereiche wie
z. B. Montagezellen

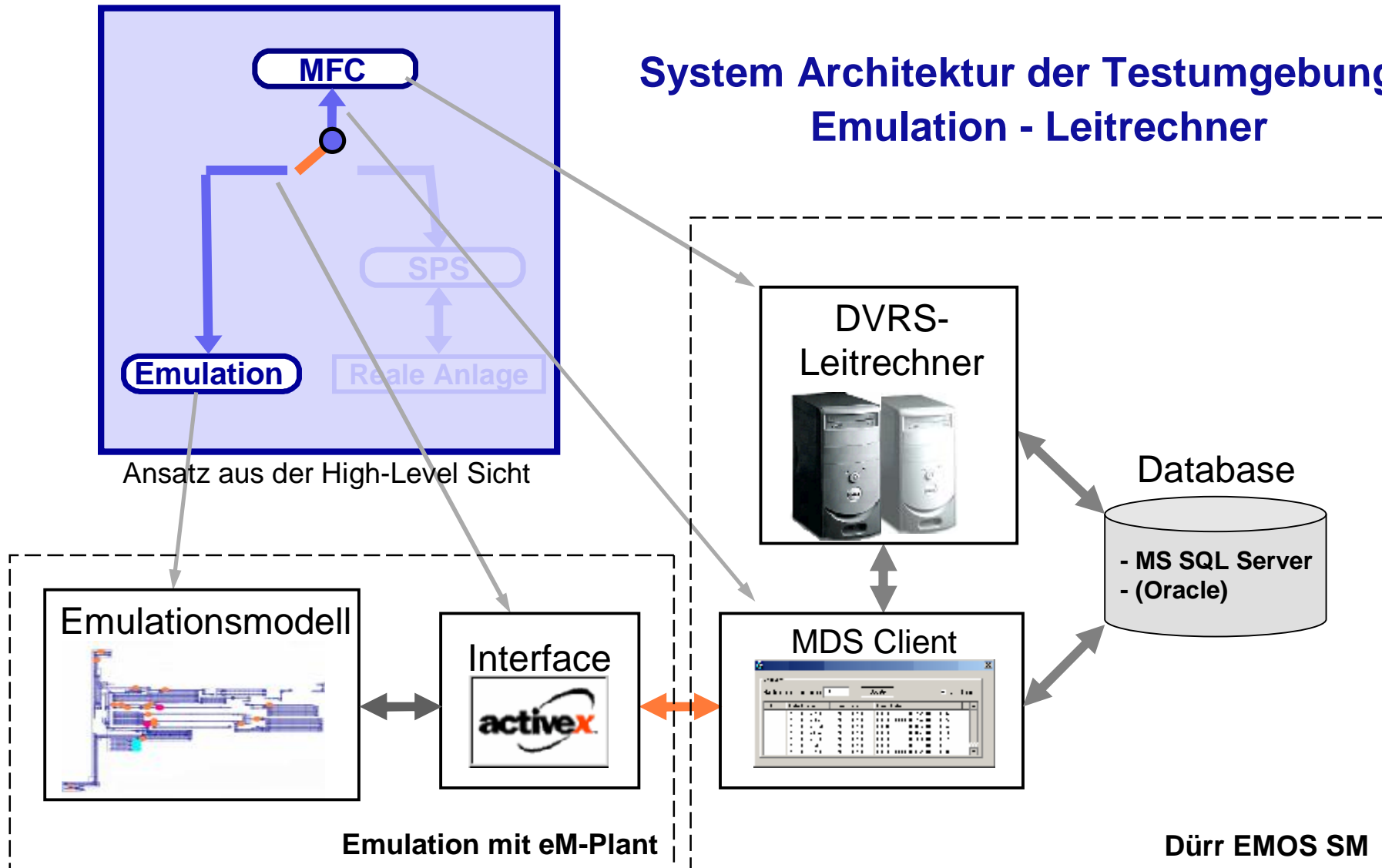


b) Ansatz aus der High-Level Sicht

Testen von Steuerstrategien des
Leitrechners,
z. B. Sequenzoptimierung

MFC = Materialflow-Controller / Leitreehner

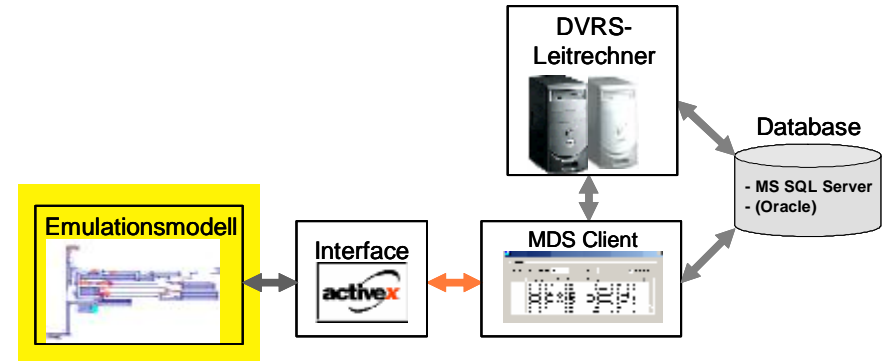
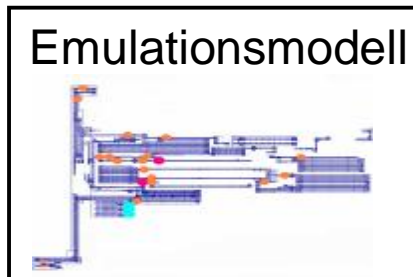
System Architektur der Testumgebung Emulation - Leitreechner



Die System-Architektur



Aufgabe der Komponente:



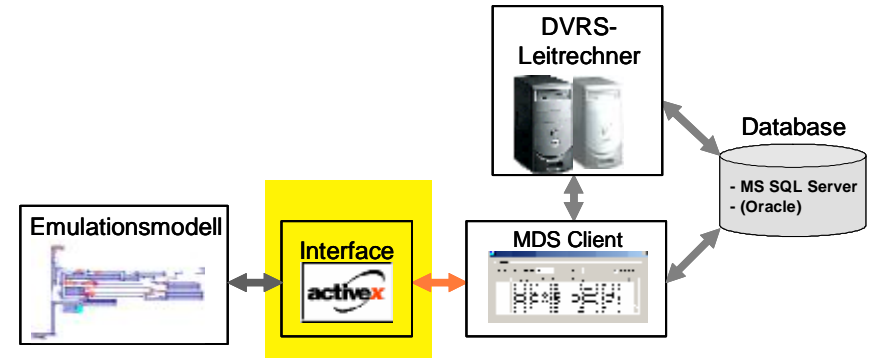
Emulation der gesamten Lackieranlage einschließlich des Painted Body Storage

- Einlasten der realen Produktionsaufträge (Karossensequenz) nach dem Rohbau in die Lackieranlage
- Schreiben der Karosendaten auf die Karosse
- Definition von Tracking- und Substitution-Points
- Prüfen der Vollständigkeit von ein- und ausgehenden Protokollen

Die System-Architektur



Aufgabe der Komponente:



Organisiert die Kommunikation zwischen Emulationsmodell und MDS Client

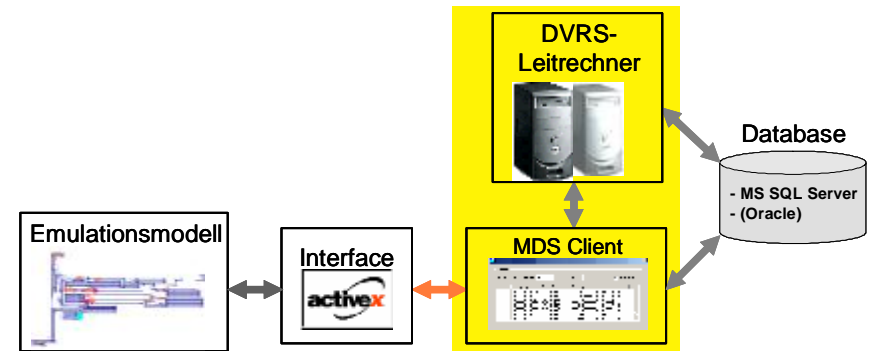
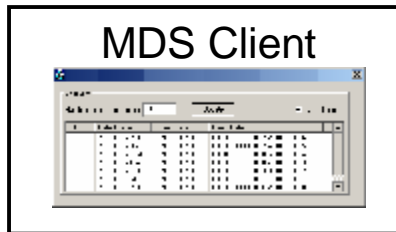
Es werden dabei 6 Telegramm- / Befehlstypen unterschieden:

1. **Tracking** à Der Karossendatensatz wird in die Datenbank geschrieben (Telegramm)
2. **Substitution** à Zwei Karossendatensätze werden ausgetauscht (Telegramm + Befehl)
3. **Update** à Der Karossendatensatz einer Karosse wird überschrieben (Telegramm + Befehl)
4. **PBS in** à Einlagern einer Karosse (Telegramm + Befehl)
5. **PBS out** à Auslagern der nächsten Karosse (Telegramm + Befehl)
6. **Status Stacker** à frei / Normalbetrieb / gestört (Telegramm + Befehl)

Die System-Architektur



Aufgabe der Komponenten:



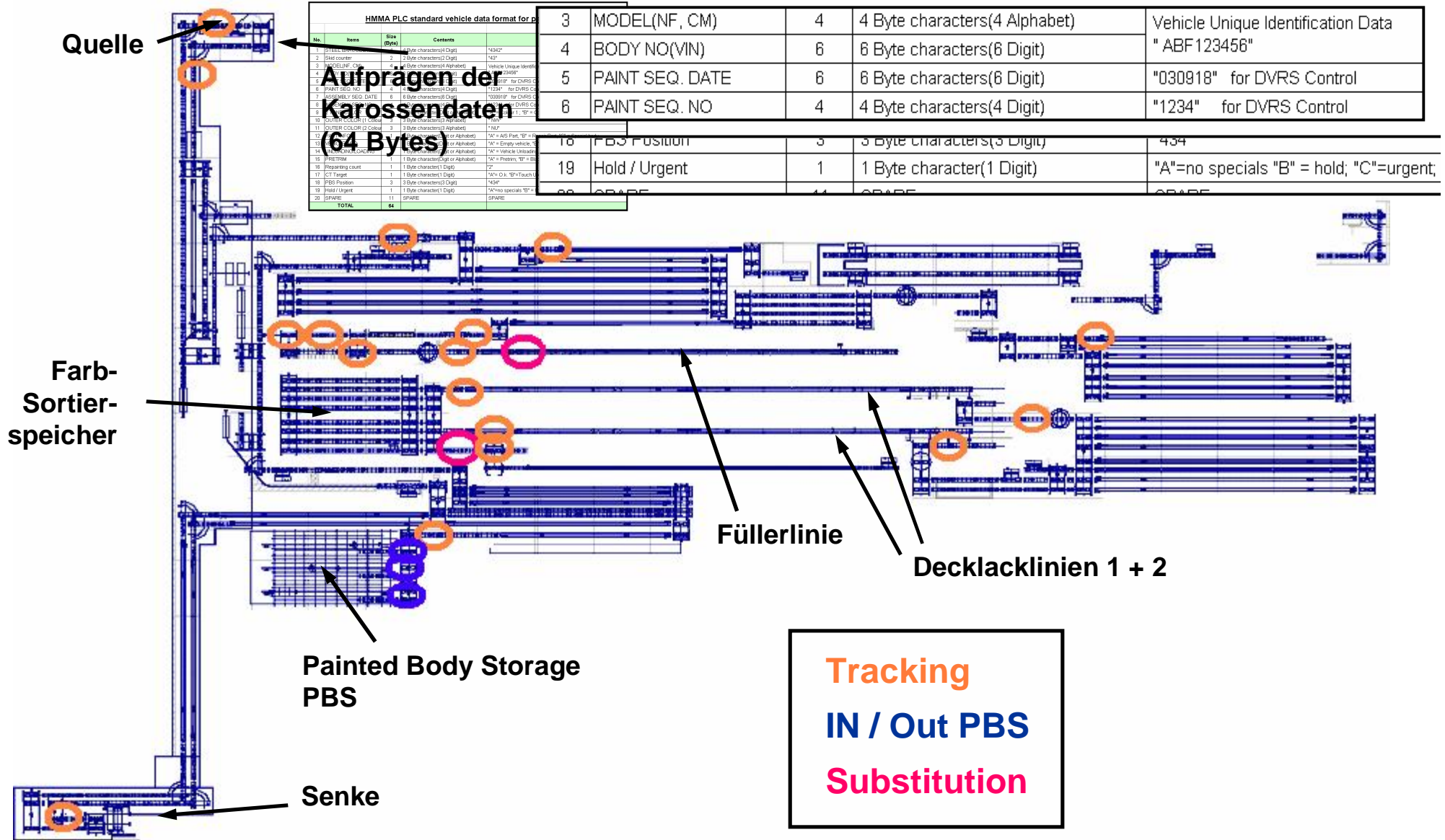
MDS Client

- Organisiert die Kommunikation mit dem DVRS-Leitrechner (EMOS SM)
- Formatiert / Konvertiert die Datensätze entsprechend dem EMOS SM Format
- Füllt die Datenbank mit den Tracking-Informationen
- Überträgt die EMOS SM Befehle an das ActiveX-Interface
- Switch zur SPS → Interface zur SPS

DVRS-Leitrechner (EMOS SM)

- Generiert die Substitution-, Up-date-, PBS-in- und PBS-out - Befehle

Die System-Architektur



Ausschnitt aus der Emulation Hyundai Alabama



mes_master_2_tk14.spp - eM-Plant 7.0 - [mod.HY.EB6 (50%)]

File View 3D Debugger Profiler Tools Help

Bearbeiten Navigieren Objekte Symbole Ansicht Extras ?

Snagit Video-Capture

Capture Statistics

- Aufgenommene Bilder: 0
- Entfallene Bilder: 0
- Dateigröße: 0
- Videolänge: 0 Sekunden
- Aufnahmelänge: 0 Sekunden

Capture Properties

- Bildgröße: 1024 x 768
- Bildrate: 5,0 Bilder/Sek
- Farben: True Color
- Komprimierung: Microsoft Video 1
- Audio-Aufnahme:

CppText

Events A

Max lines before autodelete: 10 Auto delete

| ID | Date & Time | Event Place | Event Data |
|----|----------------|-------------|-----------------------------|
| 1 | 20030801484835 | 313110118 | 0061DSAAAA0003450403020185 |
| 1 | 20030801484179 | 316117RB110 | 0004DSAAAA0009730403020004A |
| 1 | 20030801483355 | 319101RB714 | 0097DSBBBB0013870403020352 |
| 1 | 20030801483301 | 313107RB110 | 0110DSBBBB0007610403020234 |
| 1 | 20030801482829 | 317114RB212 | 0057DSBBBB0008140403020057A |

Events B

Max lines before autodelete: 10 Auto delete

01.08.2003 09:48:36.5065

Start | Kalender - Microsoft Ou... | C:\Hyundai\DVR5\MOD... | mes_master_2_tk14.sp... | CppText | DE | 09:44

Nutzen

Test und Inbetriebnahme der Steuersoftware

auf **herkömmliche** Weise

mit Hilfe der **Emulation**

Inbetriebnahme

Erst nach der Realisierung der Anlage im vollen Umfang möglich

- abhängig von der Baustellensituation
- Test von Auswertfunktionen erst im Produktionsbetrieb möglich

Bereits vor der Realisierung der Anlage möglich

- unabhängig von der Baustellensituation
- Auswertfunktionen können unabhängig von der Anlage getestet werden

Test von Steuerstrategien

- aufwändig und zeitintensiv
- viele Szenarios nicht testbar
- abhängig vom Anlagenbetreiber

- einfach und schnell
- alle Szenarios sind testbar
- unabhängig von der realen Anlage

Qualität

- wächst mit der Produktion.
à Nacharbeit erforderlich

- bereits vor dem Produktionsstart optimiert.
à Reduzierte Nacharbeit

Emulation lohnt, wenn...

- ... der Aufwand für die Erstellung des Emulationsmodells $<$ Einsparungen durch die verkürzte Inbetriebnahmezeit.
- ... Ein existierendes Simulationsmodell für die Emulation verwendet werden kann.
- ... die Emulation nach der Inbetriebnahme für den weiteren Produktionsbetrieb eingesetzt werden kann.

Kontakt:

Dr.-Ing. Robert Schulz

DÜRR Automotion GmbH
Systemtechnik Simulation
Otto-Dürr-Straße 8
70435 Stuttgart
Tel. +49 (0)711 136-2631
Fax +49 (0)711 136-1955
E-Mail robert.schulz@durr.com
www.durr.com