



Informationstechnik

Hochschule Esslingen
University of Applied Sciences

IT-Innovationen

Band 15

Juni 2015

Grußwort des Dekans

Liebe Leserinnen und Leser,

“Wir können nur über Innovation und Bildung punkten“, so Daimler-Chef Dieter Zetsche in einem Interview mit dem Handelsblatt am 28.04.2015. Beides sind wesentliche Kernelemente des Studiums der Informationstechnik an der Hochschule Esslingen.



Bildung meint die Vermittlung von Kompetenzen auf fachlichem Gebiet, das Erlernen von Grundlagen als Basis für ein lebenslanges Lernen, das Auseinandersetzen mit neuesten Technologien. Bildung meint aber auch die Vermittlung von Kompetenz im Persönlichen, Entwicklung der jungen Menschen zur beruflichen Reife, damit sie in den Unternehmen erfolgreich agieren können. Räumliche und personelle Nähe zum Fraunhofer Anwendungszentrum KEIM, dem ersten Zentrum für angewandte Forschung an einer Fachhochschule in Baden-Württemberg, ermöglicht unseren Studierenden die Auseinandersetzung mit neuesten Technologietrends. Die engen Kontakte der Professorinnen und Professoren in die umliegende Industrie öffnen unseren Studierenden regelmäßig den Zugang zu deren Entwicklungslabors mit vielfältigen innovativen Fragestellungen. Der Ihnen vorliegende Band der IT-Innovationen bezeugt die Vielfalt der Themenstellungen, mit denen unsere Absolventinnen und Absolventen betraut werden. Lassen Sie sich davon inspirieren.

Viel Freude beim Lesen wünscht Ihnen herzlichst Ihr

A handwritten signature in blue ink that reads "Nonnast". The signature is fluid and cursive.

Prof. Jürgen Nonnast
Dekan der Fakultät Informationstechnik

IMPRESSUM

ERSCHEINUNGSORT

73732 Esslingen am Neckar

HERAUSGEBER

Prof. Jürgen Nonnast
Dekan der Fakultät Informationstechnik
der Hochschule Esslingen - University of Applied Sciences

REDAKTIONSANSCHRIFT

Hochschule Esslingen - University of Applied Sciences
Fakultät Informationstechnik
Flandernstraße 101
73732 Esslingen am Neckar

Telefon +49(0)711.397-4211
Telefax +49(0)711.397-4214
E-Mail it@hs-esslingen.de
Website www.hs-esslingen.de/it

REDAKTION, LAYOUT UND DESIGN

Prof. Dr.-Ing. Reinhard Schmidt
Hochschule Esslingen - University of Applied Sciences
Fakultät Informationstechnik
Flandernstraße 101
73732 Esslingen am Neckar

SATZ, ANZEIGEN und VERLAG

Martin Gärtner, B. Eng.
Hochschule Esslingen - University of Applied Sciences
Fakultät Informationstechnik
Flandernstraße 101
73732 Esslingen am Neckar

ERSCHEINUNGSWEISE

Einmal pro Semester, jeweils Januar und Juni

DRUCK

Pixelgurus
Werbung – Werbetechnik – Digitaldruck.
Horbstraße 8
73760 Ostfildern

AUFLAGE

500 Exemplare

ISSN 1869-6457

Thomas Anders	Integration von Sensormodellen in das Simulationssystem für die Steuerung von E-Bikes	1
Erdinc Avci	Entwicklung und Implementierung von Schnittstellen zwischen Fahrerassistenz- und Infotainmentsystemen im autonom fahrenden Fahrzeug	3
Fabian Binder	Aufbau, Konfiguration und Bewertung einer digitalen High-Speed SENT-Schnittstelle zwischen Motorsteuergerät und Raildrucksensor eines Common-Rail-Systems als Ersatz für die bestehende analoge Schnittstelle	5
Timo Bühner	Entwurf und Implementierung eines auf der Arduino-Plattform basierenden Testgerätes in C++ für die Überprüfung der Montage eines Embedded Systems und zugehörigem Kabelsatz in einem Nutzfahrzeug sowie die Verifikation der empfangenen Signale der angeschlossenen Peripherie	7
Fatih Cirak	Konzeption und Implementierung einer mobilen Anwendung zur Fehlersuche eines eingebetteten Systems im Fahrzeug unter Anwendung der Vorgehensweise Lean UX	9
Paolo Di Donna	Erstellen einer Virtual Reality-Anwendung im Bereich der technischen Visualisierung mittels eines HMD am Beispiel der madness GmbH	11
Ismail Dogan	Konzeption und Implementierung eines Prototypen für einen adaptiven Empfehlungsdienst	13
Michael Dreher	Positionsbestimmung durch 2D Koppelnavigation	15
Alexander Ehni	Optimierte Datenübertragung für dynamisch anwachsende Datenblock-Ressourcen in REST Systemen	17
Tobias Fillbrandt	Entwurf, Implementierung und Laufzeitanalyse einer Java-API unter der Verwendung einer Basis-Software-Architektur auf einem Embedded-Linux-System zur Steuerung eines autonomen Rasenmähers	19
Özgen Gönül	Entwicklung von Methoden zur Objekt-Spur-Allokation für Fahrerassistenzsysteme in schweren Nutzfahrzeugen	21
Patrick Jakob	Secure-Boot-Konzept für Embedded-System-Rechnersysteme	23
Sebastian Klemm	Entwicklung eines Moduls für die Verarbeitung von analogen Eingangs- und Ausgangsdaten als Erweiterung des EasySlave Evaluation-Kits für Sercos III einschließlich Implementierung der Logik für den synchronen Betrieb des auf einem Xilinx-FPGA basierenden EasySlaves	25
Kevin Kögler Villafuerte	Integration von Smart Home Geräten in ein bestehendes Hausautomatisierungs-System über Internet-Protokolle	27
Stefan Lutz	Konzeption und prototypische Umsetzung einer Anwendung zur Unterstützung von Websitebesuchern auf Basis von browser-eigenen Technologien	29
Edouard Maihöfer	Konzept und Entwicklung einer mobilen App für die Verwaltung von heterogenen Elektro-Fuhrparkflotten	31
Lukas Martin	Vergleich moderner Authentisierungsmethoden und deren Einsatzmöglichkeiten	33

Steffen Mergheim	Verifikation von Mustererkennungsverfahren in der industriellen Bildverarbeitung am Beispiel einer Labelprüfung	35
Thomas Müller	Evaluierung eines Automotive Testsystems (HiL) mit Komponenten aus der Automatisierungstechnik am Beispiel eines Schiebedachs	37
Raphael Mußnug	Entwicklung eines REST-Webservice zur Ortdatensuche im Umfeld von Einsatzleitsystemen mit C#/ASP.NET-Web-API	39
Marcel Neumann	Konzeption und Implementierung einer Testumgebung für den Dauertest von Telematikeinheiten, mit Simulation realistischer Umgebungsbedingungen durch einen Zustandsautomaten, der gültige Signale für CAN- und serielle Schnittstellen generiert, sowie Darstellung des Testablaufs in einer grafischen Oberfläche.	41
Arno Orendi	Modellbasierte Entwicklung einer unterlagerten Kaskadenregelung für den Antrieb eines Pedelecs auf Basis eines bürstenlosen Gleichstrommotors mit Hilfe von MATLAB/Simulink einschließlich automatischer Generierung des Quellcodes für die Implementierung des Reglers auf einem Mikrocontroller der Familie dsPIC33	43
Antonio Pomarico	Modellierung der Leistungselektronik eines Kfz-Hybridantriebs	45
Janet Protzer	Analyse, Anwendung und Erweiterung der Methode Lean UX am Beispiel eines Quantified Self Monitor	47
Stefan Ringhoffer	Entwurf und Realisierung eines Fuzzingtools für kryptographische Protokolle	49
Jens Schmid	Analyse und Verbesserung der Usability eines Content-Management-Systems	51
Marc Schubert	Testautomatisierung von grafischen Oberflächen – Erhebung des Testbedarfs am Fraunhofer-Anwendungszentrum KEIM und prototypische Umsetzung	53
Daniel Secker	Grafische Visualisierung von Diagnosedaten	55
Tobias Thill	Entwicklung eines radarbasierten Notbremsassistenten für ein „Neighborhood Electric Vehicle“	57
Yannick Uhlmann	Analyse eines Testsystems für CAN- und FlexRay-Interfaces sowie Implementierung und Aufbau eines Nachfolgesystems einschließlich Migration aller Funktionen	59
Watthana Vilaysouk	Konzipierung und Realisierung des Moduls zur Verarbeitung von Einzelsignalen einer Nachricht für einen CAN-Protokollstack zur Kommunikation zwischen Steuergeräten in Fahrzeugen unter Berücksichtigung einer zeit- oder ereignisorientierten Übermittlung der Signale	61
Timo Wascheck	Umfeldmodellierung durch Belegungsgitter auf Basis von Ultraschallsensoren	63
Clemens Weißenberg	Entwurf und Implementierung eines Herzfrequenz Koinzidenzdetektor Testsystems	65
Thomas Wild	Extreme Programming	67

Integration von Sensormodellen in das Simulationssystem für die Steuerung von E-Bikes

Thomas Anders*, Werner Zimmermann, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Die Zukunft ist elektrisch und kommt auf zwei Rädern, meint man, wenn man die aktuelle Entwicklung der Verkaufszahlen von Pedelecs und E-Bikes verfolgt. Während im Jahr 2009 noch etwa 150 000 E-Bikes und Pedelecs in Deutschland über die Ladentheke gingen, waren es 2014 schon ca. 480 000 Stück [1]. Das E-Bike liegt voll im Trend und kann gerade im Stadt- oder Berufsverkehr seine vollen Stärken ausspielen, um wertvolle Minuten einsparen, aber auch ältere Menschen können Dank der elektrischen Motorunterstützung größere Strecken zurücklegen. Somit trifft das E-Bike genau den Nerv der Zeit, einer immer umweltbewussteren, schnell alternden Gesellschaft mit wachsendem Gesundheitsbewusst-

sein, in der "quality time" ein kostbares Gut ist. Pedelecs unterscheiden sich von herkömmlichen Fahrrädern durch einen zusätzlich verbauten Elektromotor, wobei die zusätzliche Leistung, die der Motor liefert, von der Tretleistung des Fahrers abhängt. Bosch wagte im Jahre 2011 den Einstieg in das hartumkämpfte E-Bike Geschäft und konnte sich in kürzester Zeit als europäischer Marktführer im Produktbereich der eBike Systeme behaupten. Hierbei wird der Motor (Mittelmotor), die Batterie und das Tacho von Bosch entwickelt und hergestellt. Diese Komponenten werden von über 50 Fahrradherstellern in ihren Pedelecs verbaut [2].



Abbildung 1: Bosch eBike Systemkomponenten

Das Fahrgefühl findet in Bosch eBike Systemen besondere Beachtung. Dabei soll sich die Fahrt mit einem eBike so natürlich wie möglich anfühlen, als ob die gesamte Leistung durch den Fahrer erzeugt werden würde. Dieses Ziel wird durch eine genaue Analyse des Fahrerwunsches erreicht. Mit am wichtigsten ist dafür die Kenntnis über das vom Fahrer vorgegebene

Drehmoment. Mit den Informationen über das Drehmoment kann der Motorregler den Fahrerwunsch erkennen und umsetzen, um damit ein angenehmes Fahrgefühl zu erzeugen. Offensichtlich wird die bedeutsame Rolle über eine gute Kenntnis des Drehmomentes auch im Falle eines Pedelecs mit Rücktrittbremse. Das System muss in der Lage sein, eine vom Fah-

*Diese Arbeit wurde durchgeführt bei der Firma Robert Bosch GmbH, Reutlingen

erer eingeleitete Bremsung via Rücktritt in Sekundenbruchteilen zu erkennen, da sonst der Motor gegen den Fahrer arbeitet.

In der angefertigten Bachelorarbeit wurde ein Sensorsystem untersucht und simulativ abgebildet, welches das vom Fahrer erzeugte Drehmoment misst. Ziel der Arbeit ist es, ein realistischeres, am Computer generiertes Eingangssignal des Drehmomentes für die Motorregelung zu entwerfen.



Abbildung 2: eBike Animation mit Bosch Komponenten

Das Drehmoment wird über die Kraft gemessen, die auf die Tretwelle wirkt. Dabei wird die vom Fahrer auf die Welle eingepreßte Kraft über das Getriebe übertragen und umgelenkt. Diese umgelenkte Kraft bringt einen doppelten Biegebalken zur Beugung und führt somit zu einer relativen Positionsänderung eines Hallsensors zu einem Magneten. Durch die geän-

derte Position von Magnet und Hallelement ändert sich die Hallspannung, welche in ein Drehmoment umgerechnet wird. Als Tooling wurde Matlab, Matlab Simulink, Matlab Simscape und Matlab SimMechanics verwendet. In einem ersten Schritt wurde ein Konzept zur simulativen Abbildung von Sensoren in Bosch eBike Systemen entworfen.

Im nächsten Schritt wurde ein Sensormodell aufgebaut, welches das Verhalten des Drehmomentsensors nach außen korrekt wiedergeben soll. Dabei wurden vor allem das elektrische Verhalten und die Temperaturabhängigkeit des Sensors betrachtet. Der Sensor wurde in ein abgespecktes Modell der Gesamtsimulation integriert und verifiziert. Das erste entworfene Modell ist nicht in der Lage, Ungenauigkeiten bei der Messung des Drehmomentes zu beschreiben, die aufgrund von Fertigungstoleranzen entstehen, sofern diese nicht schon im Gesamtmodell berücksichtigt wurden. Aus diesem Grund wurde ein zweites Modell entworfen, welches zum Ziel hat, die Diskrepanz zwischen gemessenen und durch das Modell dargestellten Drehmomentwerte zu minimieren. Hierzu sind Untersuchungen der gesamten Mechanik des Motors notwendig. Dazu wurde ein statischer Drehmoments-Prüfstand um einen Winkelsensor erweitert, um das Getriebe besser untersuchen zu können. Zusätzlich wurden das auf die Tretwelle wirkende Moment über einen Referenzsensor aufgezeichnet, sowie das Moment des im System verbauten Sensors. Mit den neu gewonnen Daten konnte Spiel im Getriebe identifiziert werden und es war möglich, dieses Spiel in der Simulation nachzuahmen und somit eine weitere Annäherung an eine Drehmoments Messung eines echten Systems zu erreichen.

[1] <http://de.statista.com/statistik/daten/studie/152721/umfrage/absatz-von-e-bikes-in-deutschland/>
Abruf: 01.05.2015

[2] http://www.bosch-ebike.de/de/company/about_us/about_us.html
Abruf: 01.05.2015

Bildquellen:

- Abbildung 1: http://www.bosch-ebike.de/de/produkte_neu/performance_line/performance_line_1.php
Abruf: 01.05.2015
- Abbildung 2: Bosch eBike System – Ride your eWorld, Seite 5

Entwicklung und Implementierung von Schnittstellen zwischen Fahrerassistenz- und Infotainmentsystemen im autonom fahrenden Fahrzeug

Erdinc Avci*, Reiner Marchthaler, Werner Zimmermann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Die Geschichte der Infotainmentsysteme begann schon vor über 90 Jahren in den USA, mit dem ersten Einbau eines Radios im Jahre 1922. Seither ist durch die enorme Entwicklung der Unterhaltungselektronik die Erwartung der Benutzer an das Infotainmentsystem im Fahrzeug rasant gestiegen, sodass die Internetverbindung im Fahrzeug und die Anbindung von Smartphones längst Stand der Technik sind. Doch neben dem Wunsch der Benutzer nach innovativer Technik, sind auch die Anforderungen der Automobilindustrie, wie Qualität, Sicherheit und lange Lebensdauer Gründe für die Komplexität.



Abbildung 1: Infotainmentsysteme:
(Radio) 1925 und 2014

Zu den Infotainmentsystemen die zur Unterhaltung dienen, kommen die Fahrerassistenzsysteme, welche den Fahrer bei seiner primären Fahraufgabe unterstützen hinzu. Diese Fahrerassistenzsysteme sind auch keine Neuheiten für die Fahrzeugindustrie. So wurde z.B. der Geschwindigkeitsregler bereits 1945 erfunden und nach 13 Jahren erstmals als Cruise Control bei Chrysler eingebaut. Die Fahrerassistenzsysteme informieren und warnen den Fahrer und erhöhen seinen Komfort und die Sicherheit, indem sie ihn aktiv bei seiner Fahrzeugführung und Fahrzeugstabilisierung unterstützen. Über die letzten 70 Jahre wurden immer mehr passive und aktive Fahrerassistenzsysteme entwickelt, was somit die Komplexität auch in diesem Bereich erhöht.



Abbildung 2: Fahrerassistenzsysteme:
Die kleinen Helfer

Im Vergleich zu den Infotainment- und Fahrerassistenzsystemen ist die Konnektivität im und um das Fahrzeug noch weitgehend Neuland für die Fahrzeugindustrie. Obwohl die Car-to-X (C2X) Kommunikation seit Ende der 90er Jahre in diversen nationalen und internationalen Forschungs- und Entwicklungsprojekten untersucht wurde, gewinnt die Vernetzung von Fahrzeugen untereinander, sowie die Vernetzung mit der Infrastruktur erst in den letzten Jahren an Bedeutung. Mit Dieser stieg auch die Anzahl der Konnektivitätspartner, sodass das X nun nicht mehr nur für Car und Infrastructure, welche für die Verbesserung der Verkehrssicherheit und der Verkehrseffizienz dienen, sondern unter anderem auch für Home steht, was den persönlichen Komfort steigert.

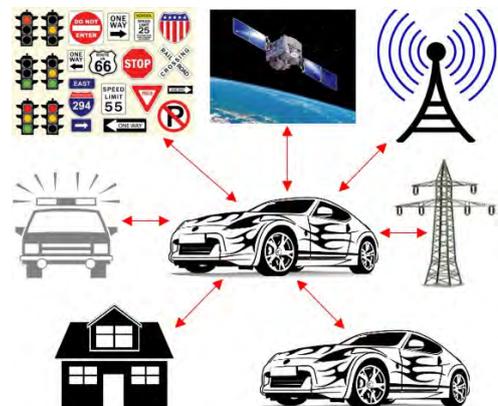


Abbildung 3: Konnektivitätspartner: C2X

*Diese Arbeit wurde durchgeführt bei der Firma Robert Bosch GmbH, Leonberg

Es wurde erkennbar, dass mit zunehmender Vielfalt und Komplexität der Infotainment-, Fahrerassistenz- und Konnektivitäts-Funktionen innerhalb eines Fahrzeugs die kognitive Belastung des Fahrers steigt. Um diese Belastung zu reduzieren wurde ein hoch automatisiertes Auto mit einer Vielzahl von neuen intelligenten Funktionen eingesetzt. Als Teil dieser Architektur wurde ein sogenannter zentralisierter HMI-Manager eingesetzt, welches Informationen anderer Steuergeräte im Fahrzeug auswertet und Fahrer-, Fahrzeugzustand und Umgebung modelliert. Neben der Reduzierung der Fahrerbelastung, die z.B. durch die Übernahme des Fahrzeugs auf Autobahnen erfolgt, war ein einzigartiges Fahrerlebnis (User Expe-

rience) ein weiterer wichtiger Aspekt dieser Arbeit. Zu Beginn dieser Arbeit lief dieser zentralisierte HMI-Manager in einem Demonstrator auf Target-Hardware, bei dem die Inputs simuliert und die Outputs zum größten Teil auf Serien-Displays liefen.

Im Rahmen dieser Arbeit wurde in Zusammenarbeit mit Experten der unterschiedlichen Fahrzeugdomänen neben der Entwicklung und Implementierung von Schnittstellen für die Kommunikation zwischen dem Robotic Operating System (ROS) und dem zentralisierten HMI-Manager, auch die Implementierung eines HTML5 Clients für die Kommunikation zwischen dem HMI-Manager und dem HTML5 Framework realisiert.

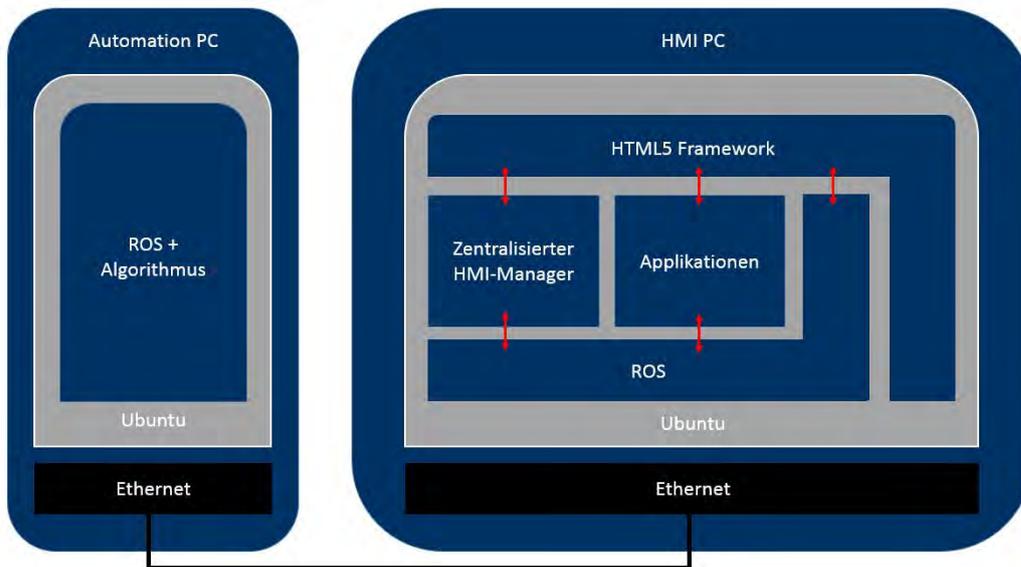


Abbildung 4: Gesamtkonzept

[1] Erdinc Avci, Bachelorarbeit, Robert Bosch GmbH CM-AI/ECB-Lr

Bildquellen:

- Abbildung 1: www.radiomuseum.org und www.teslamotors.com
- Abbildung 2: www.autobild.de
- Abbildung 3,4: Eigene Grafik

Aufbau, Konfiguration und Bewertung einer digitalen High-Speed SENT-Schnittstelle zwischen Motorsteuergerät und Raildrucksensor eines Common-Rail-Systems als Ersatz für die bestehende analoge Schnittstelle

Fabian Binder*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

In einem Common-Rail-System wird der Raildruck bisher analog übertragen und im Motorsteuergerät mit einem Analog-Digital-Wandler abgetastet. Die Aufgabe der vorliegenden Bachelorarbeit besteht darin, aus der bestehenden analogen Schnittstelle zwischen Motorsteuergerät und Raildrucksensor eine digitale Schnittstelle zu entwickeln. Dafür soll das serielle Übertragungsprotokoll SENT verwendet werden. Dieses Protokoll wurde im Rahmen einer Erweiterung (H.3) des bestehenden Standards SAE J2716 angepasst und soll für das ausgewählte Motorsteuergerät konfiguriert werden.

Das Rail als Hochdruckspeicher ist eine wichtige Komponente des Diesel-Einspritzsystems Common-Rail. Bei einem Common-Rail-System wird der Kraftstoff mit bis zu 270 MPa in den Brennraum eingespritzt. Dieser hohe Druck wird von einer Hochdruckpumpe erzeugt und im Rail zwischengespeichert. Der Raildrucksensor (RDS) übermittelt die aktuellen Hochdruckwerte an das Motorsteuergerät. In Abhängigkeit vom Druckwert wird dann der Einspritzpunkt und die Einspritzdauer festgelegt.

SENT, *Single Edge Nibble Transmission*, ist ein Übertragungsprotokoll, das hauptsächlich in

der Automobilbranche eingesetzt wird und seit 2007 standardisiert ist. Das Protokoll wird für Punkt-zu-Punkt Verbindungen verwendet, z.B. für die Sensor-Aktuator-Kommunikation. Eine unidirektionale Verbindung ist besonders gut zur Übertragung der Messwerte eines intelligenten Sensors zu einem Steuergerät geeignet [1].

Es wird eine direkte Verbindung von einem Sensor oder Aktuator zu einem Steuergerät aufgebaut. Diese Verbindung benötigt eine zumindest dreiadrige Leitung, in der eine Signalleitung, die Versorgungsspannung und die Masse enthalten sein müssen.

Eine SENT-Nachricht beginnt immer mit einem Synchronisationsimpuls, der eine feste Dauer von 56 Ticks hat. Die Dauer des Übertragungstakts beträgt normalerweise zwischen $3 \mu\text{s}$ und $90 \mu\text{s}$. Dem Sendesignal wird zudem eine Toleranz von $\pm 20\%$ zugestanden. Dadurch ist es möglich, den Takt auf bis zu $2,4 \mu\text{s}$ zu verkürzen. Da der Raildrucksensor eine Takttoleranz von $\pm 10\%$ besitzt, wurde der Übertragungstakt auf $2,67 \mu\text{s}$ festgelegt.

Nach der Synchronisation folgen 4 Bit, auch ein Nibble genannt, für die Status-Information und die serielle Kommunikation. Ein Daten-nibble ist so aufgebaut, dass er mindestens

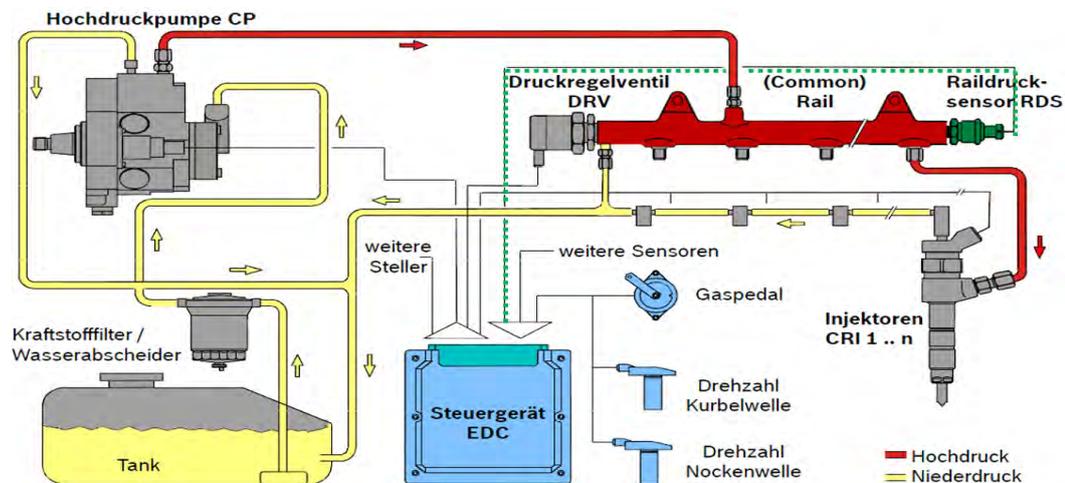


Abbildung 1: Common Rail System

*Diese Arbeit wurde durchgeführt bei der Firma Robert Bosch GmbH, Stuttgart-Feuerbach

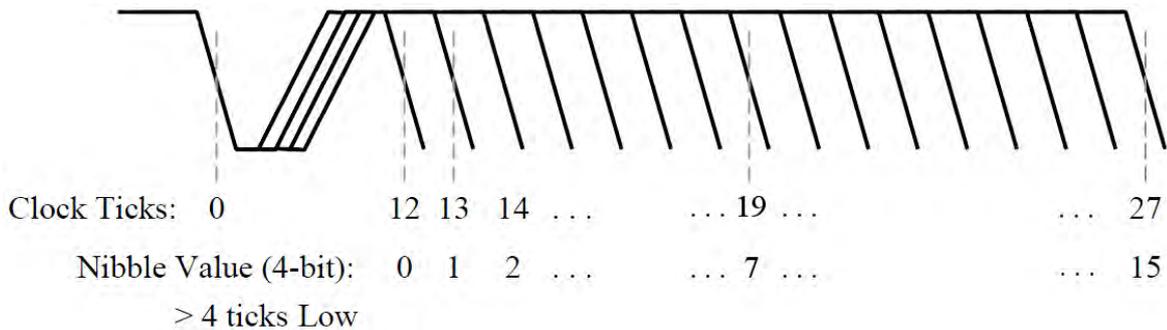


Abbildung 2: Dauer eines Datennibbles

12 und höchstens 27 Ticks dauert. Die Dauer ist von dem zu übertragenden Wert abhängig, weshalb es also keine feste Nachrichtenlänge gibt. Insgesamt können bis zu 6 Datennibble in einer Nachricht gesendet werden. Das Nachrichtenpaket wird mit einem 4 Bit breiten *Cyclic Redundancy Check* (CRC) gesichert. Zusätzlich kann zum Schluss der Nachricht ein Pause-Impuls gesendet werden, der die Abgrenzung zu einer neuen Nachricht verbessert [2].

Da die Datenrate des Standard-SENT-Protokolls für die Anwendung zur Übertragung des Raildrucks zu gering ist, wurde die Konfiguration auf Basis der High-Speed Variante von SENT durchgeführt. Bei High-Speed SENT wurde der mögliche Nutzdatenumfang auf 12 Bits gekürzt. Das entspricht der Hälfte der möglichen Nutzdaten von SENT. Zusätzlich

wurde die Aufteilung der Datenpakete verändert. Die insgesamt vier Datenpakete wurden jeweils um ein Bit auf 3 Bits reduziert. Die Dauer eines Datenpakets ist somit auf maximal 19 Ticks begrenzt. Durch diese Maßnahmen wird insgesamt die maximale Framedauer auf ca. 500 μ s beschränkt [2].

Für die Tests im Labor und im Fahrzeug wird ein spezieller Raildrucksensor mit zwei Sensorelementen benutzt. Damit ist es möglich, zeitgleich den Messwert sowohl analog als auch digital zu verarbeiten und auszuwerten. Nach diesen Messungen kann die Bitfehlerrate und die Latenzzeit der verschiedenen Übertragungen verglichen werden.

Die Ergebnisse der vorliegenden Arbeit dienen als Grundlage für weitere Studien im Bereich der Digitalisierung von Schnittstellen im Motorsteuergerät.

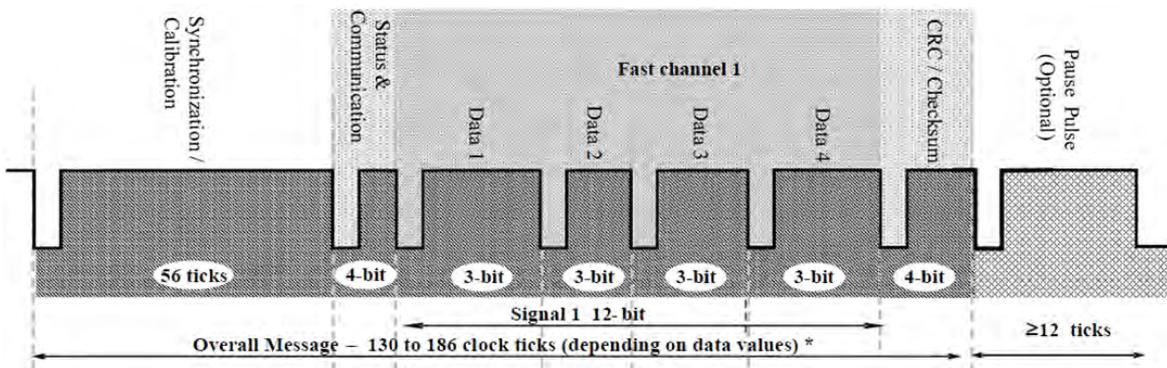


Abbildung 3: Aufbau einer High-Speed SENT-Message

[1] Zimmermann, W. / Schmidgall, R. ; Bussysteme in der Fahrzeugtechnik – Protokolle, Standards und Softwarearchitektur, Springer Vieweg, 2014, 5. Auflage

[2] SAE International, J2716 – Proposed Draft, 2010-03

Bildquellen:

- Abbildung 1: Robert Bosch GmbH
- Abbildung 2: SAE J2716, S.11
- Abbildung 3: SAE J2716, S.124

Entwurf und Implementierung eines auf der Arduino-Plattform basierenden Testgerätes in C++ für die Überprüfung der Montage eines Embedded Systems und zugehörigem Kabelsatz in einem Nutzfahrzeug sowie die Verifikation der empfangenen Signale der angeschlossenen Peripherie

Timo Bühler*, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Das herstellerunabhängige Telematiksystem der Firma Trendfire Technologies bietet ein umfangreiches Spektrum an Möglichkeiten zur Auswertung eines Fuhrparks mittels Flottenmanagement. Hierzu zählen zum Beispiel eine Fahrerbewertung, Tourenplanung, Navigation und Fahrzeugortung [1]. Kern der Lösung ist eine Telematikeinheit (siehe Abb. 1), welche Signale des Nutzfahrzeugs von verschiedenen Schnittstellen empfängt. So werden über einen CAN-Bus aktuelle Fahrzeugdaten empfangen. Diese Daten sind innerhalb der Flotten-Management-Schnittstelle definiert.



Abbildung 1: Trendfire Telematikeinheit

Das FMS-Gateway [2] ist eine standardisierte Schnittstelle, über welche Daten des Nutzfahrzeugs ausgelesen werden können. Zu diesen Daten zählen u.a. Fahrzeuggeschwindigkeit, Kilometerstand, Gewicht oder Kraftstoffgesamtverbrauch. Diese Schnittstelle wurde 2002 von führenden Herstellern von Nutzfahrzeugen wie MAN, Volvo, Scania und Daimler definiert, um Drittherstellern Zugang zu Fahrzeugdaten zu gewährleisten. Außerdem wird über den CAN-Anschluss des Tachographen der gesetzlich vorgeschriebene Download der Tachograph-Daten veranlasst. Diese Tachograph-Daten enthalten alle Fahraktivitäten sowie Verstöße des Fahrers. Über eine serielle Leitung werden zum einen am Tachographen der aktuelle Fahrerstatus abgefragt, zum anderen über ein Steuergerät im Auflieger Daten der Kühleinheit übermittelt. Abhängig davon, ob die Zündung des Fahrzeuges eingeschaltet ist oder nicht, führt die Telematikeinheit unterschiedliche Aufgaben aus.

Deshalb wird ebenfalls der Spannungspegel der Zündung abgefragt. Diese Signale werden mittels Adapterkabel im Fahrzeug abgegriffen (siehe Abb. 3). Der Einbau der Telematikeinheit wird von Mitarbeitern einer Werkstatt oder einem Monteur von Trendfire durchgeführt. Hierbei müssen je nach Fahrzeughersteller diverse Vorarbeiten durchgeführt werden, bevor die Telematikeinheit angeschlossen werden kann. Für den Einbau werden zwei Kabelsätze verwendet (siehe Abb. 3). Einen Trendfire Adapterkabelsatz, welcher an die Telematikeinheit angeschlossen wird und einheitlich für jedes Fahrzeug ist. Außerdem ein fuhrzeugspezifischer Adapterkabelsatz, da die Signale an unterschiedlichen Stellen im Fahrzeug abgegriffen werden müssen. Nach dem Einbau gibt es für den Techniker im Moment keine aussagekräftige Möglichkeit, welche eine schnelle Überprüfung der Verkabelung bietet. Erst nach dem Anschluss und Inbetriebnahme, sowie kompletten Bootvorgang der Telematikeinheit kann eine Aussage darüber getroffen werden, ob alles fehlerfrei funktioniert. Die Rückmeldung für den Techniker geschieht hierbei durch 8 LED's an der Telematikeinheit (siehe Abb. 2). Bis der Bootvorgang beendet ist und die Rückmeldung der Signale an die LED's stattgefunden hat, können je nach Fahrzeug mehrere Minuten vergehen. Zudem ist die Rückmeldung über die LED's nicht sehr aussagekräftig, da keine detaillierte Fehlerbeschreibung dargestellt werden kann. Wird ein Fehler erkannt und eine Änderung an der Verkabelung oder Konfiguration vorgenommen, kann die erneute Überprüfung wiederum mehrere Minuten dauern. Dieser Vorgang ist sehr zeitaufwendig für Techniker wie Kunde, weshalb hierfür eine bessere Lösung gesucht wird. Probleme für den Betrieb der Telematikeinheit können zum Beispiel auftreten, wenn der Spannungspegel der Zündung noch eine Restspannung hält, wodurch unerwünschte Funktionen auf der Telematikeinheit

*Diese Arbeit wurde durchgeführt bei der Firma Trendfire Technologies, Böblingen

ausgeführt werden. Über die Telematikeinheit kann außerdem der gesetzlich vorgeschriebene Download der Tachograph-Daten veranlasst werden. Dieser wird automatisch alle 60 Tage gestartet, muss jedoch vorher extra am Tachograph freigeschaltet sein, was nicht ohne weiteres von außen überprüft werden kann.



Abbildung 2: Frontseite Telematikeinheit LED's mit Beschreibung

Damit eine bessere und schnellere Überprüfung der richtigen Funktionalität gewährleistet werden kann, soll in dieser Arbeit ein externes Testgerät entwickelt werden, welches an die installierte Verkabelung angeschlossen wird (siehe Abb. 3). Dieser Tester soll dem Techniker dann Rückmeldung darüber geben, welche Daten richtig ausgewertet wurde und ob ggf. Fehler bei der Überprüfung aufgetreten sind. Als Basis für das Testgerät dient ein Arduino Due Entwicklungsboard. Ein Arduino ist ein OpenSource Entwicklungsboard, welches unmittelbar nach dem Einschalten betriebsbereit ist und somit einen schnelleren Auswertungsbeginn der Daten gewährleisten kann. Die Arduino Plattform bietet die Möglichkeit, durch Erweiterungsplatinen (=Shields) die Funktionalität des Arduino zu ergänzen. Für die Aus-

wertung der CAN-Daten wird ein CAN-Shield verwendet. Ein CAN-Controller ist bereits im Microcontroller des Arduino Due vorhanden, für die Kommunikation über den CAN-Bus wird außerdem ein CAN-Transceiver benötigt. Dieser befindet sich auf dem CAN-Shield. Zur Verifikation der korrekten Verkabelung und Konfiguration werden Daten an der angeschlossenen Fahrzeugperipherie abgefragt, welche eine schnelle Schlussfolgerung auf mögliche Fehler ermöglichen soll. Der Tester kann über den seriellen Anschluss des Tachographen das Datum, die Uhrzeit und das Fahrzeugkennzeichen abfragen. Außerdem wird über den CAN-Bus des Tachographen die Anfrage für den Tachodownload gestartet, wodurch überprüft wird, ob dieser freigeschaltet ist oder nicht. Über das FMS-Gateway werden das Fahrzeuggewicht, Tankfüllstand, gesamter Spritverbrauch, Kilometerstand, Motorbetriebszeit, FMS-Version, sowie Fahrzeugidentifikationsnummer abgefragt. Aus dem Steuergerät des Aufliegers werden Daten über einen möglichen Kühlaufleger eingelesen. Der Spannungspegel der Zündung soll bei eingeschalteter Zündung bei 12V oder 24V liegen. Dieser Wert wird über einen AD-Wandler abgefragt. Über ein LCD-Display werden während der Überprüfung die erkannten Daten und auftretende Fehler angezeigt. Zusätzlich kann bei Bedarf über einen kleinen mobilen Drucker ein Testbericht ausgedruckt werden, der für die Dokumentation des Testergebnisses dienen soll. Mithilfe dieses Testgerätes soll somit erreicht werden, dass die Überprüfung der Verkabelung und Konfiguration in Zukunft schneller und detaillierter durchgeführt werden kann.

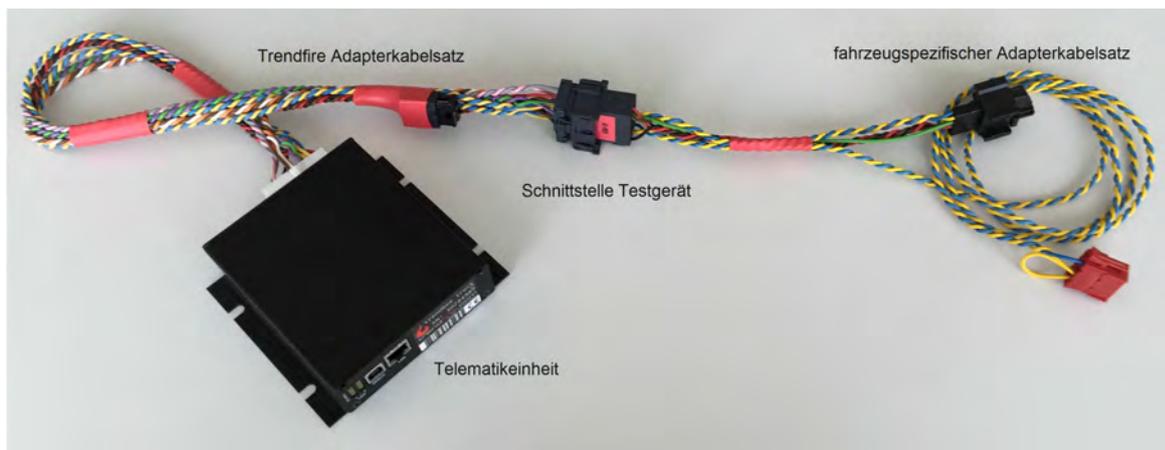


Abbildung 3: Verkabelung Telematikeinheit

[1] Homepage trendfire, www.trendfire.com

[2] FMS-Standard, www.fms-standard.com

Bildquellen:

- Abbildung 1-3: Trendfire

Konzeption und Implementierung einer mobilen Anwendung zur Fehlersuche eines eingebetteten Systems im Fahrzeug unter Anwendung der Vorgehensweise Lean UX

Fatih Cirak*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Die Softwareentwicklung spielt in der Automobilindustrie der heutigen Zeit eine immer größer werdende Rolle. Um die Konkurrenz in der Kategorie Innovation schlagen zu können, müssen immer neuere und komplexere Funktionen implementiert werden. Laut einer Untersuchung der Wiesbadener Technologie- und Innovationsberatungsgesellschaft Invenity[1] liegen schon heute 90 Prozent der autonomen Innovationen in den Bereichen Elektronik und Software. In der Entwicklungsphase der Software gehören Softwarefehler zur Tagesordnung und können lebensbedrohlich für den Nutzer werden. Damit fehlerhafte Software möglichst schnell fehlerfrei werden kann, müssen entsprechende Stellen in der Software gefunden und behoben werden. Um Fehler sichtbar zu machen beziehungsweise zu lokalisieren werden spezifische Tests durchgeführt um die Qualität der Software zu steigern.

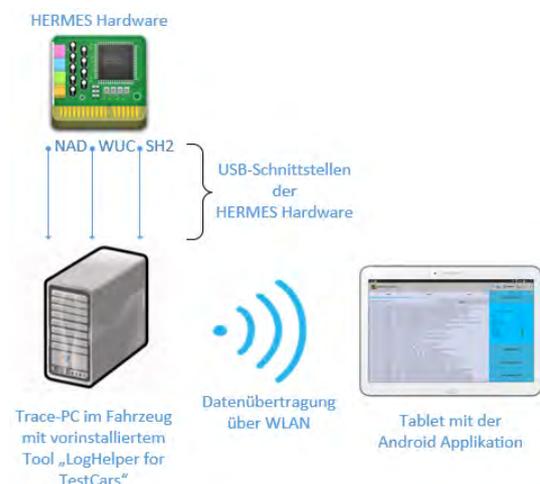


Abbildung 1: Aufbau der Testumgebung

Die Europäische Kommission hat am 8. September 2011 beschlossen die Einführung des standardisierten automatischen Auto-Notrufsystems eCall (Abk. Emergency Call, eng. für Notruf), welches ab Oktober 2015 verpflichtend in allen in Europa hergestellten und vertriebenen Neuwagen und kleineren Last-

fahrzeugen zu integrieren ist, zu unterstützen. Ziel ist es, die Anzahl der derzeit circa 28.000 jährlichen Verkehrstoten in ganz Europa signifikant zu senken[3].

Das Daimler Projekt „HERMES“ (Hardware for Enhanced Remote-, Mobility- & Emergency Services) soll die Voraussetzungen der Europäischen Kommission erfüllen. Außerdem bietet die Hardware Möglichkeiten zusätzliche Features wie zum Beispiel „Over the Air Diagnose“ zu integrieren. Diese werden in Zukunft eine kabellose Diagnose ermöglichen.

Diese Bachelorarbeit befasst sich mit dem Projekt HERMES und deren möglichen Softwarefehlern. Diese entstehen während des Entwicklungsprozesses und sind zu finden und zu analysieren. Hierzu wird eine Android-Applikation entwickelt. Diese soll den Fahrern während der Testfahrt dazu dienen, Fehler zu erkennen und die Möglichkeit bereitstellen, entsprechend reagieren zu können.

Da die Applikation im Fahrzeug genutzt werden soll, muss das Fahrzeug mit dem entsprechenden Equipment ausgerüstet sein. Hier kommt ein PC zum Einsatz der als Zugriffspunkt für die Applikation dienen soll. An diesem Trace-PC werden die seriellen Schnittstellen der HERMES Hardware angeschlossen. Das Tool „LogHelper for TestCars“ wurde entwickelt, um die Informationen der Schnittstellen über WLAN an das Tablet zu schicken. Die Informationen werden in der Applikation entsprechend dargestellt, damit die Fehlersuche erleichtert wird. Der komplette Aufbau im Fahrzeug ist in Abbildung 1 dargestellt.

Die Android Applikation bietet viele Funktionen, um eine Fehleranalyse schnell und effizient durchzuführen. Die Funktion „Filter“ ist eine der wichtigsten. Dies erlaubt es dem Nutzer die Ausgaben zu filtern. Zusätzlich kann der Nutzer mit der Funktion „Suche“ ein Wort definieren, das hervorgehoben wird. Eine weitere Funktionalität ist die Anzeige der Koppelnavigation (engl. Dead Reckoning) GPS-Daten. Diese werden auf einer integrierten Karte aufgezeichnet. Als Referenz wird die GPS Position

*Diese Arbeit wurde durchgeführt bei der Firma Harman Becker Automotive Systems GmbH, Fildderstadt

der Head Unit(HU) mit aufgezeichnet.

Aufgrund von suboptimalen Vorgehensweisen und nicht durchdachten Bedienkonzepten, kann die Bedienbarkeit unnötig erschwert werden. Die Testfahrer sollten jedoch in der Lage sein Software effektiv und effizient zu nutzen, damit sie durch ihre Anwendungssoftware bei ihrer täglichen Arbeit nicht behindert werden[4]. Aus diesem Grund wird

die Vorgehensweise Lean UX eingesetzt, welche heute als eine der effektivsten Entwicklungsmethoden für digitale Produkte zählt.

Lean UX basiert auf der Mischung verschiedener Denkansätze, um Prinzipien und Methoden zur Verbesserung der Usability und User Experience in eine agile Entwicklung zu integrieren.

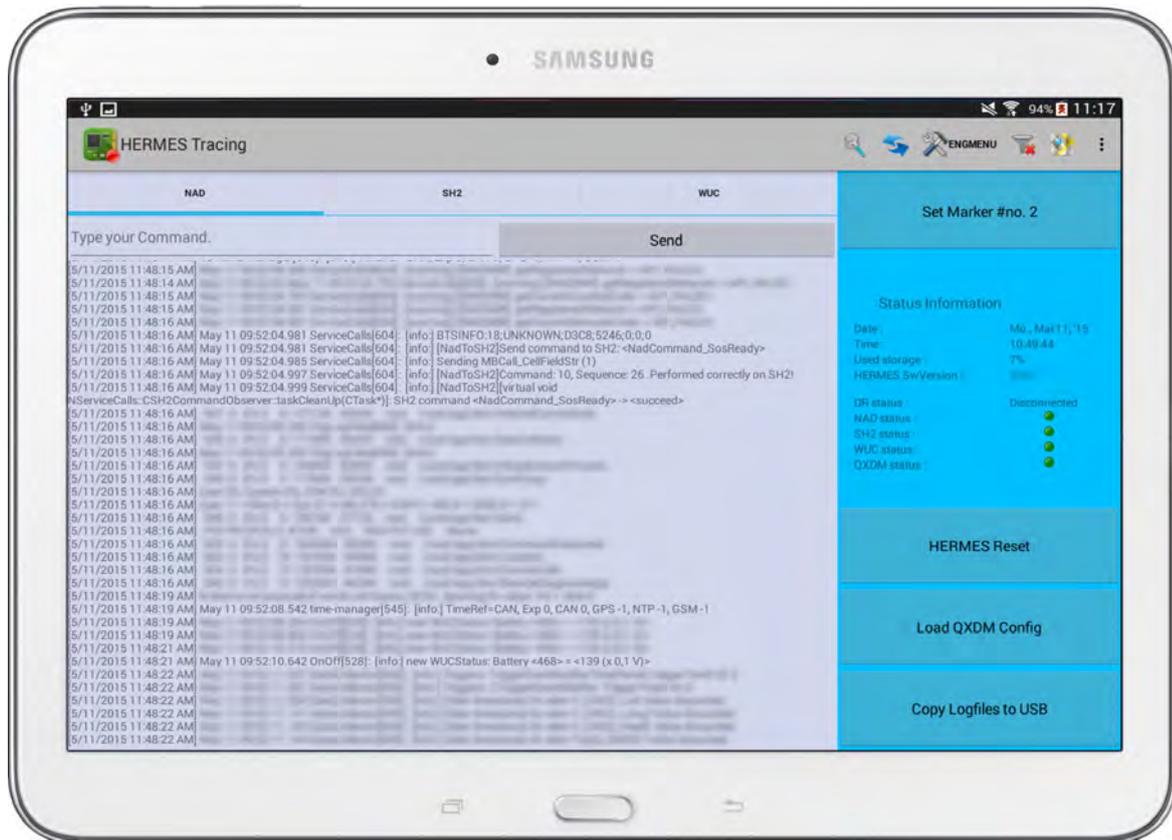


Abbildung 2: Applikation zur Anzeige von Hermes Daten

- [1] <http://www.invensity.com>
- [2] <http://vda-qmc.de>
- [3] <http://www.europarl.europa.eu>
- [4] Jacob Nielsen, Usability Engineering

Bildquellen:

- Abbildung 1,2: Fatih Cirak

Erstellen einer Virtual Reality-Anwendung im Bereich der technischen Visualisierung mittels eines HMD am Beispiel der madness GmbH

Paolo Di Donna*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Der industrielle Einsatz von Virtueller Realität ist in den vergangenen Jahren stark gestiegen. Aufgrund verfügbarer leistungsfähiger Hard- und Software sowie insbesondere hochleistungsfähiger Grafikkarten, gibt es kaum mehr einen industriellen Bereich, in dem Virtuelle Realität nicht eingesetzt wird. Die Anfänge der Virtuellen Realität finden sich in der Militärtechnik. Im industriellen Bereich findet man Virtuelle Realität vor allem bei Automobilherstellern, in der Städteplanung und Architektur, in der Medizintechnik oder für Fahr- und Flugsimulationen. Im industriellen Umfeld wird Virtuelle Realität insbesondere für schnelles Prototyping, Design-Studien, Ein- und Ausbaustudien von Komponenten und Ergonomiestudien eingesetzt. Einen Überblick über Virtuelle Realität findet sich beispielsweise in [1] und [2]. Abbildung 1 zeigt eine Virtual Reality Anwendung aus der Automobilindustrie.



Abbildung 1: Virtual Reality in der Automobilindustrie

Die Spieleindustrie hat Virtuelle Realität ebenfalls für sich entdeckt. Während bei industriellen Anwendung eine großflächige stereoskopische Visualisierung mit sogenannten Powerwalls zum Einsatz kommt, gibt es für den privaten Anwender relativ kostengünstige Head-mounted-Displays (HMD). Das HMD Oculus Rift der Firma Oculus VR fand im privaten Gaming-Bereich innerhalb kürzester Zeit eine sehr breite Anwendung. Die Oculus Rift ermöglicht ein echtzeitfähiges 360° Head Tracking des Anwender. Abbildung 2 zeigt das

HMD Oculus Rift. Weitere HMDs für den Entertainmentbereich liefern beispielsweise die Firmen Sony, Zeiss oder Cybermind.



Abbildung 2: HMD Oculus Rift

Ein Bereich, in dem Virtual Realität bisher noch keinen besonderen Einsatz gefunden hat, ist der Bereich der technischen Produktvisualisierung sowie Marketing und Werbung. Genau an diesem Punkt knüpft diese Abschlussarbeit an. Bisher findet man für vorhandene technische Produktvisualisierungen lediglich Fotoserien mit verschiedenen Ansichten zum Produkt und einer Lupenfunktion zur Vergrößerung. Beispiele hierzu sind elektronische Versandhäuser, wie beispielsweise Amazon oder Conrad Electronic Versand. Das Ziel dieser Arbeit ist es, die Möglichkeiten und das Potenzial von Virtual Reality in diesem Bereich mithilfe der Erstellung einer Beispielanwendung aufzuzeigen und anhand dieser Beispielanwendung prototypisch zu lösen. Abbildung 3 zeigt ein Beispiel für eine technische Produktvisualisierung.

Der technische Fortschritt eröffnet auch hier neue Möglichkeiten. Jedoch verfügen technische Visualisierungen, im Vergleich zu Videospielen, oft nicht über die notwendigen spieltypischen Elemente und Interaktionsmöglichkeiten, die einen Einsatz von Virtual Reality deutlich mehr intensivieren würden. Spieltypische Elemente können jedoch durch Gamification-Ansätze hinzugefügt werden. Unter Gamification oder Gamifizierung versteht man die Anwendung von spieltypischen

*Diese Arbeit wurde durchgeführt bei der Firma madness GmbH, Göppingen

Elementen auf Anwendungen mit spielefremdem Kontext. Spieletypische Elemente sind beispielsweise Erfahrungspunkte, Highscores, Fortschrittsbalken, Auszeichnungen oder virtuelle Güter. Die Integration von spieletypischen Elementen hat das Ziel, eine Motivationssteigerung beim Anwender zu erreichen. Dabei werden weniger herausfordernde, als monoton empfundene oder auch zu komplexe Aufgaben leichter gemeistert. Mit Hilfe der Gamifizierung lässt sich eine deutliche Verbesserung der Benutzermotivation, beim Lernerfolg oder auch einer Kundenbindung feststellen [3].

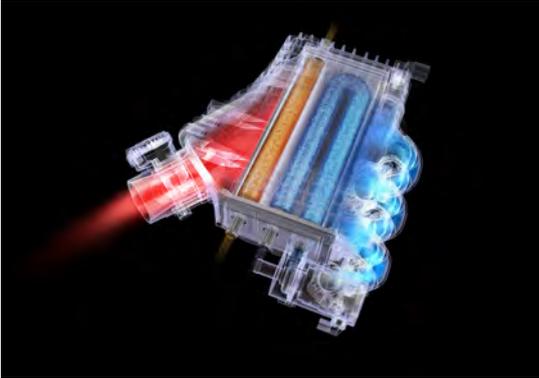


Abbildung 3: Technische Visualisierung: Ladeluftkühler von Mahle

Damit spielerische Elemente für eine Gamifizierung in eine technische Produktvisualisierung integriert werden können, wird eine Game Engine benötigt. Eine Game Engine ist ein Framework für die Programmierung von Computerspielen. Sie stellt Funktionen und Methoden bereit, die für die Steuerung und die Visualisierung des Spielverlaufs benötigt werden. Die am häufigsten verwendeten Game Engines sind derzeit die CryEngine, die Unity Engine und die Unreal Engine.

Eine der Aufgaben dieser Arbeit ist deshalb die Evaluierung verschiedener Game Engines. Die

dabei verwendeten Kriterien sind beispielsweise die unterstützten Programmiersprachen sowie Dateiformate und Lizenzfragen. Zur stereoskopischen Bildausgabe soll ein HMD verwendet werden, um eine möglichst immersive Visualisierung zu gewährleisten. Deswegen werden aktuell verfügbare HMDs auf verschiedene Kriterien hin geprüft, wie z.B. Latenzzeiten, Darstellungsqualität und Anschaffungskosten.

Ein besonderes Augenmerk muss auf das Eingabegerät gelegt werden. Anders als im Vergleich zu routinierten Computerspielern kann man davon ausgehen, dass potenzielle Anwender einer technischen Visualisierung deutlich weniger erfahren sind, um beispielsweise mit einem Gamepad zu interagieren. Es werden verschiedene Eingabegeräte evaluiert hinsichtlich einer intuitiven Bedienbarkeit.

Für die Optimierung der Usability der Anwendung werden im Verlauf der Entwicklung der Anwendung wiederkehrende Versuche mit Probanden durchgeführt. Mithilfe der daraus gewonnenen Daten können schließlich verschiedene Elemente der Anwendung, wie z.B. die Interaktionsmöglichkeiten, angepasst und verbessert werden. Die Anwendung selbst soll in verschiedene Abschnitte untergliedert sein. Die Untergliederung soll vor allem dazu dienen, die Anwendung in verschiedene Interaktionsstufen aufzuteilen. So soll zu Beginn das Erlebnis des Anwenders eher ruhig empfunden werden, um so den Anwender eine gewisse Eingewöhnungszeit zu bieten. Der User entscheidet dann selbst, wann und ob die nächsten Abschnitte und damit eine Steigerung der Interaktion, wie z. B. eine höhere Navigationsgeschwindigkeit, erreicht werden sollen.

Letztendlich soll das anhand der entwickelten Beispielanwendung erlangte Know-how als Basis für künftige Virtual Reality-Projekte dienen und die Beispielanwendung ein Demo-Projekt für die Firma madness GmbH bereitstellen.

[1] M. Brill: Virtuelle Realität, Springer Verlag, 2009

[2] M. Gutierrez, F. Vexo, D. Thalmann: Stepping into Virtual Reality, Springer Verlag 2008

[3] Gamification: <http://de.wikipedia.org/wiki/Gamification>, Zugriff am 10.05.2015

Bildquellen:

- Abbildung 1: IGI, Detroit, Michigan, USA, 2015
- Abbildung 2: Oculus VR, Menlo Park, California, USA, 2015
- Abbildung 3: madness GmbH, Göppingen, 2015

Konzeption und Implementierung eines Prototypen für einen adaptiven Empfehlungsdienst

Ismail Dogan*, Reiner Marchthaler, Werner Zimmermann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

In der heutigen Zeit nimmt der Funktionsumfang der Infotainmentsystem und die Anzahl der Fahrerassistenzsysteme im Fahrzeug stetig zu (siehe Abbildung 1).



Abbildung 1: Fahrerablenkung

75 Jahre nach Einführung des ersten Autoradios hat sich das Infotainmentsystem beachtlich entwickelt. Spätestens mit den Fahrerassistenzsystemen sind Fahrzeugfunktionen und Infotainmentsysteme zu einer Einheit verschmolzen. Ein Infotainmentsystem bezeichnet bei Fahrzeugen die Zusammenführung von Autoradio, Navigationssystem, Freisprecheinrichtung, die Anzeige von Fahrerassistenzsystemen und weitere Funktionen in einer zentralen Bedieneinheit. Des Weiteren entstand durch den Boom der Smartphones der Bedarf, diese auch während der Fahrt zu nutzen, daher wurden die Funktionen eines Smartphones, wie Nachrichten versenden, über das Internet Musik abspielen oder auch im Internet surfen in das Infotainmentsystem integriert. Wie in Abbildung 2 zu erkennen ist, wurden einzelne Funktionen zusammengefasst und sind immer mehr zu einer Einheit verschmolzen. Die Analoginstrumente wurden durch Displays verdrängt. Dadurch ergibt sich die Möglichkeit die Infotainmentsysteme flexibler zu gestalten. Die herkömmlichen Einzelinstrumente für die Informationen wie zum Beispiel Fahrgeschwindigkeit, Motordrehzahl und Tankfüllstand wurden zunächst durch kostengünstigere Kombiinstrumente, ersetzt (siehe Abbildung 2, Nr. 2).

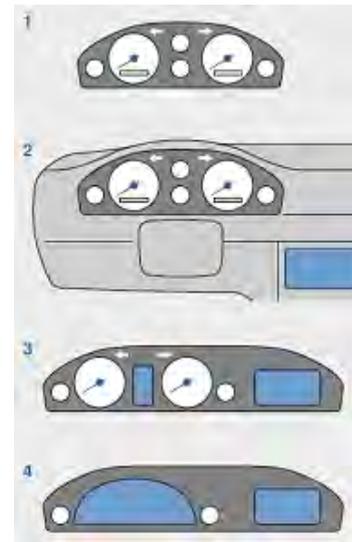


Abbildung 2: Historische Entwicklung des Infotainmentsystems

Kombiinstrumente fassen mehrere Informationseinheiten in einem Gehäuse zusammen. Mit der Zeit entstand im vorhandenen Bauraum bei ständigem Informationszuwachs das moderne Kombiinstrument mit mehreren Zeigerinstrumenten und zahlreichen Kontrollleuchten (siehe Abbildung 2, Nr. 3). Seit 2005 werden Displays für die Darstellung analoger Instrumente genutzt (siehe Abbildung 2, Nr. 4). Diese Technik wird die herkömmlichen Anzeigen aus Kostengründen allerdings nur langsam verdrängen.



Abbildung 3: Ein Bosch HMI-Demonstrator

*Diese Arbeit wurde durchgeführt bei der Firma Robert Bosch GmbH, Leonberg

Bei den Fahrerassistenzsystemen stehen oft Sicherheitsaspekte, aber auch die Steigerung des Fahrkomforts im Vordergrund. Wachsender Funktionsumfang der Infotainmentsysteme, die der Fahrer während der Fahrt nutzen möchte führen vermehrt zu Fahrerablenkungen. Um der Fahrerablenkung entgegen zu wirken sind Gegenmaßnahmen notwendig.

So entstand die Frage, wie der Fahrer bei der Nutzung der Systeme wie Infotainmentsysteme und Fahrerassistenzsysteme unterstützt werden kann. Als Lösungsansatz wurde ein adaptives System gewählt.

Der Fahrer soll durch einen proaktiven, adaptiven Empfehlungsdienst unterstützt werden, um die Fahrerablenkung zu reduzieren. Das Nutzungsverhalten des Fahrers soll beobachtet und daraus Gewohnheiten abgeleitet werden, die ihm dann in der entsprechenden Situation vorgeschlagen werden, um die Bedienschritte zur gewünschten Funktion zu reduzieren [1]. Zur Evaluierung des Empfehlungsdienstes soll ein Prototyp auf Serienhardware und mit Seriensoftware umgesetzt werden und in einer Fahrsimulatorumgebung de-

monstriert werden, um ein möglichst realistisches Erleben für die Probanden zu ermöglichen.

Der Bosch HMI-Demonstrator (siehe Abbildung 3) ist mit fünf Displays bestückt, die auch in der Serie verwendet werden. Des Weiteren wurde die Software für den Demonstrator aus einem Serienprojekt abgeleitet und in dem Demonstrator umgesetzt, um in der Zukunft die neuen Funktionalitäten in der Serienentwicklung effizienter zu integrieren.

Bei dieser Bachelorarbeit wird das Konzept der Softwarearchitektur des adaptiven Empfehlungsdienstes, unter der Berücksichtigung von Aufwand und ausgewählten Funktionen, die für den Demonstrator hinreichend sind, konkretisiert, entwickelt und in die bestehende Architektur (siehe Abbildung 4) eines Demonstrator integriert [2].

Zur Konzeptionierung der Architektur wurde das Tool Enterprise Architect verwendet. Für die Umsetzung wurden hauptsächlich die Tools Visual Studio, CGI Studio/ Scene Composer und IAR Visual State verwendet. Die Programmiersprache hierbei war C++.

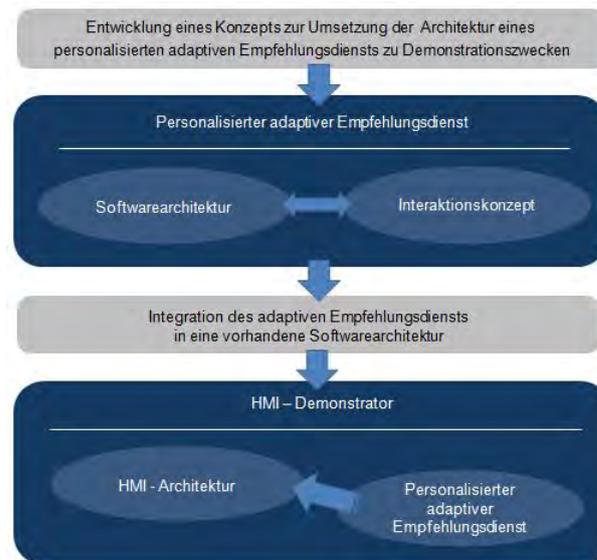


Abbildung 4: Integration des adaptiven Empfehlungsdienstes in die bestehende HMI-Architektur

[1] Nadine Siegmund, Personalized situation-adaptive User Interaction in the Car. AutomotiveUI, 2013.

[2] Ismail Dogan, Konzeption und Implementierung eines Prototypen für einen adaptiven Empfehlungsdienst. Bachelorarbeit, 2015.

Bildquellen:

- Abbildung 1: www.automotiveit.com
- Abbildung 2: Konrad Reif. Fahrstabilisierungssysteme und Fahrerassistenzsysteme. Vieweg + Teubner, 2010.
- Abbildung 3: Robert Bosch GmbH, Center of Competence HMI
- Abbildung 4: Eigene Darstellung

Positionsbestimmung durch 2D Koppelnavigation

Michael Dreher*, Reiner Marchthaler, Werner Zimmermann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Inertialsensoren erfreuen sich in der Gegenwart einer weiten Verbreitung, die eine Vielzahl von neuen Verwendungsmöglichkeiten mit sich bringt. Die einzelnen Sensoren sind, durch die Massenproduktion und fortschreitende technische Entwicklung, sehr klein und preiswert geworden. In der Vielzahl der Fälle ist die verwendete Trägerplattform vergleichsweise sehr kostspielig und wäre bei abhandeln nicht ohne weiteres zu ersetzen. Aus diesem Grund soll die Position eines solchen Trägersystems durch ein integriertes Überwachungssystem eingeschränkt werden.

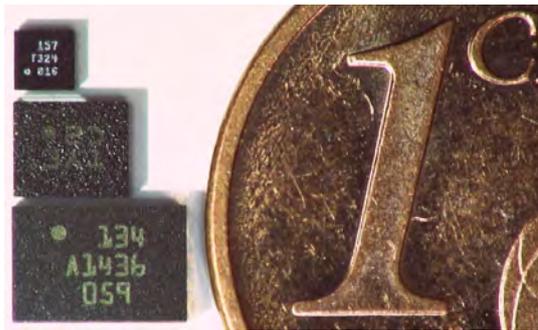


Abbildung 1: Inertialsensoren von Bosch Sensortec

Für diese Aufgabenstellung sind bereits mehrere Systeme am Markt vorhanden, diese ermitteln über einen GPS Sensor die aktuelle Position und alarmieren bei größeren Abweichungen. Der große Nachteil solcher Systeme ist die direkte Abhängigkeit von der externen Infrastruktur. Sollte zum Beispiel kein GPS-Signal vorhanden sein, so kann die Funktion nicht mehr gewährleistet werden. Dies ist schon in den meisten Gebäuden der Fall. Um diese Probleme zu vermeiden soll in dieser Arbeit eine autarke Lösung erarbeitet werden. Dies soll durch Positionserfassung mittels Inertialnavigation realisiert werden. Die Distanzüberwachung soll bei einer wählbaren Abweichung von 10–50cm zum ursprünglichen Standpunkt, oder bei Bewegungen länger als 60 Sekunden, Alarm schlagen.

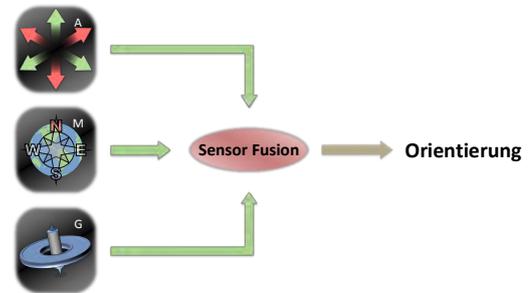


Abbildung 2: Sensorfusion

Zu diesem Zweck werden mehrere aufeinander folgende Bewegungen im Zwei-dimensionalen Raum erfasst. Zur Untersuchung stehen zwei unterschiedliche Plattformen zur Verfügung. Die Sensormodule kommen von der Bosch Sensortec GmbH (BST) aus Reutlingen und beinhaltet Beschleunigungs-, Drehraten- und Magnetfeldsensoren. Zur Steuerung wird ein ARM Cortex-M3 Mikrokontroller verwendet, an diesen sind zusätzlich ein Bluetooth Low Energy (BLE) Modul sowie ein Buzzer und eine LED angeschlossen.



Abbildung 3: Entwicklungsplattform auf Messwagen

Die Orientierung lässt sich über die vorhandene BST Sensorbibliothek BSX-Lite in aufbereiteter Form auslesen. Die BSX-Lite nutzt hierfür eine Sensorfusion zwischen den verwendeten Sensortypen um die jeweiligen Abweichungen

*Diese Arbeit wurde durchgeführt bei der Firma Bosch Connected Devices and Solutions GmbH, Reutlingen

der Einzelsensoren auszugleichen. Beschleunigungswerte sind ausschließlich in Rohform verfügbar und müssen entsprechend verarbeitet werden. Für die benötigten Algorithmen werden frühzeitig, zum Testen geeignete Bewegungsabläufe, als Systemtests definiert. Die zu den Bewegungsabläufen aufzuzeichnenden Sensordaten, werden in ihrer rohen und aufbereiteten Form aufgezeichnet. Auf Basis der Aufgezeichneten Daten werden anschließend geeignete Algorithmen, zur Verarbeitung, in Matlab entwickelt. Ein geeigneter Algorithmus soll letztendlich als lauffähiger Code in die Ge-

räteplattformen integriert und durch Systemtests verifiziert werden. Bei der Realisierung wird nicht nur die Beschleunigung sondern auch Rauschen integriert was zu sehr großen Fehlern und somit einer sehr großen Toleranz führt. So kann mit großer Wahrscheinlichkeit der überwachte Radius nicht genau genug gemessen werden. Es muss vielmehr ein Bereich zwischen zwei Radien definiert werden in dem das Gerät Alarm schlagen soll. Die zuvor erwähnte Zeitüberschreitung wird an den gewählten Bereich angepasst um das Zeitliche abdriften durch Fehler einzugrenzen.

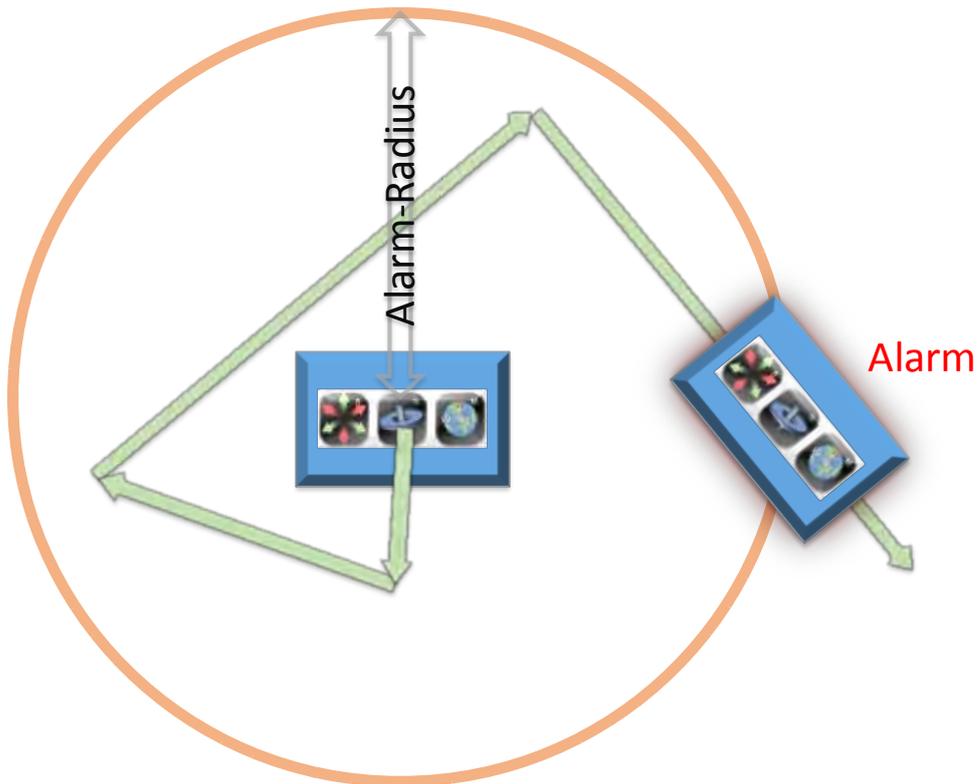


Abbildung 4: Alarmerkennung beim Verlassen eines Radius

Bildquellen:

- Abbildung 1-4: Eigene Darstellung

Optimierte Datenübertragung für dynamisch anwachsende Datenblock-Ressourcen in REST Systemen

Alexander Ehni*, Reiner Marchthaler, Werner Zimmermann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

In der Veröffentlichung „The Computer for the 21st Century“ [1] beschreibt Mark Weiser mit den Worten „Ubiquitous computing“ 1991 zum ersten Mal den Grundgedanken einer rechnergestützten zukünftigen Welt. In dieser Zukunftswelt sollen herkömmliche Computer durch intelligente Geräte, die den Menschen unmerklich bei seinen Aktivitäten unterstützen, abgelöst werden. Ein aktuelles Beispiel für intelligente Geräte stellt in diesem Zusammenhang eine Smartwatch dar. Eine Smartwatch ist eine Uhr, die durch eine Computereinheit erweitert wurde und am Handgelenk getragen werden kann. Diese Kombination aus Armbanduhr und Computer ermöglicht dem Nutzer z.B. über neue Nachrichten auf dem Handy informiert oder bei seinen Sportaktivitäten unterstützt zu werden.



Abbildung 1: IoT (Internet of things)

Beschäftigt man sich näher mit den technischen Rahmenbedingungen, so trifft man auf das Thema „Internet of Things“. In diesem Konzept „kommunizieren Dinge, intelligente Objekte, Smart Objects, miteinander“. [2] Dabei ist der Begriff „Dinge“ weit gefasst und kann im Kontext mit Personen und Tieren, Haushaltsgeräten, Kraftfahrzeugen, medizinischen Geräten, Computern, Sensoren, Stromnetze und vielen anderen Objekten stehen. [2] „Jeder einzelne Gegenstand eines Internet of Things ist eindeutig durch eine Internetadresse (URL) gekennzeichnet und kann über das Internet angesprochen werden und mit angeschlossenen Informationssystemen interagieren“. [2] Bei der Interaktion zwischen Informationssys-

temen und intelligenten Geräten ist eine effiziente Datenübertragung essentiell. Diese Datenübertragung beinhaltet neben einem robusten Übertragungsprotokoll die Eigenschaft, in sehr kurzer Zeit mehrere Geräte mit Informationen zu versorgen. Dabei findet der Informationsaustausch überwiegend per Funk statt, wie z.B. dem General Packet Radio Service (GPRS).

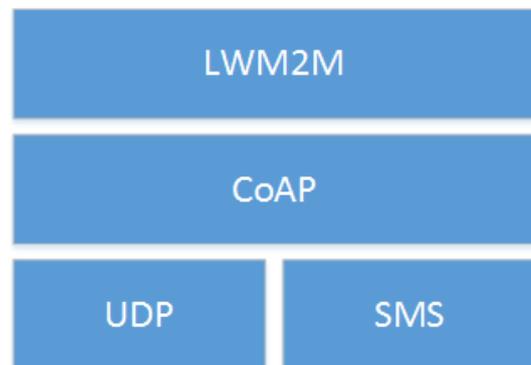


Abbildung 2: LWM2M Protokollstack

Um diese Datenverbindung möglichst leistungsfähig zu gestalten, kann das UDP-Protokoll als Transportprotokoll für IP-basierte Verbindungen verwendet werden. Das UDP-Protokoll bietet die Möglichkeit, mit Hilfe von Sockets, eine verbindungslose Datenkommunikation für höhere Protokollschichten bereitzustellen. Erweitert man nun den zuvor beschriebenen Protokollstack auf Applikationsebene durch das Constraint Application Protocol, so wird eine Möglichkeit geschaffen, Daten und Ressourcen zwischen Machine to Machine (M2M) Kommunikationspartnern auszutauschen. Das Constraint Application Protocol (CoAP) ist ein Kommunikationsprotokoll für leistungsschwache Mikroprozessoren mit dem Ziel, Datenressourcen zwischen M2M Teilnehmern auszutauschen. Als Weiterentwicklung des CoAP-Protokolls gilt das Lightweight M2M Protokoll (LWM2M). Hierbei handelt es sich um ein Managementprotokoll, welches ein standardisiertes Geräte Management für das „Internet of Things“ bereit stellt und

*Diese Arbeit wurde durchgeführt bei der Firma Bosch Connected Devices and Solutions, Reutlingen

Datenkommunikation auf Basis von Ressourcen erlaubt. In dieser Bachelorarbeit soll ein neuer Ansatz für eine effiziente Maschine to Machine (M2M) Übertragung von dynamisch anwachsenden Datenblock Ressourcen untersucht werden.

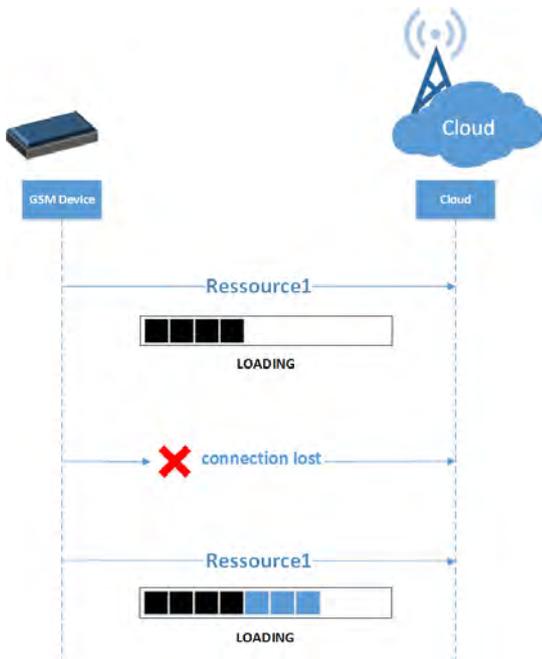


Abbildung 3: Unterbrechung der Funkverbindung

Die derzeitige Problematik bei der Übertragung innerhalb von CoAP/LWM2M Systemen besteht darin, dass bei einer Erweiterung von Datenblock-Ressourcen oder bei einem Verbindungsabbruch des Funksystems während einer Übertragung, die gesamte Ressource erneut gesendet werden muss. Nach einer erfolgreichen Planungsphase soll das neue Konzept auf einem Linux Betriebssystem umgesetzt und später auf ein Embedded Gerät mit Funkverbindung portiert werden.

Damit das Ziel, die Übertragung einer dynamisch anwachsenden Datenblockressource erreicht werden kann, müssen die speziellen Eigenschaften des Lightweight Machine to Machine Protokolls mit den Eigenschaften des Constraint Application Protokoll kombiniert werden. Die Kombination aus zwei Protokollebenen erlaubt es, eine Datenübertragung für dynamisch anwachsende Datenblockressourcen bereitzustellen. Zusätzlich besteht die Möglichkeit, bei einem Verbindungsabbruch, an der letzten bekannten Datenstelle die Übertragung fortzusetzen und eine redundante Datenübertragung zu vermeiden.

[1] Weiser Mark, The Computer for the 21st Century, Veröffentlichung Scientific American [1991]

[2] IT Wissen, IoT (Internet of things), <http://www.itwissen.info/definition/lexikon/Internet-of-things-IoT-Internet-der-Dinge.html>

Bildquellen:

- Abbildung 1: <http://www.gadgetreview.com/wp-content/uploads/2015/02/IoT-Graphic.png>
- Abbildung 2,3: Alexander Ehni

Entwurf, Implementierung und Laufzeitanalyse einer Java-API unter der Verwendung einer Basis-Software-Architektur auf einem Embedded-Linux-System zur Steuerung eines autonomen Rasenmähers

Tobias Fillbrandt*, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Roboter gewinnen im Alltagsleben und Haushalt immer mehr an Bedeutung und werden schon seit geraumer Zeit verwendet, um gewisse Arbeiten zu erleichtern, wie zum Beispiel das Staubsaugen oder das Reinigen eines Pools. Automatisierte Vorgänge mit Hilfe von Robotern werden durch die Verknüpfung mit Smart-Home-Technologien schon in naher Zukunft in unserem Alltagsleben ein fester Bestandteil sein [3]. Eine Umfrage prognostiziert für das Jahr 2020 in einem konservativen Szenario schon eine Million Smart-Home-Haushalte in Deutschland und in einem progressiven Szenario sogar 1,45 Millionen Haushalte [4].



Abbildung 1: Indego Rasenmäher

Der autonome Rasenmäher "Indego" der Firma Bosch GmbH bietet neben dem autonomen Mähen und Aufladen der Batterie des Rasenmähers ein Navigationssystem an, um eine Rasenfläche mittels Kartierung und Berechnung sauber und effizient in Reihen zu mähen. Bestimmte Modelle enthalten auch schon Lösungen zur Steuerung mit mobilen Endgeräten, so dass das Mähen im heimischen Garten vom Urlaubsort aus oder auf Reisen ermöglicht wird. (Abbildung 1)

Aufgrund des wachsenden Interesses wird das existierende Indego-Rasenmäher-System weiterentwickelt und mit zusätzlichen Funktionen ausgestattet. Um die neuen Funktionen während der Entwicklungsphase in das be-

währte System integrieren zu können, wird ein Rechner als Bypass hinzugefügt und mit Hilfe einer seriellen Schnittstelle mit dem Rasenmäher verbunden. Messwerte oder IST-Werte werden zyklisch vom Rasenmäher gesendet und ebenso werden SOLL-Werte zur Einspeisung in die entgegengesetzte Richtung vom Bypass an den Rasenmäher übermittelt. Dadurch wird erreicht, dass neue Funktionen implementiert werden können, ohne das eingebettete System des Rasenmähers zu verändern. Zusätzlich wird mit der hardwareunabhängigen Entwicklung wertvolle Zeit im Entwicklungsprozess eingespart. Ein erheblicher Nachteil stellt jedoch die Kabelverbindung der seriellen Schnittstelle dar. Um die Entwicklung noch weiter zu optimieren, ist das Ziel dieser Arbeit, eine Java-API (Application Programming Interface) auf einem Embedded-Linux-System zu entwickeln, die eine Wifi-Verbindung des Bypasses mit dem Rasenmäher ermöglicht. Das zusätzliche Linux-Board wird in den Rasenmäher eingebaut und mit der existierenden seriellen Schnittstelle verbunden. Zur Bewertung der entwickelten API soll diese mit Hilfe einer Laufzeitanalyse evaluiert werden.

Die Softwarearchitektur enthält verschiedene Abstraktionsebenen, die den Ablauf der Informationsverarbeitung innerhalb der API beschreiben. Zuerst werden die Rohdaten der seriellen Schnittstelle eingelesen und als Nachricht mit ID und Zeitstempel interpretiert. Die detaillierten Informationen der Nachricht werden in der darauffolgenden Abstraktionsebene als Ressourcen abgelegt und einem REST-Webservice bereitgestellt. (Abbildung 2) REST (Representational State Transfer) ist ein ressourcenorientierter Architekturstil mit dem Fokus auf Skalierbarkeit, Erweiterbarkeit und Interoperabilität [2]. Die Voraussetzungen Client-Server, Uniform Interface, Code on Demand der REST-Architektur ermöglichen eine einfache M2M-Kommunikation (Machine to Machine) zum Austausch von Informationen bei verteilten Systemen [1].

*Diese Arbeit wurde durchgeführt bei der Firma Robert Bosch GmbH, Renningen

Als Hardware wird das Cubietruck der Firma Cubietech verwendet, welches mit einem 1GHz ARM®Cortex™-A7 Dual-Core und einem integrierten Wifi-Modul ausgestattet ist. Mit der Integration der Java-API erwartet man außerdem neue Erkenntnisse über das Ver-

halten der verwendeten Basis-Architektur auf Embedded-Geräten wie die des Cubietrucks. Durch das zusätzliche Linux-Board im Rasenmäher und der Java-API könnten zudem neue Anwendungsfälle für einen vernetzten autonomen Rasenmäher entstehen.

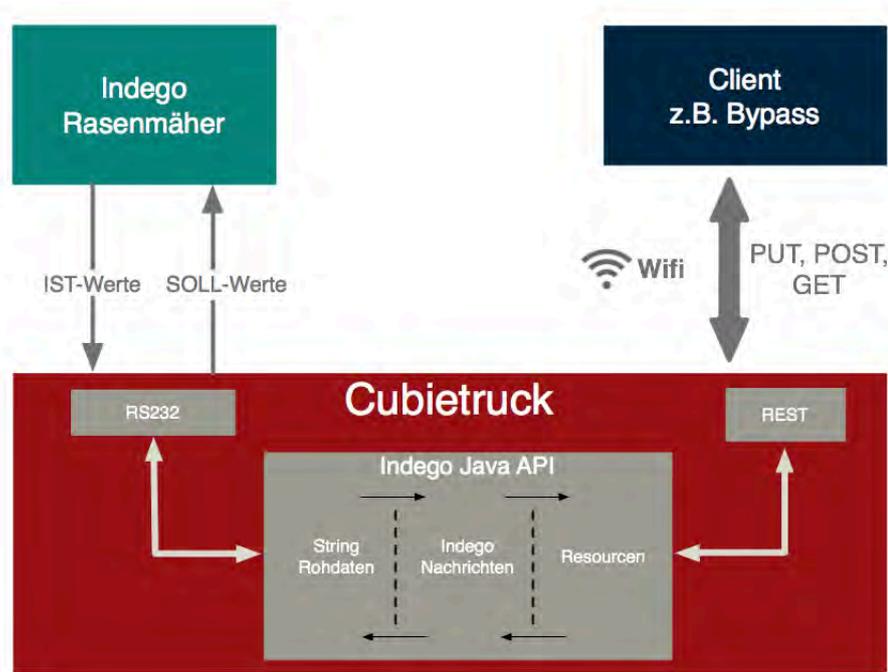


Abbildung 2: Systemübersicht

- [1] Michael Dazer, RESTful APIs – Eine Übersicht, Technical University Berlin, https://www.snet.tu-berlin.de/fileadmin/fg220/courses/WS1112/snet-project/restful-apis_dazer.pdf, Letzter Aufruf am 10.5.2015
- [2] Cesare Pautasso, REST: Advanced Research Topics and Practical Applications, Springer, 2014 (1.Auflage)
- [3] Matthias Haun, Handbuch Robotik: Programmieren und Einsatz intelligenter Roboter, Springer Vieweg, 2013 (2.Auflage)
- [4] Deloitte, So smart wohnen die Deutschen zukünftig, Statista, 2013

Bildquellen:

- Abbildung 1: <http://www.venovatoools.ch/media/images/org/BoschIndego.jpg>
- Abbildung 2: Eigene Abbildung

Entwicklung von Methoden zur Objekt-Spur-Allokation für Fahrerassistenzsysteme in schweren Nutzfahrzeugen

Özgen Gönül*, Martin Stämpfle, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Seit einigen Jahrzehnten beschäftigen sich Fahrzeughersteller und -zulieferer hinsichtlich der Fahrzeugsicherheit mit der Optimierung der Fahrzeuge und deren Komponenten. Aufgrund dessen ist die Zahl der Verkehrsunfälle in Folge technischer Defekte deutlich zurückgegangen. Um weiterhin Verkehrsunfälle vermeiden zu können, beziehungsweise die Unfallfolgen zu minimieren, wurde ein besonderes Augenmerk auf die Unterstützung des Fahrzeugführers gelegt. Daraufhin erfolgte der erste Durchbruch mit der Entwicklung des Antiblockiersystems (ABS) in den 70er Jahren.

In schweren Nutzfahrzeugen wurde das ABS erst im Jahr 1981 eingeführt. Das Elektronische Stabilitätsprogramm (ESP) wurde im Jahr 2001 im Lkw-Bereich eingeführt. Seit 1. November 2014 ist ESP bei allen, neu zugelassenen Nutzfahrzeugen in Deutschland gesetzlich vorgeschrieben [1] [2]. Diese und weitere Fahrsicherheitsysteme haben trotz steigender Transportleistung zu einem Rückgang der tödlichen Unfälle geführt. Die folgende Abbildung veranschaulicht den Rückgang der Lkw-Unfälle in den vergangenen Jahren [3].

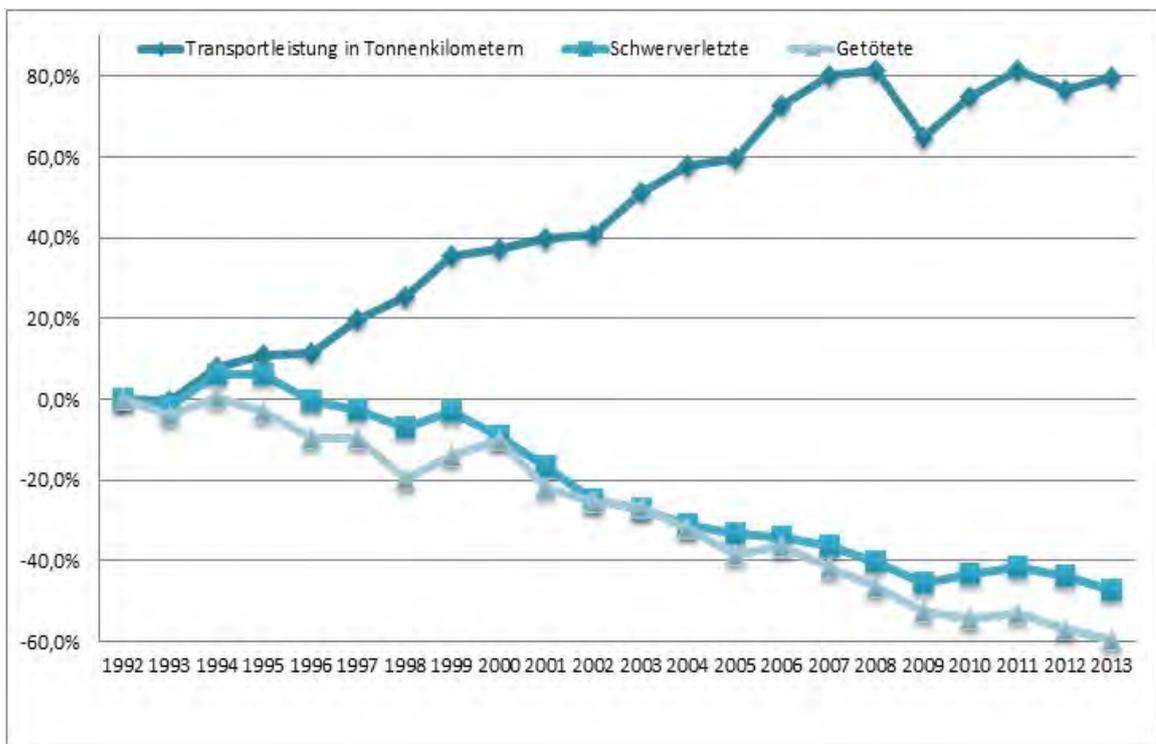


Abbildung 1: Getötete und Schwerverletzte bei Lkw-Unfällen auf deutschen Straßen im Vergleich zur Lkw-Transportleistung

Der Notbrems-Assistent in schweren Nutzfahrzeugen, auch „Active Brake Assist“ genannt, ist die jüngste Entwicklung unter den Fahrerassistenzsystemen, die heute im Einsatz sind. Es handelt sich hierbei um ein prädiktives

Sicherheitssystem, welches sich in Personenkraftwagen sowie in Nutzfahrzeugen als hilfreich erweisen kann. Bei der Erkennung eines unvermeidbaren Auffahrunfalls übernimmt der Notbremsassistent die Aufgabe, eine autono-

*Diese Arbeit wurde durchgeführt bei der Firma Daimler AG, Untertürkheim

me Notbremsung des Fahrzeugs einzuleiten. Um Gefahrensituationen von „normalen“ Fahr-situationen zu unterscheiden, ist eine richtige Erkennung von Objekten vor und neben der Fahrbahn elementar [4].



Abbildung 2: Mercedes-Benz Actros bei einer Notbremsung

Die Aufgabe dieser Arbeit besteht darin, relevante, also potentiell gefährliche Objekte, von nicht-relevanten Objekten zu unterscheiden. Als Relevanzbereich wird die Spur des eigenen Fahrzeugs bezeichnet. Befindet sich ein stehendes oder fahrendes Objekt eindeutig auf dieser Spur, wird dieses Objekt als relevant eingestuft. Befindet es sich eindeutig neben dieser, ist es nicht-relevant. Die Evaluierung, ob eine Gefahrensituation vorliegt, erfolgt über den Abstand und die Geschwin-

digkeitsdifferenz zum eigenen Fahrzeug [5].

Um neu entwickelte Algorithmen zu testen, wird mittels PreScan-Software eine simulierte Fahrzeugumgebung geschaffen, die in sämtliche Fahrsituationen nachgestellt werden können. PreScan bildet aus diesen Testscenarien und simulierten Sensoren ein MATLAB Simulink-Modell. Der zu testende Algorithmus wird in dieses Simulink-Modell integriert und kann somit auf Fehler und Bugs überprüft werden. Dieses Vorgehen hat den Vorteil, dass sehr komplexe Testscenarien erstellt werden können, die im öffentlichen Verkehr nicht durchführbar sind.

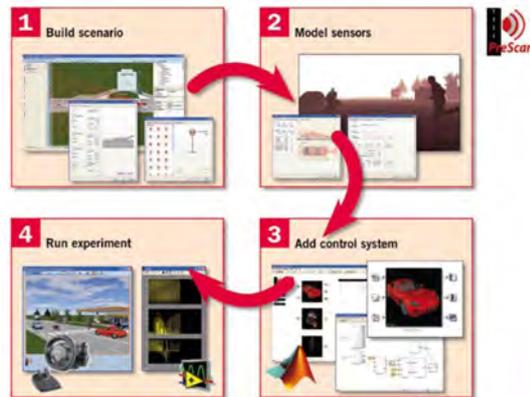


Abbildung 3: Vier Schritte der Entwicklung und des Testens in PreScan

- [1] <http://blog.daimler.de/2014/07/02/gastbeitrag-was-eine-unfallversicherung-mit-lkw-assistenzsystemen-und-automatisiertem-fahren-zu-tun-hat/>
 [2] <http://www.kfz-auskunft.de/news/5438.html>
 [3] BG-Verkehr – http://www.bg-verkehr.de/arbeitssicherheit-und-gesundheitsschutz/aktionen-und-kampagnen/fahrer-assistenz-systeme/statistische-hintergrundinformationen/04%208-1_Lkw-Unfaelle_und_Lkw-Transportleistung_1992-2006.pdf/view
 [4] K.-L. Haken, Grundlagen der Kraftfahrzeugtechnik, Esslingen: Carl Hanser Verlag München, 2011
 [5] <http://www.daimler.com/dccom/0-5-1210220-49-1521413-1-0-0-1210338-0-0-135-0-0-0-0-0-0-0.html>

Bildquellen:

- Abbildung 1: BG-Verkehr-statistische Hintergrundinformationen
- Abbildung 2: Daimler AG
- Abbildung 3: ATZautotechnology

Secure-Boot-Konzept für Embedded-System-Rechnersysteme

Patrick Jakob*, Werner Zimmermann, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

F&S Elektronik Systeme entwickelt und fertigt sogenannte eingebettete Computer Module. Diese Module verfügen über eine leistungsfähige ARM CPU (Cortex-A Prozessor). Auf den Modulen kommt ein Betriebssystem wie z.B. Linux oder Windows Embedded zum Einsatz. Der Einsatz dieser Module in der Industrie- und Medizintechnik nimmt kontinuierlich zu. Dabei hat die Verfügbarkeit dieser Geräte höchste Priorität. Um dies zu gewährleisten müssen die Geräte stabil laufen also geringe Ausfallquoten haben und natürlich robust sein, da die Umgebung im industriellen Bereich hohe EMV, Temperatur-, Feuchtigkeits- oder sonstige mechanische Beanspruchung darstellen kann.

Ein wichtiger Punkt der immer mehr an Bedeutung gewinnt, ist das Thema Security. Gründe hierfür sind die zunehmende Vernetzung der Geräte, der Schutz geistigen Eigentums und der Schutz gegen Manipulation. Da das Thema Security sehr großflächig ist, sollte man sich als erstes die Frage stellen, gegen welche Art von Angriff der Schutz wirken soll. Im Rahmen der Arbeit steht der Schutz des Bootprozesses gegen Manipulation und Auslesen des Quellcodes im Vordergrund. Um näher auf die Sicherheitsfunktionen eingehen zu können, wird zuerst der Bootvorgang erläutert.

Der Bootvorgang der eingebetteten Geräte besteht aus 3 Bootloadern (Abb. 1). Sobald das Gerät gestartet wird, startet der CPU ROM-Loader, der vom Hersteller der CPU implementiert wurde. Dieser liest den F&S eigenen First Level Bootloader N-Boot aus dem Flash und startet diesen. Aufgabe des N-Boot ist die Initialisierung des DRAM Speichers, Einstellung der Clocks und Initialisierung von Schnittstellen, die für den weiteren Bootvorgang notwendig sind. Der N-Boot lädt den für das jeweilige Betriebssystem spezifischen Bootloader U-Boot (Linux)/E-Boot (Windows) und führt diesen aus. Nachdem der U-Boot/E-Boot gebootet wurde, wird der Linux- oder Windows-Kernel ausgeführt. Dieser implementiert die jeweiligen Funktionen und lädt das Root Filesystem, von dem später die Benutzeranwendung ausgeführt wird.

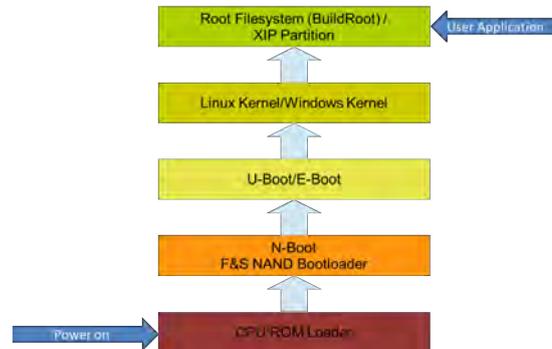


Abbildung 1: Bootvorgang des eingebetteten Systems

In der Abschlussarbeit geht es um die untersten beiden Bootebenen, den sogenannten N-Boot und U-Boot. Der eingesetzte Prozessor stellt eine Vielzahl an Sicherheitsfunktionen zur Verfügung. Diese Funktionen gehören zu einem Modul namens High Assurance Boot (HAB). Wir entschieden uns bei der Umsetzung für die Verifizierung und die Verschlüsselung des N- und U-Boots. Mit der Verifizierung des Bootloaders stellt man sicher, dass nur der Bootloader, der korrekt signiert wurde, geladen und ausgeführt werden darf. Die Signierung wird auch als digitale Unterschrift bezeichnet. Um dies umzusetzen bietet der Hersteller des Prozessors ein Programm an, das die Signierung des Bootloaders vornimmt. Zum besseren Verständnis des Signierungsprozesses wird dieser nun erläutert (Abb. 2).

Um eine Signierung zu erstellen, werden digitale Zertifikate benötigt. Diese werden von einer Zertifizierungsstelle (kurz CA) ausgestellt. Ein digitales Zertifikat ist vergleichbar mit einem digitalen Ausweis und wird in unserem Fall beim Public-Key-Verfahren eingesetzt. [1] Dies bedeutet, es werden 2 verschiedene Schlüssel eingesetzt. Der Public Key, der öffentlich zugänglich sein kann, und der Private Key der geheim und nur dem Inhaber bekannt ist[2].

Um die Signierung vorzunehmen wird über den eigentlichen Bootloader eine Hash-Summe gebildet. Diese wird mit dem Private Key verschlüsselt und es entsteht eine Binärdatei. Die entstandene Datei wird als Signierung be-

*Diese Arbeit wurde durchgeführt bei der Firma F&S Elektronik Systeme, Stuttgart-Vaihingen

zeichnet und an den eigentlichen Bootloader angehängt. Das neu entstandene Image wird auf dem eingebetteten Gerät im Flash abgespeichert. Nach einem Neustart überprüft der ROM-Loader den aus dem Flash geladenen Bootloader. Dazu liest der ROM-Loader den Public Key aus den sogenannten OTP (One-Time-Programmable) Fuses und entschlüsselt damit die Signatur des Bootloaders. In der Signatur befindet sich die Referenzhash für den Bootloader. Danach berechnet der ROM-Loader die Hash für den Bootloader. Wenn diese identisch sind, darf der Bootloader gestartet werden. Wenn nicht wird die Ausführung verweigert.

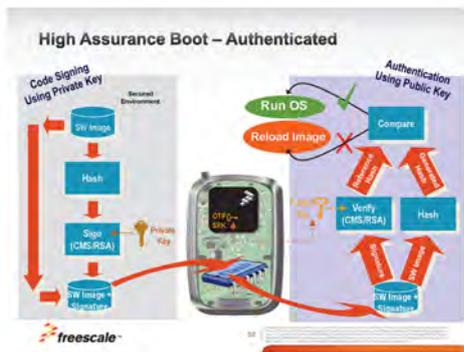


Abbildung 2: Signierung und Authentifizierung des Bootloaders

Zum Verschlüsseln des Bootloaders wird ein Private Key benötigt, der ebenfalls verschlüsselt und auf dem eingebetteten Gerät abgespeichert wird. Sobald das Image geladen wird, entschlüsselt das System den verschlüsselten Private Key.



Abbildung 3: Ver- und Entschlüsselung des Bootloaders

Sobald diese beiden Sicherheitsfunktionen aktiv sind (Authentifizierung und Verschlüsselung des Bootloaders), kann ein Angreifer weder ein nicht signiertes Image noch ein falsch signiertes Image übertragen und speichern. Zugleich kann dieser auch nicht den Code des Boot Image auslesen, da dieser verschlüsselt ist.

- [1] IT-Wissen Digitales Zertifikat: <http://www.itwissen.info/definition/lexikon/Digitales-Zertifikat-digital-certificate.html>
 [2] IT-Wissen Public-Key-Verfahren: <http://www.itwissen.info/definition/lexikon/Public-Key-Verfahren-public-key-method.html>

Bildquellen:

- Abbildung 1: F&S Elektronik Systeme – interne Quelle
- Abbildung 2 + 3: Freescale Vybrid Security Solutions: http://www.freescale.com/files/training_pdf/FTF/2012/americas/WBNR__FTF12_ENT_F0043.pdf?lang_cd=en

Entwicklung eines Moduls für die Verarbeitung von analogen Eingangs- und Ausgangsdaten als Erweiterung des EasySlave Evaluation-Kits für Sercos III einschließlich Implementierung der Logik für den synchronen Betrieb des auf einem Xilinx-FPGA basierenden EasySlaves

Sebastian Klemm*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Netzwerke und Systeme mit harten Echtzeitanforderungen sind aus unserer heutigen Zeit nicht mehr wegzudenken. Vor allem in der Automatisierungsindustrie kommen solche Echtzeitnetzwerke verstärkt zum Einsatz. Hierfür bietet Sercos mit Sercos III ein offenes und IEC-konformes Netzwerk, das neben den harten Echtzeitanforderungen zusätzlich auch noch eine hohe Ausfallsicherheit gewährleisten kann. Da das Sercos III-Protokoll auf dem Ethernet-Standard basiert, lassen sich die Sercos III-Komponenten untereinander einfach mittels Ethernet (Lichtwellenleiter oder Twisted-Pair-Kabel) verbinden. Eine hohe Ausfallsicherheit gewährleistet hier insbesondere die Verwendung einer Ring-Topologie, da das Netzwerk auch im Falle eines Kabelbruchs weiterhin funktionsfähig bleibt. Dies wird dadurch gewährleistet, dass ein Sercos III Ring bei Kabelbruch in zwei durch den Master verbundene Liniensegmente zerfällt (vergleiche Abbildung 1) [1].

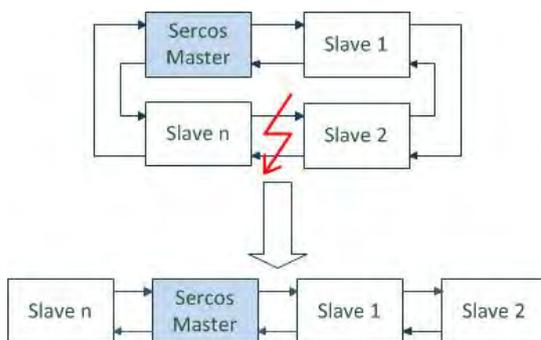


Abbildung 1: Ring-Topologie wird zu Linien-Topologie bei Kabelbruch

Für eine einfache Anbindung von Sercos III-Slaves gibt es den sogenannten Sercos EasySlave. Dieser umfasst einen FPGA IP-Core sowie die dazugehörigen Softwarebibliotheken. Der FPGA IP-Core bietet hierbei den kompletten Funktionsumfang eines vollwertigen Sercos III-Slaves. Dies ermöglicht eine einfache Integration des Slaves in ein Sercos-Netzwerk. Darauf

aufbauend bietet das Sercos EasySlave Evaluation Kit des Steinbeis-Transferzentrum Systemtechnik eine komfortable Möglichkeit digitale IO-Geräte zu integrieren. Dieses Kit umfasst ein FPGA-Modul mit integriertem Sercos EasySlave, sowie eine Erweiterungsplatine für den Anschluss von digitalen I/O-Geräten. Der Kern des FPGA-Moduls ist hierbei ein FPGA vom Typ Spartan 6 des Herstellers Xilinx [2].

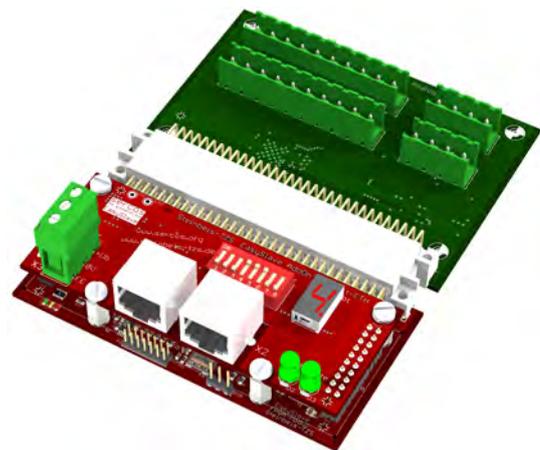


Abbildung 2: FPGA-Modul (rot) mit analoger Erweiterungsplatine (grün)

Im Rahmen der Bachelorarbeit wurde eine weitere Erweiterungsplatine für analoge I/O-Geräte realisiert (vergleiche Abbildung 2). Diese arbeitet mit dem bestehenden FPGA-Modul zusammen, was durch die Anpassbarkeit der FPGA-Hardware ermöglicht wird. Die Ansteuerung der auf der Erweiterungsplatine verbauten Komponenten wurde für die vorliegende Arbeit mittels VHDL-Modulen realisiert. Diese Module werden im FPGA als eigene Hardware-Bausteine umgesetzt. Dadurch ist es möglich, die Ansteuerung der analogen I/O-Geräte sowie die Sercos III-Kommunikation parallel, also unabhängig voneinander, zu implementieren. Weiterhin sollen dadurch Zykluszeiten von unter 125 Mikrosekunden bei der Verarbeitung analoger Daten erreicht werden.

*Diese Arbeit wurde durchgeführt bei der Firma Bosch Rexroth, Lohr am Main

Für die analoge Erweiterungsplatine wurden ein 16-Bit Analog-Digital-Wandler mit insgesamt acht Eingangskanälen sowie zwei 16-Bit Digital-Analog-Wandler mit je einem Ausgangskanal verwendet. Eine Besonderheit bei den Digital-Analog-Wandlern ist, dass diese im sogenannten „Daisy Chain“-Modus betrieben werden. Sie sind also in Reihe geschaltet und beide Wandler können mit nur einer Datenleitung angesprochen werden.

Ein weiterer Schwerpunkt der Arbeit war es, alle Komponenten synchron zueinander zu verwenden. Hierfür bietet Sercos III die Möglichkeit, alle Sercos III-Slaves zu synchronisieren. Damit dies möglich ist, müssen für die Eingänge der einzelnen Slaves (Port A und Port B) jeweils die Synchronisations-Offsets ermittelt werden. Hierfür wird durch den Master die Ring-Verzögerungszeit, also die Zeit, die ein Paket im Ring für einen kompletten Durchlauf benötigt, ermittelt. Anschließend ergibt sich der Synchronisations-

Offset aus dem Delta zwischen der Ring-Verzögerungszeit und der Zeit, die ein Paket bis zum entsprechenden Port benötigt [3]. Dieser Synchronisations-Offset wird für jeden der beiden Ports separat berechnet und im Slave gespeichert. Hierbei ist zu beachten, dass die Summe der Synchronisations-Offsets von Port A und Port B der Ring-Verzögerungszeit entspricht. Wenn während eines Kommunikationszyklus die Synchronisationsaufforderung kommt (in Abbildung 3 innerhalb des MST Blocks), kann der Slave nun mittels dem Synchronisations-Offset des entsprechenden Ports sowie der vom Sercos III-Master vorgegebenen Synchronisations-Verzögerungszeit einen mit allen anderen Slaves und dem Master synchronen Zeitpunkt bestimmen. Im Falle der analogen Erweiterungsplatine wird zu diesem Zeitpunkt die Konvertierung der Daten an den Wandlern gestartet. Dies gewährleistet eine synchrone Datenverarbeitung aller Sercos III-Slaves.

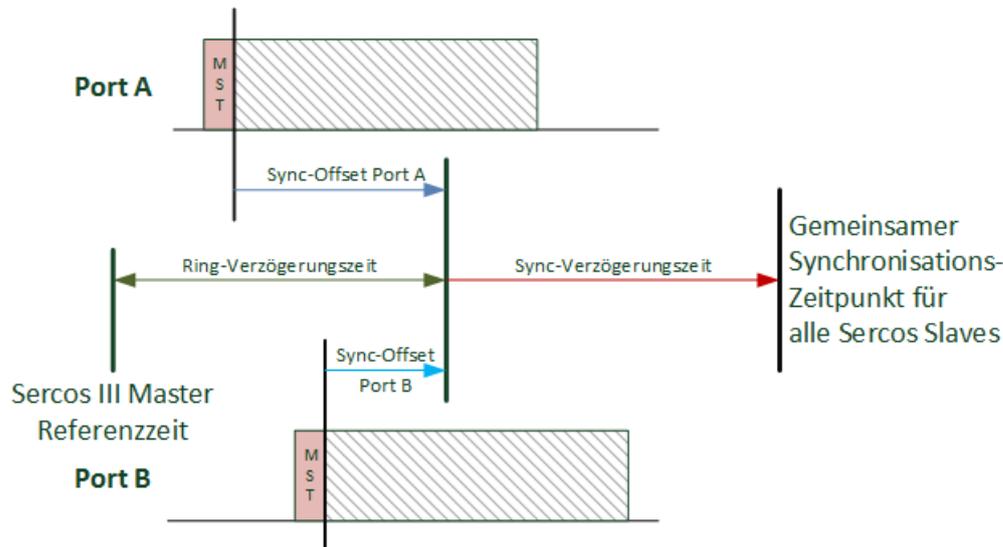


Abbildung 3: Synchronisation eines Sercos III-Slaves

[1] Feldbusse.de: Sercos, <http://www.feldbusse.de/sercos/sercos.shtml>
Stand: 05.05.2015

[2] Steinbeis TZS: Sercos EasySlave Evaluation Kit, <http://www.steinbeis-tzs.de/easyslave/>
Stand: 05.05.2015

[3] Sercos International: Communication Specification, Internes Wiki. Version: 1.3.1-1.12,
Stand: 06.05.2015

Bildquellen:

- Abbildung 1,3: Sebastian Klemm
- Abbildung 2: Steinbeis TZS

Integration von Smart Home Geräten in ein bestehendes Hausautomatisierungs-System über Internet-Protokolle

Kevin Kögler Villafuerte*, Werner Zimmermann, Karlheinz Höfer

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Automatisierung wird seit jeher eingesetzt um Abläufe zu optimieren. Als bekannte Beispiele lassen sich Landwirtschaft und Industrie nennen. Die Ausmaße der Produktion von Gütern und ihre Effizienz haben durch Automatisierung immer weiter zugenommen. Seit jüngerer Zeit erlangte auch die Automatisierung in den eigenen vier Wänden größere Beliebtheit. Durch Fortschritt der Technik ergaben sich Möglichkeiten der Vernetzung sowie mit Smartphones ideale Steuerelemente für eingebettete Systeme. Namhafte Hersteller bieten eigene Smart Home Geräte mit proprietären Steuerungen an. Diese sind nicht kompatibel mit Produkten anderer Hersteller oder anderen Funkstandards. Das Smart Home System namens homee hat sich diesem Problem angenommen. Homee vereint mehrere Funkprotokolle und unterstützt damit verschiedene Produkte unterschiedlicher Hersteller. Durch logische Verknüpfungen zwischen diesen Geräten entsteht ein Mehrwert, welche sonst nicht möglich wäre.



Abbildung 1: Smart Home mit dem homee

Im Rahmen dieser Bachelorarbeit werden drei Smart Home Geräte in das homee-System integriert. Das Ziel ist es zum einen, beispielhafte Kommunikationsabläufe für Geräte mit Internet-basierten Datenübertragung zu erarbeiten, sowie diese in die gewohnte Steuerung des homee-Systems einzubinden. Bei diesen Geräten handelt es sich um Schaltsteckdosen von AVM, eine Wetterstation von netatmo sowie Pflanzen-Sensoren für Pflanzentöpfe von Koubachi. AVM Router sind in Deutschland

weit verbreitet. 2010 waren sie mit 68% Anteil Marktführer in diesem Segment [1]. Mit ihren intelligenten Schaltsteckdosen arbeiten sie an einer eigenen Smart Home Lösung. Netatmo bietet mit seiner Wetterstation eine große Anzahl Sensoren für Innenräume sowie auch für Außenbereiche. Mit zusätzlichen Modulen können weitere Messdaten gesammelt werden, wie zum Beispiel die Niederschlagsmenge mit dem Regenschirm. Koubachi bieten mit ihren Pflanzensensoren eine Automatisierung für Hobbygärtner. Dieser kann mit den Messdaten feststellen, an was es den Pflanzen mangelt.

Einzel betrachtet ergeben Funktionen solcher Geräte nur eine spezielle Automatisierungsfunktion. Werden jedoch mehrere Geräte gemeinsam eingesetzt, lassen sich mit homee Bedingungen setzen und somit verschiedene Funktionen logisch verknüpfen. Dadurch erhöhen sich die Möglichkeiten für den Nutzer.

Bisher unterstützt homee Funkstandards wie Zwave oder EnOcean, viele Geräte basieren auf dieser Technologie. Da homee über das Internet mit der homee-App kommuniziert, ist es naheliegend, auch Geräte zu unterstützen, die über Internet-Protokolle kommunizieren. Wichtig ist, bei der Übertragung von Daten auf die Sicherheit zu achten. Viele Übertragungen über das Internet finden verschlüsselt statt. Außerdem gibt es Authentifizierungsverfahren mit Passwörtern, um zu erkennen, ob es sich um den wahren Nutzer handelt, oder Zertifikate, die eine „chain of trust“ bilden, um zu erkennen, ob der Kommunikationspartner tatsächlich der erwartete Server ist. Die Kommunikation findet in allen Fällen zwischen einem Server und dem homee-System statt. Zur Übertragung der Daten wird das bewährte Internet-Protokoll HTTP verwendet. HTTP wurde mit der Gründung des World Wide Web 1991 definiert und wird noch heute hauptsächlich in Internet-Browsern verwendet [2]. Es basiert auf TCP/IP und ist somit in die höheren Schichten des ISO/OSI-Schichtenmodells einzuordnen. Die Kommunikation über HTTP funktioniert verbindungsorientiert zwischen zwei Parteien. Jedoch wird jede Nachricht für sich selbst betrachtet. Es besteht keine Information

*Diese Arbeit wurde durchgeführt bei der Firma Codeatelier GmbH, Ostfildern

über eine Sitzung im Protokoll. Daher wird bei der Authentifizierung von der Anwendung ein Sitzungs-Token generiert. Dieses weißt den Nutzer bei jeder Übertragung aus. Die Datensätze werden mit XML oder JSON formatiert. Das Auslesen erfolgt mit einem Parser. Die

selektierten Daten werden aufbereitet und in das homee-System integriert. Der Nutzer kann so Sensorwerte einsehen und Aktoren steuern. Die Bedienung erfolgt intuitiv über Smartphone, Tablet-Computer oder PC.



Abbildung 2: Schnittstellen zwischen User und einem Smart Home Gerät

[1] Hayo Lücke, AVM – Neues von der Fritz!Box, 02.03.2010, <http://www.onlinekosten.de/news/artikel/38282/0/AVM-Neues-von-der-FritzBox>
Stand: 12.05.2015

[2] Tim BL, The original HTTP as defined in 1991, 1991, <http://www.w3.org/Protocols/HTTP/AsImplemented.html>
Stand 17.05.2015

Bildquellen:

- Abbildung 1: Jürgen Vielmeier, Homee startet Crowdfunding-Kampagne, 26.11.2013, <http://neuerdings.com/2013/11/26/homee-crowdfunding/>
Stand: 15.05.2015
- Abbildung 2: eigene Abbildung

Konzeption und prototypische Umsetzung einer Anwendung zur Unterstützung von Websitebesuchern auf Basis von browsereigenen Technologien

Stefan Lutz*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Die Menschen verrichten heutzutage immer mehr ihrer alltäglichen Aufgaben online. Seien es Einkäufe, das Buchen des Urlaubs, die Wahl einer neuen Versicherung oder eines Strom-anbieters. Doch viele vermissen dabei etwas, dass in einem Fachgeschäft ganz normal ist: die Beratung.

Einer repräsentative Umfrage von Bit-kom Research innerhalb der Bundesrepublik Deutschland zufolge, fand gut jeder achte Besucher eines Online-Shops (13 Prozent) keine Beratungsmöglichkeit, als er sie brauchte. Jeder Fünfte (21 Prozent) hat schon einmal einen Berater kontaktiert und insgesamt wollte schon jeder dritte Besucher (13 Prozent) einen Berater kontaktieren [1].

Auch immer mehr professionelle Software-lösungen, werden mittlerweile als Webanwendungen realisiert.

Die Firma TeamViewer entwickelt eine Softwareanwendung zur Fernwartung verschiedener Computersysteme und mobiler Geräte. Um den wachsenden Markt der Webanwendungen bedienen zu können, hat TeamViewer folgendes Ziel für die vorliegende Abschlussarbeit gesetzt: es soll ein Werkzeug zur Fernwartung von Web-Anwendungen entwickelt werden, das ohne zusätzliche Installation auskommt.

Der Vorteil für den Kunden ist einfach: Ein Kunde hat ein Problem mit der Benutzung einer browserbasierten Anwendung und ein Mitarbeiter des Herstellers dieser Anwendung möchte dem Kunden assistieren. Dazu muss der Kunde z.B. auf eine Schaltfläche in der Webanwendung klicken. Schließlich wird eine Verbindung zu einem Supporter / Berater hergestellt. Dieser sieht den Browserinhalt des Kunden und kann entsprechende Hilfestellung geben indem er dabei hilft bestimmte Bereiche auf der Webseite zu finden und den Kunden bei der Bedienung oder dem Ausfüllen von Formularen unterstützt.

Diese Art der Zusammenarbeit auf einer Webseite wird auch als Co-Browsing bezeichnet.

Im Rahmen der Abschlussarbeit wurden

zunächst Technologien zum kontinuierlichen Abgreifen von Browserinhalten evaluiert. Der Fokus lag darauf, dass ein möglichst identisches Bild vom aktuell sichtbaren Bereich des Browsers abgegriffen und übertragen wird. Allerdings gibt es hier viele Faktoren, die es erschweren ein genaues Abbild zu reproduzieren: die Schwierigkeiten reichen von der unterschiedlichen Darstellung bei unterschiedlichen Browsern, bis hin zu Browser-Plugins wie etwa Adblocker, welche das Bild der Seite verändern.

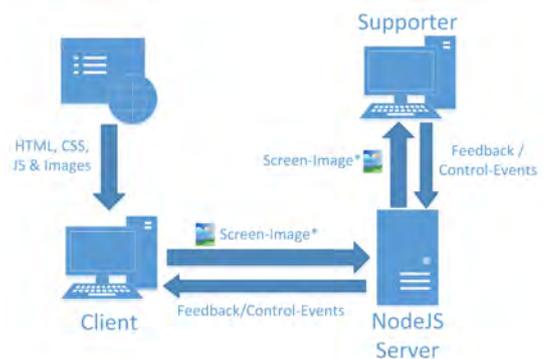


Abbildung 1: Grundkonzept

Bei der Wahl der Technologie zum Abgreifen des Bildschirminhaltes wurde auch der Anwendungsfall berücksichtigt, dass es viele Web-basierte Anwendungen gibt, welche in lokalen Netzwerken eingesetzt werden, wo die Ressourcen der Webseite (Bilder, Stylesheets) nicht öffentlich verfügbar sind. Bedacht werden musste außerdem, dass der Kunde auf der Seite navigiert und der Browser so die Übertragung unterbricht, um eine neue Seite zu laden. Für diese Problematik war es wichtig, eine möglichst transparente Lösung zu finden, so dass der Endanwender die Applikation wie gewohnt bedienen kann.

Es wurde auch sehr großen Wert darauf gelegt, dass das System mobile Browser unterstützt. Diesbezüglich tritt ein weiterer großer Vorteil des Co-Browsing zutage: die Unabhängigkeit zu den Betriebssystemen. Berücksichtigt werden müssen „nur“ die Browser. Ansonsten kann, bei korrekter Umsetzung, ei-

*Diese Arbeit wurde durchgeführt bei der Firma TeamViewer GmbH, Göppingen

ne Person mit ihrem Tablet oder Smartphone genauso gut unterstützt werden wie jemand an seinem Desktop PC. Was umso wichtiger

ist, wenn man sich aktuelle Statistiken ansieht, wonach es mittlerweile mehr mobile als Desktop Nutzer gibt [2].

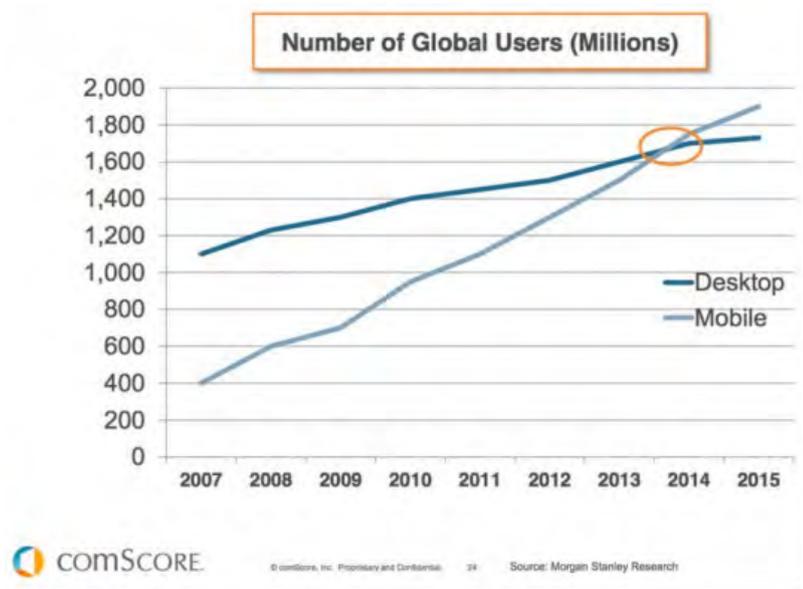


Abbildung 2: Vergleich der Anzahl an mobilen und Desktop Nutzern

Ein wichtiger Teil der Arbeit war es auch, die Grenzen der Technologie zu identifizieren und Alternativlösungen zu dokumentieren. Dies betrifft auch die Simulation von Nutzereingaben durch die Maus bzw. Touch oder die Tastatur. Die Events, die bei einer Eingabe ausgelöst werden, können zwar in JavaScript auch manuell ausgelöst werden, dennoch gibt es immer noch erhebliche Unterschiede zwischen einem Simulierten Mausklick / Tastenanschlag zu einem „echten“ durch den Nutzer.

Als serverseitige Technologie kam NodeJS zum Einsatz und für die Datenübertragung Secure Websockets. NodeJS eignet sich hier für die prototypische Umsetzung ganz besonders, da auf Server und Client dieselbe Sprache und

z.T. dieselben Bibliotheken eingesetzt werden konnten.

Für die Sitzungsverwaltung kam die Team-Viewer REST-API für spontanen Support zum Einsatz.

Eine große Herausforderung auf der Seite des Clients war es vor allem, dass das Umfeld im Gegensatz zur „normalen“ Webentwicklung gänzlich unbekannt ist. Schließlich soll das Tool in beliebige Webseiten integriert werden können, die sich alle unterschiedlicher Frameworks bedienen können.

Am Ende der Arbeit kam ein durchaus erfolgreicher Prototyp zustande, der durch Skript-Injektion auf vielen Seiten getestet werden konnte.

[1] BITKOM – Online-Shopper erwarten Kundenberatung; http://www.bitkom.org/de/themen/54866_79898.aspx abgerufen

[2] Smart Insights Ltd. – Statistics on mobile usage and adoption to inform your mobile marketing strategy <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>

Bildquellen:

- Abbildung 1: Smart Insights Ltd. <http://www.smartinsights.com/wp-content/uploads/2014/03/Mobile-stats-vs-desktop-users-global-550x405.png>
- Abbildung 2: Eigene Darstellung

Konzept und Entwicklung einer mobilen App für die Verwaltung von heterogenen Elektro-Fuhrparkflotten

Edouard Maihöfer*, Manfred Dausmann, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Aufgrund der Treibhausgasemissionen sucht die Automobilindustrie seit geraumer Zeit nach Alternativen zum Verbrennungsmotor. Dabei stellte sich heraus, dass ein Elektromotor in verschiedenen Eigenschaften einem Verbrennungsmotor überlegen ist [1][2]. Um die Eigenschaften eines Elektroantriebs richtig auszunutzen zu können, werden intelligente Managementsysteme benötigt, welche einem Fahrer jederzeit und in Echtzeit den Ladezustand sowie die Entfernungen zum nächsten Ladepunkt oder Vorschläge für eine sinnvolle Route übermitteln. Deren Komplexitätsgrad erhöht sich noch, wenn Mobilitätsressourcen oder Reiseketten geteilt werden sollen [3].



Abbildung 1: Abschließende Übersicht der Buchung

Da diese Art von Beförderungsmittel noch große Skepsis und Vorbehalte bei den Endverbrauchern herbeiruft (vgl. [4]), ist es umso wichtiger ihnen eine Möglichkeit zu geben, dies in einer vertrauten Umgebung auszuprobieren und testen zu können. In einem ersten Schritt soll das Reservieren, Informieren und

Überwachen eines Elektrofahrzeuges von einem mobilen Endgerät aus ermöglicht werden.

Zu diesem Zweck soll im Rahmen der Bachelorarbeit eine App entwickelt werden, mit deren Hilfe ein Benutzer eine Buchung von Fahrzeugen einer heterogenen Elektro-Fuhrparkflotte durchführen kann und damit gewisse Berührungspunkte überwinden kann.

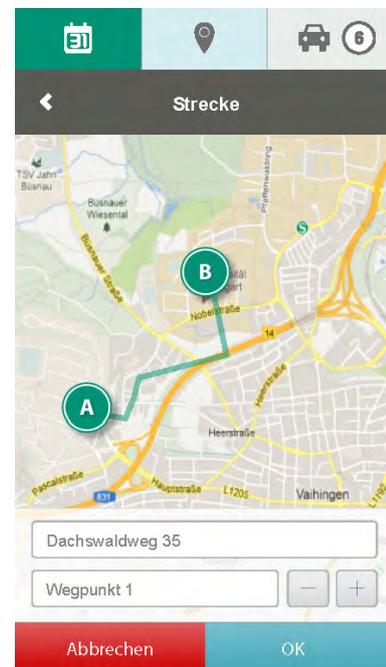


Abbildung 2: Übersicht der gebuchten Strecke über Google Maps

Die wichtigsten Anforderungen hierbei sind das Look & Feel und ein Responsive Webdesign. Die App soll dem Benutzer eine intuitive und wiedererkennende GUI anbieten, welche er begeistert benützt. Mit der App soll es möglich sein, einen kompletten Buchungsvorgang abzuschließen angefangen mit dem Log-In Fenster bis hin zur Buchungsübersicht. Weiterhin soll die gebuchte Strecke mithilfe von Google Maps angezeigt werden können und es soll eine Gamification eingebaut werden, um den Endbenutzer zu einem ökologischen Selbstbewusstsein zu ermutigen. Abbildungen 1–3 zeigen die ersten Entwürfe der Be-

*Diese Arbeit wurde durchgeführt bei der Firma Fraunhofer Anwendungszentrum KEIM, Esslingen

nutzeroberfläche für diese App. Eine weitere wichtige Anforderung ist, dass die App plattformunabhängig sein soll. Normalerweise erstellt man eine App mit dem vom Hersteller des jeweiligen Betriebssystems, angebotenen Software Development Kit. Dies hat zur Folge, dass man für jede Art von Betriebssystem die App neu schreiben muss. Dadurch entstehen jedoch große Kosten sowie ein erhöhter Arbeitsaufwand. Es gibt bereits eine Reihe von Frameworks, welche es ermöglichen plattformunabhängige Apps zu erstellen. Jedoch besitzen diese Frameworks auch Nachteile: Die mit ihnen entwickelten Apps besitzen in der Regel nicht das Look & Feel einer nativen App und leiden oft an Performance und anderen Problemen.

In dieser Bachelorarbeit soll untersucht werden, ob und wie es möglich ist, kostengünstig eine App mit Hilfe der genannten Frameworks zu entwickeln. Um einen Überblick zu bekommen und ein für den Anwendungsfall passendes Framework zu finden, sollen zuerst mittels des jeweiligen Frameworks Prototypen erstellt werden. Dabei zeigen sich die Stärken und Schwächen, aufgrund deren ein Framework ausgewählt werden kann. Im Anschluss erfolgt dann mit dem ausgewählten Framework die Entwicklung des finalen Prototyps.

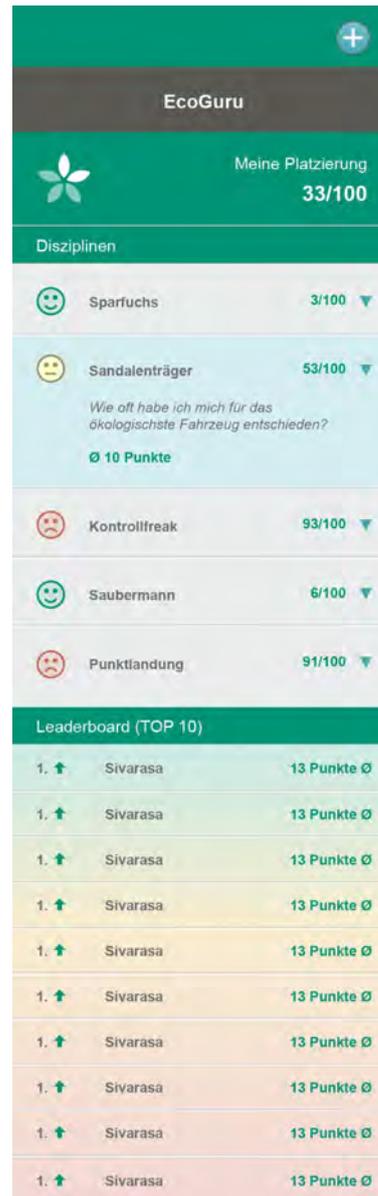


Abbildung 3: Ansicht des aktuellen Punktestandes (Gamification)

[1] <http://de.wikipedia.org/wiki/Elektroauto> Abgerufen am 18 Mai 2015

[2] <http://www.bem-ev.de/warum-die-elektromobilitat-sich-durchsetzen-wird/> Abgerufen am 18 Mai 2015

[3] <http://www.keim.iao.fraunhofer.de/de/verbundprojekte/living-lab-efleet.html> Abgerufen am 18 Mai 2015

[4] <http://www.zeit.de/auto/2011-04/elektroauto-akzeptanz> Abgerufen am 18 Mai 2015

Bildquellen:

- Abbildung 1–3: Fraunhofer Anwendungszentrum KEIM

Vergleich moderner Authentisierungsmethoden und deren Einsatzmöglichkeiten

Lukas Martin*, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Die Notwendigkeit von Methoden zur sicheren Kommunikation ist allgegenwärtig. Spionageaffären und gestohlene Nutzerdaten sind in aller Munde. Angreifer versuchen mit den unterschiedlichsten Mitteln, an Daten zu kommen. Ein gutes Beispiel dafür ist das Phishing [1]. Ein Betrüger schleust beim Benutzer eine gefälschte, vom Äußeren aber kaum von der echten zu unterscheidende, Internetseite ein. Wie gewohnt gibt der Benutzer seinen Benutzernamen und sein Passwort ein. Und damit hat der Angreifer auch schon alles, was er benötigt. Die Antwort auf die Frage, wie man sich vor solchen Angriffen schützen kann, lautet Authentisierung. Diese bildet, neben einer Verschlüsselung, die Grundlage für eine sichere Kommunikation mit einem entfernten System. Damit weist sich ein Nutzer gegenüber einem System aus, beweist also seine Identität. Genauso muss auch ein System dem Nutzer beweisen können: „Ich bin es wirklich“.

Es geht also darum Sicherheit zu schaffen, dass das Gegenüber dasjenige ist, für den es sich ausgibt. Dazu bestehen nach [2] folgende grundsätzliche Möglichkeiten:

- Etwas wissen – Passwort oder Pin
- Etwas besitzen – Schlüssel oder Chipkarte
- persönliche Eigenschaften – Fingerabdruck, Handvenenabdruck, Stimme

Darauf aufbauend haben sich verschiedene Systeme zur Authentisierung am Markt etabliert. All diese Verfahren haben aber auch gemeinsame Probleme. Dazu soll ein gewöhnlicher Authentisierungsvorgang im Detail beleuchtet werden. Ein einfaches Beispiel für einen solchen Vorgang ist die Anmeldung auf einer Webseite vom heimischen PC aus (siehe Abbildung 1).

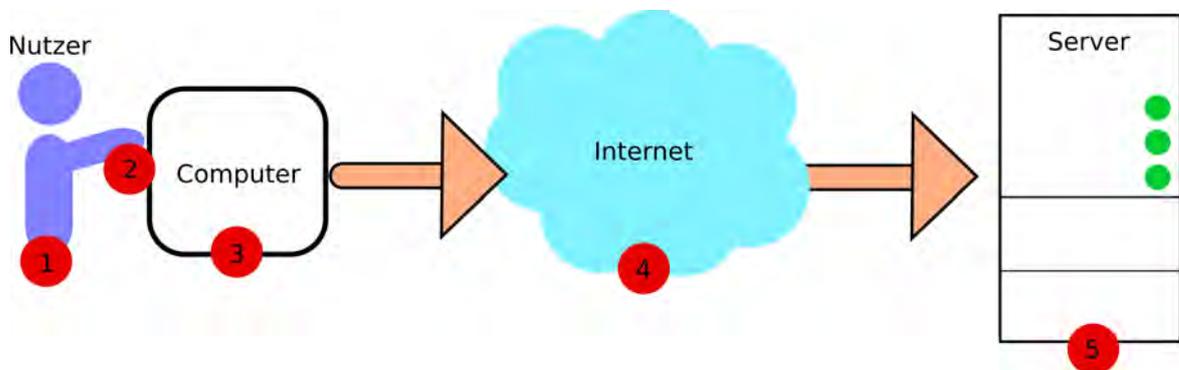


Abbildung 1: Informationsfluss bei einer Authentisierung

Der Nutzer (1) selbst stellt durch sein Verhalten ein Sicherheitsrisiko dar. Er kann durch sein Verhalten dazu beitragen, ein Sicherheitssystem unbrauchbar zu machen, beispielsweise indem er ein Passwort notiert und die Notiz offen lagert. Viele Nutzer gehen mit ihren Passwörtern nicht sorgfältig genug um. So werden beispielsweise identische Passwörter für verschiedene authentisierende Systeme verwendet. Gelangt das Passwort in die falschen Hände, sind damit alle weiteren Anwendungen ebenfalls in Gefahr. Durch Keylogger in

hard- oder softwareform lassen sich Benutzerkennungen und Passwörter leicht extrahieren (2). Viren und Trojaner suchen auf dem Computer (3) nach verwertbaren Informationen. Im weiten Internet (4) kann bei unzureichender Verschlüsselung jeder mitlesen. Viele Angriffe gelten aber auch den Zielsystemen selbst (5). Da diese aber meistens besonders geschützt sind, versuchen sich die Angreifer eher an einem der früheren Angriffspunkte. Genau hier setzt der Schutz durch Authentisierungsverfahren an. Statische Passwörter sind rela-

*Diese Arbeit wurde durchgeführt bei der Firma Fujitsu TDS GmbH, Neckarsulm

tiv schnell zu knacken. Oftmals sind die Passwörter zu einfach und können mittels Brute-Force-Attacken, also das Passwort in sehr vielen Versuchen zu erraten, ermittelt werden.

Gefragt sind dynamische Passwörter, deren Gültigkeit begrenzt ist. Somit wird es dem Angreifer extrem erschwert, Zugang zu Systemen zu erlangen. Solche dynamischen Verfahren bringen aber auch eine komplexere Umgebung und aufwändigeren Betrieb mit sich.

In dieser Abschlussarbeit werden konkret folgende Verfahren miteinander verglichen: Das klassische Passwortverfahren, verschiedene OTP-Verfahren (One Time Password) wie TAN, iTAN und smsTAN sowie zeitbasierte TAN-Verfahren wie RSA Tokens (siehe Abbildung 2).



Abbildung 2: SecurID RSA-Token

Immer mehr ziehen die biometrischen Verfahren in unseren Alltag ein [3]. Deshalb wurden auch diese in den Vergleich mit aufgenommen – Fingerabdruck-, Handvenen- sowie Irisabbildscanner. Weitere spezielle Verfahren sind die zertifikatsbasierte Authentisierung, anonyme Authentisierung sowie Zero-Knowledge-Verfahren und räumliche, GPS-gestützte Verfahren. All diese Verfahren haben ihre Vor- und Nachteile, welche in dieser Arbeit ermittelt, verglichen und nach verschiedenen Gesichtspunkten gewichtet werden. Der Vergleich umfasst an erster Stelle die Sicherheit,

aber auch den Implementierungsaufwand, der administrative Aufwand sowie Kosten und Benutzerakzeptanz.

Durch eine Umfrage zur Nutzung von Passwörtern und Pins konnten verschiedene sicherheitsrelevante Problemstellen gefunden und daraus Handlungsempfehlungen abgeleitet werden. Beispielsweise vertrauen viele Nutzer ihre Passwörter einem Passwortmanagementsystem an. Dadurch lassen sich beliebig lange und vor allem komplexe Passwörter sicher erstellen und verwahren, ohne das es nötig ist, verschiedenste kryptische Zeichenkombinationen auswendig zu lernen. Somit werden Wörterbuchattacken praktisch ausgeschlossen. Die Handhabung ist sehr einfach, was der Benutzerakzeptanz enorm entgegenkommt.

Als weitere Erkenntnis konnte in der Ortsabhängigkeit eines Authentisierungssystems ein großer Sicherheitsgewinn erkannt werden. Gebäudesicherung findet beispielsweise in der Regel immer am selben Ort statt, der regelmäßig Zutrittsberechtigte Personenkreis ändert sich in überschaubarem Umfang. Somit kann ein gehärtetes, entkoppeltes System verwendet werden. Fingerabdrucksensoren können über Gebäudeleittechnik miteinander verbunden werden, eine Schnittstelle nach außen ist nicht notwendig. Die gespeicherten Daten verlassen das Gebäude nie und es gibt keine Verbindung zu anderen Systemen. Damit ist ein Angriff auf dieses System zumindest extrem erschwert.

In der Arbeit wird auch die Kombination zweier Authentisierungsverfahren näher beleuchtet werden. Die sogenannte Zwei-Faktor-Authentisierung offeriert vor allem in Hinblick auf die finanziellen Aspekte einige gute Möglichkeiten.

- [1] BSI, Webseite, https://www.bsi-fuer-buerger.de/BSIFB/DE/GefahrenImNetz/Phishing/phishing_node.html,
Aufgerufen am 12. Mai 2015
- [2] Stephan Spitz / Michael Pramateftakis / Joachim Swoboda: Kryptographie und IT-Sicherheit. 2., überarbeitete Auflage Vieweg + Teubner Verlag, 2011
- [3] Fingerprint vs. Venenscan vom 01.08.2012, Webseite, <http://www.sicherheits-berater.de/startseite/artikel-ohne-abo/biometrische-verfahren-fingerprint-vs-venenscan.html>,
Aufgerufen am 12. Mai 2015

Bildquellen:

- Abbildung 1,2: Lukas Martin

Verifikation von Mustererkennungsverfahren in der industriellen Bildverarbeitung am Beispiel einer Labelprüfung

Steffen Mergheim*, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Der Produktbereich Montageanlagen und Sondermaschinen der Robert Bosch GmbH entwickelt und fertigt Prüfanlagen für die Bosch-Werke weltweit. Sie werden zur optischen Inspektion unterschiedlicher Produkte genutzt. Steuergeräte, wie sie im Kfz-Bereich eingesetzt werden, zählen ebenfalls zu den Produkten. In Abbildung 1 ist ein solches Steuergerät dargestellt.



Abbildung 1: Steuergerät aus dem Kfz-Bereich

Diese Steuergeräte enthalten zur grundsätzlichen Kennzeichnung Produktlabels. Diese Labels enthalten desweiteren aber auch Informationen über die eingebaute Variante und der vorhandenen Softwarestände. Somit gilt für die Labels, dass nicht nur ein unbeeinträchtigtes Erscheinungsbild vorliegt, sondern auch die in Klarschrift aufgebrauchten Inhalte müssen auf Lesbarkeit und Inhalt geprüft werden. Bei diesen Inhalten handelt es sich um Logos, Textfelder unterschiedlicher Größe und unterschiedlicher Schriftarten, sowie Barcodes und Data-Matrix-Codes, wie in Abbildung 2 zu sehen.

Bisherige Ansätze erfordern das trainieren eines Mustertemplates für jede Labelvariante. Dazu erfolgt manuell eine Zuweisung von Prüfregionen für die einzelnen Label-Bestandteile um der Bildbearbeitungssoftware mitzuteilen um welche Art von Inhalt es sich handelt und welche maximalen Abweichungen vorliegen dürfen. Die Position und die Größe der

Logos und der Datamatrixcodes sind immer gleich, der Barcode hingegen kann in seiner Breite variieren und die Textfelder können für jeden Druck einen komplett anderen Inhalt enthalten.



Abbildung 2: Produktlabel eines Steuergeräts

Anhand des Wissens über die Prüfregion und dessen Inhaltsmerkmale ist eine Segmentierung durch Schwarz-Weiß-Flächen möglich [1]. Anschließend wird diese durch unterschiedliche Algorithmen untersucht und anhand von bekannten Trainingsdaten klassifiziert. Ist nach der Erkennungsphase der Inhalt ermittelt, so muss nun in einem zweiten Schritt diese Erkennung verifiziert werden. Also in welchem Ausmaß kann es sich auch tatsächlich um den gelesenen Inhalt handeln. Hier wird vor allem die Druckqualität überprüft, denn sie ist in der Regel der Grund für eine abweichende Klassifizierung. So ist es sehr schwierig ähnliche Ziffern voneinander zu trennen. B und 8 unterscheiden sich nur in sehr geringer Pixelfläche und 0 ist mit O häufig ununterscheidbar. Abbildung 3 zeigt die Unterscheidung zwischen der Klassifizierung und der Verifikation anhand von Ziffern.

Das Ziel dieser Arbeit ist es mehrere Fehlerquellen zu untersuchen um eine stabilere Aussage hinsichtlich der Richtigkeit der Klarschrift festzulegen. Neben Inhaltsfehlern werden auch Aufbringfehler wie Positionsfehler und Oberflächenkratzer inklusive Druckerfehler wie defekte Pixelreihen, ungleichmäßige Druckerschwärze und weitere Unstimmigkeiten als Fehlerquellen überprüft. Außerdem ist die Festlegung der Aussage, ab wann ein Prüfteil gut bzw. schlecht eingeordnet wird zu veri-

*Diese Arbeit wurde durchgeführt bei der Firma Robert Bosch GmbH, Stuttgart-Feuerbach

fizieren, da hier selbstverständlich eine Grauzone vorliegt. Dazu wird die Grauzone in mehrere Fehlerklassen eingeteilt.

Um die Grundlagen der Klassifizierung zu verstehen, wird zu dem auf die Algorithmen der unterschiedlichen Mustererkennungsverfahren eingegangen. Für die Codeerkennung liegen sowohl für Barcode, als auch für Datamatrixcode standardisierte Normen vor. Die Objekterkennung wird anhand von Template Matching sowie Shape-based Matching erläutert. Für die Texterkennung sind die Verfahren für den k-Nächsten-Nachbarn (kNN), den Support-Vektor-Klassifikator (SVM) und die künstlichen Neuronale Netze (ANN) beschrieben [2].

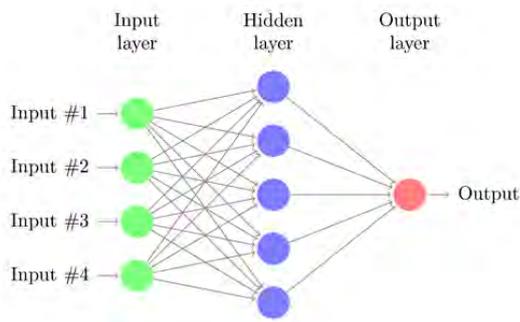


Abbildung 3: Simple Darstellung eines künstlichen Neuronales Netzes (ANN)

In Abbildung 4 ist die Funktionsweise eines einfachen Neuronales Netzes dargestellt. Dabei werden anhand bestimmter Merkmale eines Musters Entscheidungswerte trainiert. Solche Merkmale können beispielsweise die horizontale oder vertikale Position eines Objekts sein oder dessen Breite bzw. Höhe, aber auch die absolute Anzahl an Pixeln oder deren Durchschnittsgrauwert. Die Entscheidungswerte des Musters können so zur Klassifizierung beliebiger Eingangsobjekte verwendet werden. Die Entscheidungswerte werden dazu mit den unterschiedlichen Merkmalswerten Input #1 bis Input #N (Input layer) des Eingangsobjektes multipliziert. Durch diese berechneten Werte wird das Netz gewählt, dass die größte Übereinstimmungswahrscheinlichkeit aufweist und es wird die zugehörige Klassifizierung in der versteckten Schicht (Hidden layer) für den Output (Output layer) verwendet. Dabei ist zu erwähnen, dass ein Neuronales Netz nicht nur aus einer versteckten Schicht bestehen muss, sondern auch beliebig viele versteckte Schichten enthalten kann. Der Vorteil davon ist, dass noch mehr Einflüsse auf die Klassifizierung erfolgen können, allerdings wird das Netz auch mit jeder Schicht komplexer.

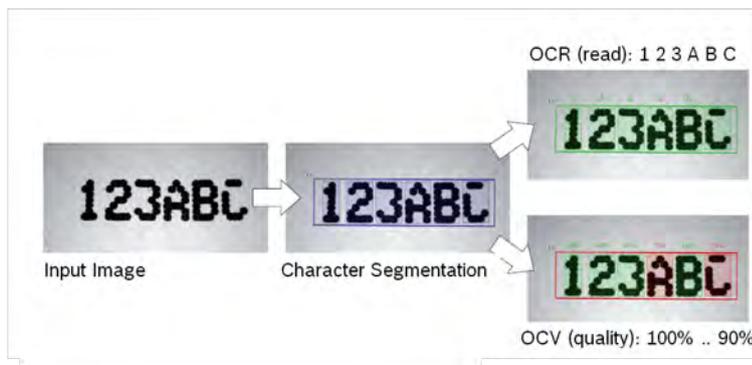


Abbildung 4: Unterscheidung zwischen Klassifizierung und Verifikation von Ziffern

[1] Christian Demant, Bernd Streicher-Abel, and Axel Springhoff. Industrielle Bildverarbeitung – Wie optische Qualitätskontrolle wirklich funktioniert. 3. Auflage. Springer Berlin Heidelberg, 2011.

[2] Oliver Kramer. Dimensionality Reduction with Unsupervised Nearest Neighbors, volume 51 of Intelligent Systems Reference Library. Springer Berlin Heidelberg, 2013.

Bildquellen:

- Abbildung 1 – 2: Robert Bosch GmbH
- Abbildung 3: <http://www.texample.net/media/tikz/examples/PNG/neural-network.png>
- Abbildung 4: http://www.baumer.com/fileadmin/user_upload/international/Downloads/TR/Baumer_VT-OCR-vs-OCV_TR_DE_1105.pdf

Evaluierung eines Automotive Testsystems (HiL) mit Komponenten aus der Automatisierungstechnik am Beispiel eines Schiebedachs

Thomas Müller*, Werner Zimmermann, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Die steigende Komplexität von Steuergeräten und anderen elektronischen Geräten zieht eine höhere Fehleranfälligkeit mit sich. Um dem entgegen zu wirken benötigt man eine Lösung um elektronische Geräte auf Funktionalität testen zu können. Steuergeräte können aber oftmals nicht direkt im Fahrzeug getestet werden. Fehler und Zustände des Steuergeräts können somit nicht jederzeit reproduziert werden. Seit einigen Jahren steigt daher die Anfrage an Methoden des HiL-Testens, bei denen reale Geräte in eine Simulationsumgebung getestet werden (HiL, Hardware in the Loop). Zukünftig wird der Bedarf mit vermehrten sicherheitskritischen Anwendungen in Fahrzeugen weiter steigen. Autonom fahrende Fahrzeuge stehen hierbei im Fokus.

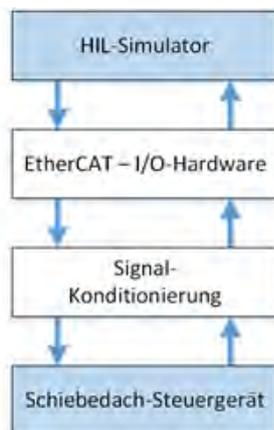


Abbildung 1: Vereinfachter Prüfaufbau

Werden Steuergeräte einzeln getestet, spricht man von Komponenten-HiL. Diese Komponenten-HiL werden an alle Schnittstellen des elektronischen Geräts angeschlossen und simulieren die komplette Umwelt, in welcher sich das Gerät im laufenden Betrieb befindet. Am Beispiel des in dieser Arbeit verwendeten Schiebedachs einer Fahrzeugbauerei simuliert der Simulator mit Hilfe digitaler Ein- & Ausgänge sowie des *Controller Area Networks (CAN)* die Umgebung des Steuergeräts. Somit muss kein reales Schiebedach mit zugehöriger Aktorik und Sensorik an das

Steuergerät angeschlossen werden um es auf Funktionsfähigkeit prüfen zu können. Zukünftig sollen HiL-Simulatoren mit kostengünstiger Hardware produziert und entwickelt werden. Dabei kommt das Industriefeldbussystem EtherCAT zum Einsatz. Die dazugehörige Hardware von Industriefeldbussystemen wird in hoher Stückzahl gefertigt und ist dementsprechend günstig im Preis. EtherCAT ist sehr gut skalierbar und vereinfacht somit die Hardware-Anpassungen für HiL-Simulatoren. Für das Schiebedach-Steuergerät soll nun die passende EtherCAT-Hardware ausgewählt und evaluiert werden. Ein wichtiger Punkt ist unter anderem das Echtzeitverhalten des Simulators mit der EtherCAT-Schnittstelle. Grundsätzlich wird EtherCAT in der Industrie eingesetzt, weshalb Signalkonditionierungen zu Automotive-Schnittstellen nötig sind. Abbildung 1 zeigt einen vereinfachten Prüfaufbau.



Abbildung 2: EtherCAT-Klemmen

EtherCAT ist echtzeitfähig und setzt vollständig auf das TCP/IP-Protokoll. Hierzu werden Daten von EtherCAT im Daten-Frame eines üblichen Ethernet-Frames zyklisch versendet. EtherCAT besitzt ein eigenes Typfeld im Ethernet-Frame. Die Koppellemente des Feldbussystems werden an handelsübliche Netzwerkkarten angeschlossen und arbeiten im Voll-Duplex Modus und Fast-Ethernet (100 Mbit/s). EtherCAT ist beliebig erweiterbar, von einer einzelnen I/O-Klemme mit einfachster Topologie bis hin zu mehreren Tausend I/Os und komplexer Baum-Struktur mit Redundanz

*Diese Arbeit wurde durchgeführt bei der Firma Berner & Mattner Systemtechnik GmbH, S-Vaihingen

Entwicklung eines REST-Webservice zur Ortdatensuche im Umfeld von Einsatzleitsystemen mit C#/ASP.NET-Web-API

Raphael Mußnug*, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Im Bereich der Einsatzkoordination durch Leitstellen von Rettungsdiensten und Feuerwehren kommt es in der Regel auf jede Sekunde an. Deshalb ist es im Moment eines Notfalleinsatzes wichtig, dass die Informationen über Ort und Stelle des Einsatzziels möglichst schnell im System aufgenommen und zur weiteren Alarmierung verarbeitet werden können. Dafür ist neben einem performanten Algorithmus zum Finden der Adresse auch eine hierarchisch geordnete Datenstruktur notwendig. Dies bedeutet zum Beispiel, dass Orte Regionen sein können, die wiederum Orte enthalten und diese meistens in Ortsteile untergliedert sind. Zudem umfassen Orte generell Straßen. Eine Straße selbst kann je nach Untergliederung und Länge zu unterschiedlichen Orten gehören. Innerhalb dieser Struktur gibt es zusätzlich noch Objekte, die spezielle Gebäude von Aldi bis Zollamt darstellen können.

Des Weiteren soll die Suche nach Ortsdaten von unterschiedlichen Anwendungen verwendet werden: Zum einen von kleineren Webanwendungen, die in umliegenden externen Wachen benutzt werden, aber auch von komplexen Desktopanwendungen, welche auf den zentraleren Leitstellen zum Einsatz kommen. Deshalb bietet sich innerhalb eines verteilten Systems an, die Suche über einen zentralen Webservice zur Verfügung zu stellen. Dieser wird im Intranet gehostet und kann über HTTP-Requests im Sinne von REST (Representational State Transfer) angesteuert werden. In Abbildung 1 ist der vereinfachte prinzipielle Aufbau eines ASP.NET-Web-API-Service dargestellt. Bei jedem HTTP-Request werden eine Reihe von Prozeduren durchlaufen. Im Message Handler, welcher bei jeder Anfrage aufgerufen wird, können allgemeine Dinge wie eine Sicherheits-Tokenvalidierung durchgeführt werden. Wenn die Messagehandler durchlaufen wurden, können optionale Action- und Attributfilter implementiert werden. Diese sind in der Form [Attributfiltername] im Controller einzufügen und behandeln in der Regel eine Aussage, durch die der Request abgebrochen oder weiter durchgeführt wird.

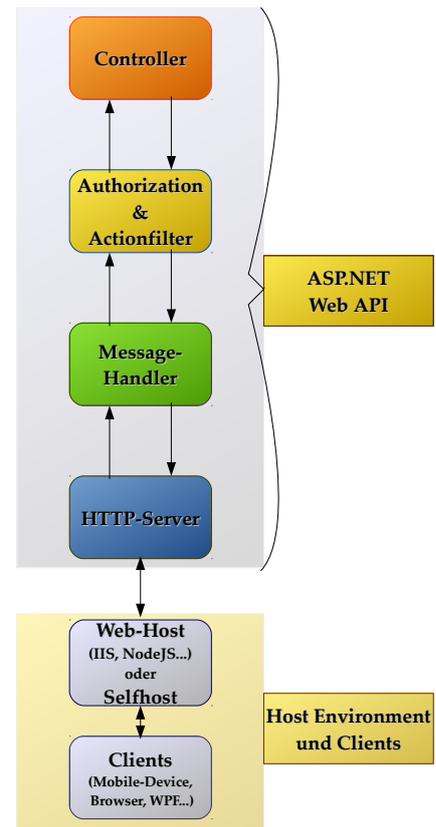


Abbildung 1: Web API Übersicht

Wenn diese Pipe durchlaufen ist, wird im Controller die eigentliche HTTP-Methode aufgerufen. Diese Methoden umfassen die Standardmethoden wie `get()`, `get(Id)`, `push()`, `put(Id)` und `delete(Id)`. Für jede Entität, im Falle dieser Arbeit hauptsächlich Ort, Straße oder Objekt, gibt es genau einen Controller. Zur eindeutigen Ansteuerung dessen wird über eine einmalige URI eine Route definiert. Diese ist in der Datei `WebAPIConfig.cs` standardmäßig wie folgt implementiert (siehe Abbildung 2): „routeTemplate: `api/{controller}/{id}`“. Je nach Domain und Netzwerkkonfiguration lau-

*Diese Arbeit wurde durchgeführt bei der Firma Swissphone Systems GmbH, Leinfelden-Echterdingen

tet die vollständige URI: `http://{IP-Adresse-Server}.{Domaine}:{Port}/api/{controller}/{id}`.

Um Suchbegriffe oder Teile davon übertragen zu können, kann die URI um Querystrings erweitert oder in einfachen Fällen über selbst definierte Header im Request übertragen werden. Dies führt zu einem weiteren Fokus dieser Abschlussarbeit: dem Suchalgorithmus. Nachdem der Controller erreicht wurde, wird der Request weiter verarbeitet und führt in der Regel zu einem Repository. In diesem wird die Verbindung zur Datenbank hergestellt und über SQL-Befehle oder PL-SQL-Prozeduren die Daten je nach Suchstring gefiltert.

```
using System.Web.Http;
namespace WebApplication1
{
    public class WebApiApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            GlobalConfiguration.Configure(config =>
            {
                config.MapHttpAttributeRoutes();

                config.Routes.MapHttpRoute(
                    name: "DefaultApi",
                    routeTemplate: "api/{controller}/{id}",
                    defaults: new { id = RouteParameter.Optional }
                );
            });
        }
    }
}
```

Abbildung 2: Web API Routen-Konfiguration

Da es sich bei Adressdaten in den meisten Fällen um Eigennamen handelt, welche unterschiedlich geschrieben und ausgesprochen werden können, ist es für die Suchanfrage wichtig, den einfachen Vergleich von Zeichenketten um eine Ähnlichkeitssuche zu erweitern. Sonderzeichen, gleich klingende Worte, sowie kleinere Schreibfehler oder die Transposition von Buchstaben sollen kein Hindernis darstellen (Eine Transposition sind vertauschte Buchstaben innerhalb eines Wortes.). Um dieser Anforderung gerecht zu werden, gibt es zwei zentrale Ansätze, welche im folgenden kurz umrissen werden. Beim phone-

tischen Ansatz wird versucht, ähnlich klingende Worte zu finden, indem entweder nach bestimmten Regeln einzelnen Buchstaben Zahlenwerte zugeordnet oder ganze Worte modifiziert werden. Problematisch hierbei ist die Abhängigkeit von der jeweiligen Sprache, da die Regeln dieser angepasst sein müssen. Ein französisch ausgesprochener Name wird im Deutschen anders ausgesprochen. Der zweite Ansatz ist allgemeiner und umfasst alle Ähnlichkeiten. Begriffe sind hier die Hamming-Distanz oder die Levenstein-Distanz. Erste behandelt direkte Unterschiede und trifft bei einem Vergleich einzelner Zeichen zweier Zeichenketten eine Aussage über die Anzahl der verschiedenen Zeichen. Dies kann jedoch sehr schnell ungenau werden. Die Levenstein-Distanz ermittelt die Anzahl der Operationen, welche nötig wären, um ein Wort in ein anderes umzuwandeln. Für die meisten dieser Grundprobleme gibt es Standardimplementierungen von Seiten der Datenbankhersteller. Um diese jedoch sinnvoll nutzen zu können, ist es hilfreich zu wissen, wie diese Algorithmen im Grunde funktionieren. [1]

Zum Testen und Darstellen der Ergebnisse wurde im Rahmen dieser Arbeit eine Webseite erstellt. Innerhalb dieser werden in einer Formularansicht mithilfe von Autocompletion-Input-Feldern asynchron je nach eingegebenem Begriff entsprechende Suchergebnisse angezeigt, bis das gewünschte ausgewählt wird. Bei Auswahl wird dann die komplette Datenstruktur per `get(Id)`-Methode vom Server nachgeladen. Auf der Testoberfläche wird zu jedem Request die benötigte Zeit und die Anzahl der gefundenen Datenobjekte angezeigt. Dies ermöglicht, eine Aussage über die Performanz treffen zu können. Ein weiterer Testfall betrifft die reine Suchbegriffsnutzung. Es werden alle gefundenen Daten in einer Tabelle visualisiert. Dadurch kann der Suchalgorithmus nach seiner Ergebnisqualität bewertet werden. Dies ist vor allem wichtig, um den Ähnlichkeitsalgorithmus entsprechend eichen zu können.

[1] Wilz, Martin: Aspekte der Kodierung phonetischer Ähnlichkeiten in deutschen Eigennamen, Universität Köln

Bildquellen:

- Abbildung 1: Eigene Darstellung vereinfacht nach <http://www.asp.net/posters/web-api/ASP.NET-Web-API-Poster.pdf>.
- Abbildung 2: <http://www.asp.net/web-api/overview/advanced/configuring-aspnet-web-api>

Konzeption und Implementierung einer Testumgebung für den Dauertest von Telematikeinheiten, mit Simulation realistischer Umgebungsbedingungen durch einen Zustandsautomaten, der gültige Signale für CAN- und serielle Schnittstellen generiert, sowie Darstellung des Testablaufs in einer grafischen Oberfläche.

Marcel Neumann*, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Einer der Haupttransportmittel in Deutschland, mit denen Güter transportiert werden, ist der Lastkraftwagen. So wurden im Jahre 2013 ungefähr 2,9 Milliarden Tonnen an Gütern transportiert [1]. Um als Spedition einen Überblick über seine Lastkraftwagen und Fahrer zu behalten, können Systeme für das Flottenmanagement in die LKWs eingebaut werden, sogenannte Telematik Einheiten. Die Firma Trendfire Technologie GmbH entwickelt schon seit Jahren ihre eigene Software und Hardware für Telematikeinheiten. Der trendbeeTRUCK (siehe Abbildung 1) ermittelt den aktuellen GPS Standort, analysiert die Fahrweise, überwacht die Lenkzeiten des Fahrers und vieles mehr [2].



Abbildung 1: Telematikeinheit von Trendfire

Diese Einheit wird direkt in einen Lastkraftwagen eingebaut, in dem sie dann dessen Daten erfasst, analysiert und den Speditionen bzw. dem Fahrern zur Verfügung stellt. Zu diesen Daten gehören zum Beispiel Geschwindigkeit, Kilometerstand und Tachographendaten. Über die Flotten Management Schnittstelle, kurz FMS, werden diese an die Telematikeinheit weitergeleitet. Diese standardisierte Schnittstelle wurde von den europäischen Lastkraftwagen Herstellern im Jahre 2002 erstellt und bietet so eine Möglichkeit, herstellerübergreifend Telematik Anwendungen zu entwickeln. [3] Über den Global System for Mobile Communications Standard, kurz GSM, werden dann die gesammelten Daten an ein Rechenzentrum gesendet und können über ein Webinterface von den Speditionen abgerufen werden (siehe Abbildung 2).

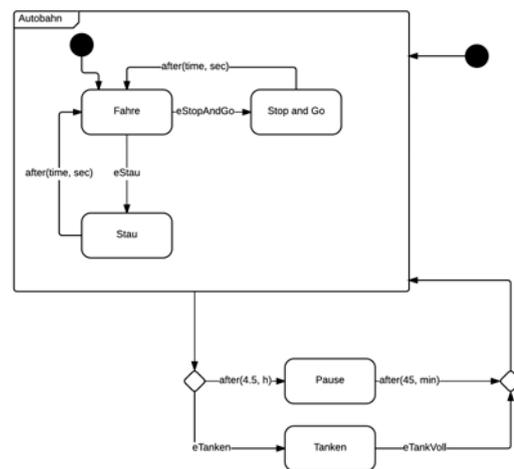


Abbildung 2: Ausschnitt des Zustandsautomaten

Für das Qualitätsmanagement soll nun ein neues Testtool entwickelt werden. In der dafür ausgeschriebenen Bachelorarbeit soll für die Telematikeinheit eine Testumgebung realisiert werden, die mehrere solcher Einheiten einem Langzeittest unterzieht und dabei ihre Funktionsfähigkeit überprüft. Diese Testumgebung soll an mehrere solcher Telematikeinheiten ständig neue und gültige Daten schicken. Um dies zu gewährleisten ist es notwendig, eine Simulation zu entwickeln, die eine Fahrt in einem LKW modelliert und dabei realistische Daten wie z.B. Geschwindigkeit, Kilometerstand, Tachographenstand und Motortemperatur generiert. Zusätzlich sollen aktuelle Statusdaten, der Telematikeinheit, wie Speicherauslastung, Prozentsauslastung, Laufzeit, Versionsnummer und Identifikationsnummer zu einer Einheit in der Teststation ausgelesen werden. Da auf der Einheit eine embedded Linux Version läuft, ist es möglich, sich über eine Debug Schnittstelle an der Box anzumelden und dann über diese Leitung mit dem Betriebssystem zu kommunizieren. Die gesammelten Daten zu den Einheiten, die getestet werden und die Daten, die von der Simulation gene-

*Diese Arbeit wurde durchgeführt bei der Firma Trendfire Technologies GmbH, Böblingen

riert werden, sollen dann auf einer grafischen Oberfläche dargestellt werden. Die Simulation basiert auf einem Zustandsdiagramm. Da die Simulation bei einem Dauerlauf nicht immer die gleichen Zustände durchlaufen soll, werden manche Ereignisse zufällig generiert. Dadurch bekommt der Zustandsautomat Eigenschaften einer Markow Kette, da die Zustandsfolge für jede neu angefangene Simulation anders ist als die der vorherigen Simulation. So simuliert die Simulation eine Fahrt auf einer Autobahn. Hier kann es nun passieren, dass es zu einem Stau kommt oder dass der Fahrer eine Pause einlegt oder Tanken muss. Dadurch können Daten generiert werden, die einer echten LKW Fahrt sehr nahe kommen, um dies zu erreichen werden Zustände für Landstraßen, Autobahnen, Dörfer, Städte und Ampeln definiert, in denen entsprechend Werte generiert werden.

Die so generierten Daten über den Fahrverlaufes werden als CAN-Nachricht nach dem FMS Standard formatiert und an die Tele-

matikeinheit verschickt. Dafür wird der PCAN USB-CAN Adapter von PEAK benutzt, mit dem man über USB von einem Rechner aus CAN-Nachrichten verschicken kann. Des Weiteren sollen Daten von einem digitalen Tachographen simuliert werden. Dabei sollen wie bei einem echten digitalen Tachographen jeder Sekunde über eine serielle Leitung Daten zu der Lenkzeit, Arbeitszeit, Bereitschaftszeit, Ruhezeit, Geschwindigkeit, gefahrene Kilometer, aktuelle Zeit und Identifikationsnummern zu Fahrer und Fahrzeug übertragen werden. Der Vorteil einer auf Zustandsautomaten basierenden Simulation bietet hier, dass einfach differenziert werden kann, in welchem Zustand (Arbeit, Ruhe, Fahrt) sich der Fahrer befindet und dementsprechend eine Tachographen Nachricht generiert werden kann. In Zukunft soll diese Testumgebung die hohe Qualität der der Hard- und Software von Trenfire Technologies GmbH weiterhin aufrechterhalten.

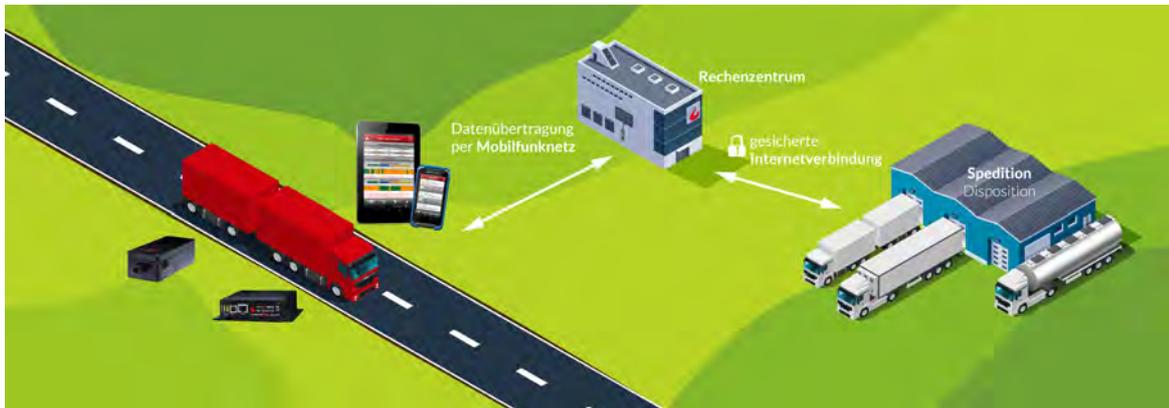


Abbildung 3: Überblick über die Funktionsweise einer Telematikeinheit

- [1] http://www.kba.de/DE/Statistik/Kraftverkehr/deutscherLastkraftfahrzeuge/Inlandsverkehr/2013_vd3_kurzbericht.html?nn=652360
 [2] <https://www.trendfire.com/fahrzeugortung.htm>
 [3] http://www.fms-standard.com/Truck/down_load/fms_document_ver03_vers_14_09_2012.pdf

Bildquellen:

- Abbildung 1,3: Trendfire Technologies GmbH
- Abbildung 2: Marcel Neumann

Modellbasierte Entwicklung einer unterlagerten Kaskadenregelung für den Antrieb eines Pedelecs auf Basis eines bürstenlosen Gleichstrommotors mit Hilfe von MATLAB/Simulink einschließlich automatischer Generierung des Quellcodes für die Implementierung des Reglers auf einem Mikrocontroller der Familie dsPIC33

Arno Orendi*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Elektromobilität ist in den Medien ein sehr präsent Thema. Einen großen Anteil hieran haben E-Bikes. Auf diesem Themengebiet forscht EVOMOTIV in der Entwicklung eines eigenen *Pedal Electric Cycles*, kurz Pedelec. Ziel ist es, ein Pedelec zu entwickeln, welches nicht auf den ersten Blick als solches zu erkennen ist.

Herzstück des unterstützenden Antriebes ist in diesem Fall ein *Brushless Direct Current* Motor (BLDC-Motor). Er ist ein Ableger der Synchronmaschine. Seine Bauart unterscheidet sich dadurch, dass er keine Bürsten bzw. Schleifkontakte benötigt. Bei dem verwendeten Motor handelt es sich zusätzlich um eine Außenpolmaschine, dies bedeutet, dass die Permanentmagnete außen am Gehäuse befestigt sind. Abbildung 1 zeigt den prinzipiellen Aufbau dieser Maschine.

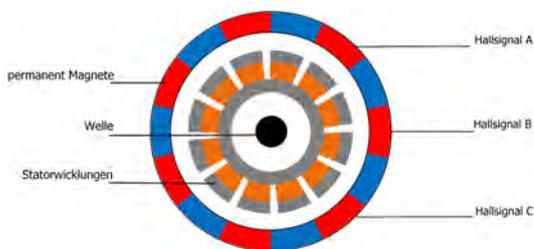


Abbildung 1: Aufbau eines BLDC-Motors

Außen sind die 7 Polpaare zu erkennen. Die 12 innenliegenden Statorwicklungen sind starr befestigt, lediglich das Gehäuse mit den Magneten und die Antriebswelle in der Mitte rotieren.

Der BLDC wird mit Hilfe einer Blockkommutierung (*Six-Step-Commutation*) in Betrieb genommen. Dabei wird in sechs Schritten, anhand einer Wahrheitstabelle, eine dreifache Halbbrückenschaltung so angesteuert, dass Wechselspannungen an den drei Phasen des Motors anliegen. Die Wahrheitstabelle ist abhängig von der Position des Rotors. Diese wird

mit Hilfe von Hall-Sensoren erfasst [1].

Als Grundlage zum Aufbau der Regelung dient ein mathematisches Modell des Motors. Dieses Modell ist in eine elektrische und eine mechanische Teilstrecke zu unterteilen. Die daraus resultierenden Gleichungen werden über die Entwicklungsumgebung MATLAB/Simulink in ein Modell überführt. Anhand dieses Modells werden die Regler berechnet und aufgebaut.

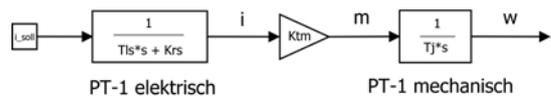


Abbildung 2: Mathematisches Modell eines BLDC-Motors

Das obige Schaubild zeigt das entworfene mathematische Modell des Motors.

Die Regelung des Motors erfolgt über die Spannung, welche an den Phasen des Motors anliegt. Sie ist auch die Stellgröße. Resultierend aus der Spannung fließt ein Strom, welcher proportional zu dem entstehenden Drehmoment ist. Der Stromregler stellt über den Tastgrad, auch *Duty Cycle*, eine Spannung ein. Der Tastgrad wird der Pulsweitenmodulation (PWM) zugeführt. Dadurch werden mit Hilfe der Wahrheitstabelle die entsprechenden Schalter angesprochen. Der Mittelwert aller Phasen gibt die notwendige Spannung und somit auch den Strom an. Hierdurch kann die Drehzahl und das Drehmoment des Motors gesteuert werden.

Die Regelung des Motors wird in Form eines Kaskadenreglers aufgebaut. Dabei wird das Drehmoment unterlagert mit Hilfe eines Zustandsreglers geregelt. Überlagert dazu wird die Drehzahl geregelt. Für beide Regler werden zusätzlich Führungsgrößen aufschaltungen verwendet, um Regelabweichungen im eingeschwungenen Zustand zu verhindern. Der prinzipielle Aufbau der Regelung ist in Abbildung 3 dargestellt.

*Diese Arbeit wurde durchgeführt bei der Firma EVOMOTIV GmbH, Leinfelden-Echterdingen

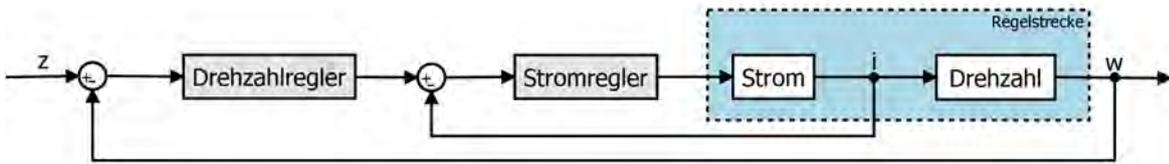


Abbildung 3: Kaskadenregelung

Der Test des entworfenen Systems erfolgt in zwei Schritten. Zuerst wird ein Simulink-Modell mit einer Nachbildung des BLDCMotors und dem Wechselrichterschalter (dreifache Halbbrücke) erstellt. Der BLDC-Motor Block wird dann mit realen Motorparametern parametrisiert. Diese sind die maximale Drehzahl, die Induktivität und auch das Massenträgheitsmo-

ment des Motors. Anhand des Modells kann vorab eine Auslegung der Regler erfolgen.

Der zweite Schritt ist die Inbetriebnahme des Reglers mit der Zielhardware. Diese besteht aus Evaluations-Board, Mikrocontroller und Motor. Abbildung 4 zeigt den vereinfachten Versuchsaufbau.

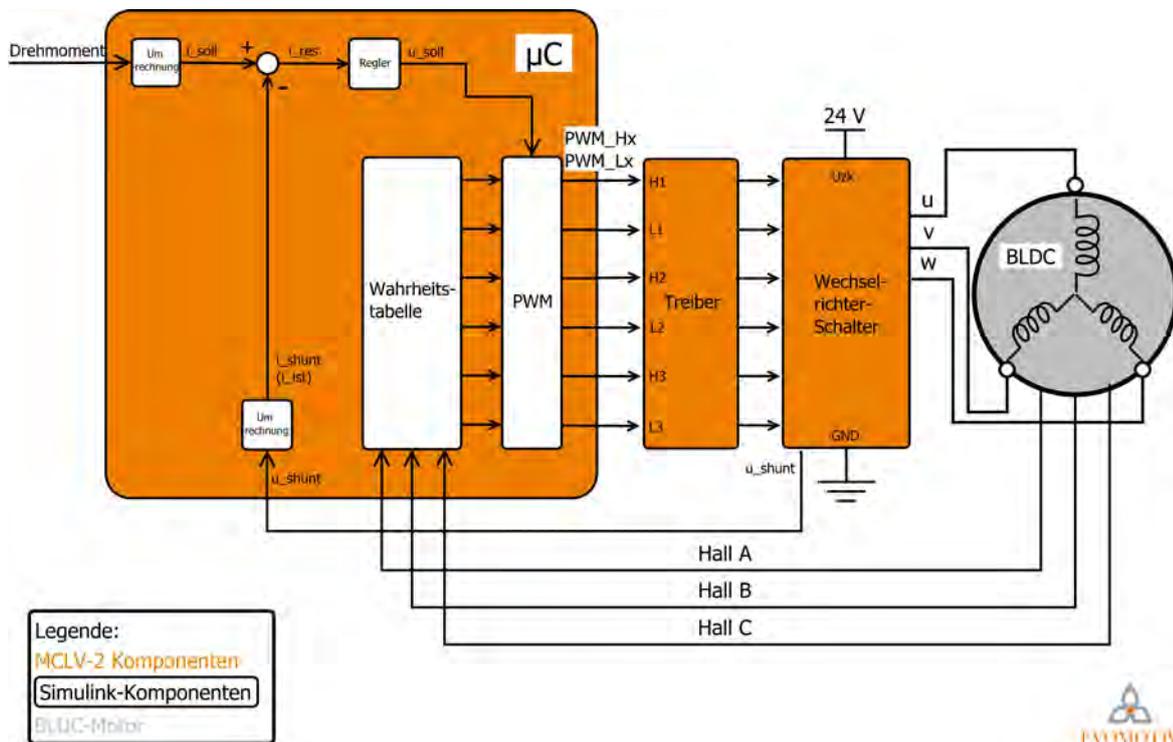


Abbildung 4: Vereinfachter Versuchsaufbau

[1] Microchip Technology. Brushless DC (BLDC) Motor Fundamentals. Dallas: Microchip Technology Inc., 2003

Bildquellen:

- Abbildung 1-4: Eigene Abbildung

Modellierung der Leistungselektronik eines Kfz-Hybridantriebs

Antonio Pomarico*, Werner Zimmermann, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Für das Jahr 2020 wird die europäische Gesetzgebung die CO₂-Abgaswerte auf 95 $\frac{g}{km}$ setzen. Das ist eine Absenkung von ca. 27% auf die bisherigen Abgaswerte [1]. Diese Abgasreduktion kann trotz der optimierten Verbrennungsmotoren kaum noch durch motorische Maßnahmen allein erreicht werden. Eine mögliche Lösung dafür ist die Elektrifizierung bzw. Hybridisierung des Antriebsstrangs. Dabei wird zwischen Hochvolt-Bordnetzen (HV-Bordnetz) und 48 Volt-Bordnetzen (48V-Bordnetz) unterschieden. HV-Bordnetze werden für Elektrofahrzeuge und Hybridfahrzeuge eingesetzt, da diese hohe Leistungsanforderungen besitzen. Während 48V-Bordnetze für MildHybridfahrzeuge eingesetzt werden können, um die Komfortfunktionen zu unterstützen, wie zum Beispiel Start-Stop-Funktionalität, Rekuperation oder E-Boost aber auch Nebenaggregate wie Klimaanlage, Lenkhilfen oder Heizung.



Abbildung 1: Mercedes-Benz S 500 Plug-In Hybrid

In einem HV-Bordnetz werden unter anderem IGBT-Inverter verwendet. Das sind Halbleiterbauelemente, die die Vorteile eines Bipolartransistors und Vorteile eines Feldeffekttransistors vereinigen [2]. Diese sind für sehr hohe Spannungen (> 150V) ausgelegt und können damit die hohen Anforderungen der elektrischen Maschine erfüllen. Zum Schutz der IGBTs im Sperrbetrieb werden Freilaufdioden eingesetzt. Diese übernehmen den Laststrom der IGBTs, so dass diese nicht belastet werden. Befindet sich der Motor nicht im Motorbetrieb sondern im Generatorbetrieb, wird über die Freilaufdiode die Rückspeisung der elektrischen Energie ermöglicht.

Grundsätzlich wird die Energie in Form einer Gleichspannung aus einer elektrischen Energiespeicher bereitgestellt. Durch entsprechende Ansteuerungsverfahren (PWM, Blocktaktung o.Ä.) und mithilfe einer Leistungselektronik (Abbildung 2) lässt sich die Gleichspannung in eine Wechselspannung umformen um somit den elektrischen Motor zu betreiben [3].

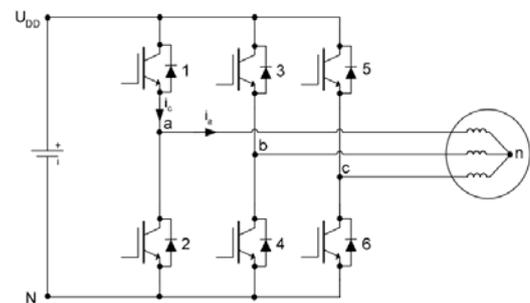


Abbildung 2: Schaltbild eines Inverters

Über die gesamte elektrische Antriebsstrangkette entstehen Verluste. Diese ergeben sich durch die Energieumwandlung von elektrischer Energie bis zur mechanischen Fahrerenergie. Die dadurch entstehenden Verluste werden als Wärme freigesetzt und sorgen für einen Temperaturanstieg am Leistungsmodul. Hierbei soll beachtet werden, dass alle Halbleiterparameter selbst temperaturabhängig sind und daher auch einen Einfluss auf die Verlustleistung haben. Die Leistungselektronik wird durch Temperaturhübe belastet, dadurch beschränkt sich die Lebensdauer eines solchen Bauteils. Die Verluste in einem IGBT und in einer Freilaufdiode unterteilen sich jeweils in Durchlass- und Schaltverluste sowie vernachlässigbare Sperrverluste.

Ziel dieser Arbeit ist es, das Verhalten einer Leistungselektronik für die dynamische Beschreibung eines Hybridantriebsstrangs modular zu simulieren. Durch die Modularität wird eine einfachere Parametrierung erreicht, aber auch dass die einzelnen Bausteine als Schnittstelle für weitere Systeme verwendet werden können. Im Gegensatz zur Komponentenbetrachtung ist für die Gesamtsystembetrachtung wichtig, dass die einzelnen Module keine

*Diese Arbeit wurde durchgeführt bei der Firma Daimler AG, Stuttgart

hochdetaillierte Implementierung bekommen, da diese sehr rechenaufwendig und daher eine lange Simulationszeit in Anspruch nehmen. Darüber hinaus wird ein Modul zur Betrachtung

der thermischen Belastung der Komponenten benötigt, um später Aussagen über die Lebensdauer treffen zu können.

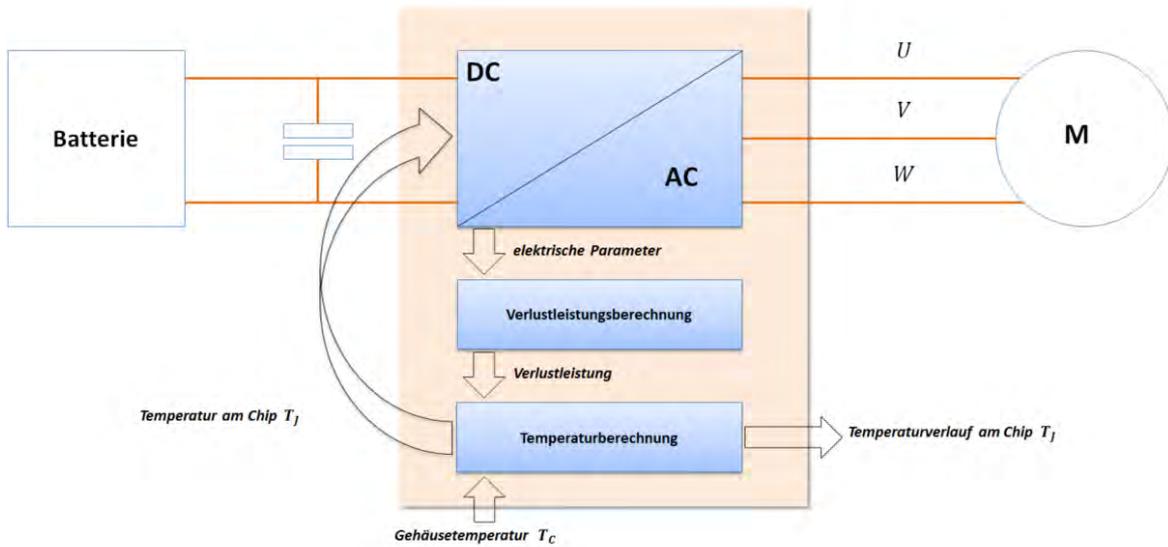


Abbildung 3: Modellübersicht

Für die Modellierung der Komponenten wird hierbei die Toolbox Simscape aus MATLAB/Simulink verwendet. Simscape ist eine mögliche Umgebung zur Modellierung und Simulation von physikalischen Systemen. Die Stärke dieser Toolbox liegt darin, dass verschiedene Domänen (elektrisch, mechanisch, hydraulisch, thermisch usw.) in einer einzigen Umgebung als Gesamtmodell modelliert und simuliert werden können.

Um einen funktionsfähigen 3-Phasen Wechselrichter zu realisieren benötigt dieser diverse Schnittstellen (Abbildung 3). Für die Bereitstellung der Gleichspannung am Eingang des Inverters wird ein bereits bestehendes Batteriemodell aus Simulink genutzt. Da es sich nicht um eine ideale Gleichspannung handelt, wird diese über einen Zwischenkreiskondensator geglättet und gepuffert. Um den IGBT-Inverter anzusteuern wird ein PWM-Modul modelliert. Am Ausgang des Inverters wird ein einfaches Motormodell modelliert.

[1] ICCT, „European Vehicle market statistics,“ 2014. [Online]. [Zugriff am 30 April 2015]

[2] http://de.wikipedia.org/wiki/Bipolartransistor_mit_isolierter_GateElektrode [Zugriff am 01. Mai 2015]

[3] H. Schäfer, Neue elektrische Antriebskonzepte für Hybridfahrzeuge, expert verlag, 2007, p. 264

Bildquellen:

- Abbildung 1: www.mercedes-benz.de
- Abbildung 2: Infineon – IGBT Power Losses Calculation Using the Data-Sheet Parameters
- Abbildung 3: Eigene Abbildung

Analyse, Anwendung und Erweiterung der Methode Lean UX am Beispiel eines Quantified Self Monitor

Janet Protzer*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Schneller, höher, weiter, besser. Die Forderung nach Perfektion, die nicht nur im Charakter des Menschen, sondern auch in dessen Neurologie verankert ist, ist einer der Gründe, warum der Mensch heute an der Spitze der Nahrungskette steht. Dieser Drang nach Verbesserung, möglichst bis zur Perfektion, steigert nicht nur die Forderungen des Menschen an sich selbst, sondern auch die Bedürfnisse und Erwartungen in Bezug auf ein Softwareprodukt. Diese hohen Erwartungen steigern letztendlich die Ansprüche an den Entwicklungsprozess, da die Prozesse an die neuen Qualitätsansprüche angepasst werden müssen.

Die Motivation dieser Arbeit liegt in dieser aktuellen Veränderung der Methodiken und Prozesse in der Softwareentwicklung, speziell am Beispiel der Methode "Lean UX". Laut Jeff Gottelf, einem Vorreiter des Lean UX, eine Denkweise, ein Prozess, aber auch eine Methode, mit der Software effektiver und Benutzerorientierter erstellt werden kann [1].



Abbildung 1: Lean UX Zyklus

Im Allgemeinen arbeitet Lean UX nach dem abgebildeten Leitsatz "Think – Make – Check". Es wird intuitiv, mit Vermutungen über das Produkt und den User, gearbeitet, die Ideen im Team visualisiert und mit realen Usern getestet. Diese Methodik wird solange iteriert, bis das bestmögliche Produkt für den User entstanden ist. Es wird demnach das Softwareprodukt mehr und mehr aufgebaut und verfeinert, bis es die Benutzerbedürfnisse befriedigt, ohne im Vorfeld Kenntnisse über den User zu besitzen. Diese Methoden, Denkweisen und

der Prozess von Lean UX wird im Rahmen dieser Arbeit analysiert und angewendet. Ziel ist hierbei jedoch weniger ein Produkt unter Anwendung der Methoden von Lean UX zu erstellen, sondern mehr die Elemente von Lean UX zu testen, die Anwendbarkeit und Umsetzbarkeit zu bewerten und die Vor- und Nachteile zu erkennen. Als Fall- und Übungsbeispiel während eines hierfür durchgeführten Workshops dient die Konzeption einer App zur Selbstdokumentation mit dem Schwerpunkt Quantified Self.



Abbildung 2: Quantified Self Logo

„Quantified Self“ (QS), also „Selbstquantifizieren“, was so viel bedeutet wie „sich selbst abmessen“, ist eine Bewegung, die sich mit dem Versuch der Selbstperfektion beschäftigt. Auch wenn die Bewegung unter ihrem Namen unbekannt scheint, ist sie nicht neu. Das Prinzip des „Self Tracking“, also Selbstvermessen, ist bereits aus Bereichen wie dem Leistungssport bekannt. Selbstquantifizierer messen Schlaf-tiefe, sportliche Aktivitäten, Schritte, aber auch ausgefallene Dinge wie Bakterien im Mund und verlorene Haare. Diese Datensammlung dient dazu Zusammenhänge zwischen Aktionen und Reaktionen zu erkennen und Erkenntnisse, gemäß dem Motto der QS (siehe Abbildung) „self knowledge through numbers“, über die eigene Person und der Handlungen zu erlangen und diese dann bewusst beeinflussen zu können. Hierzu gibt es heutzutage unzählige Möglichkeiten, da der Mensch zum einen die gesellschaftliche Freiheit, zum anderen die technologische Freiheit hat, die mit dem Stand der Technik einhergeht. Früher diente ein Zettel und Stift als einziges Werkzeug, heute werden hierfür sogenannte Wearable Devices eingesetzt. Diese erledigen die Dokumentation in vielen Bereichen selbst und speichern die gesammelten Daten ab. Möchte man diese Daten nun aber auswerten und visualisieren, muss man zu weiteren Mitteln greifen. Diese Mit-

*Diese Arbeit wurde durchgeführt bei der Firma Fraunhofer Institut Anwendungszentrum KEIM, Esslingen

tel können zum Beispiel Apps auf dem Smartphone sein. Geeignete Software zu finden ist bei einem individuellen Thema wie diesem jedoch äußerst schwierig, obwohl es an Auswahl auf dem Markt nicht mangelt. Jeder „Self-Tracker“ oder „Selbst-Beobachter“ entscheidet selbst welche Daten er erhebt und welche ihm wichtig sind, in welchen Intervallen diese Werte erhoben und aufgezeichnet werden. Diese Entscheidungen hängen immer stark von der Zielsetzung der Analyse ab [2]. Und unter diesen Voraussetzungen sind viele Anwendungen letztendlich nicht individualisierbar genug, enthalten zu viele unnötige Funktionen und zu wenige der gewünschten Funktionen, oder sind einfach längst nicht mehr aktuell, wenn sie endlich auf den Markt gebracht werden. Anhand dieses speziellen Themas mit den facettenreichen Usern und vielseitigen Anwendungsfeldern wird deutlich, dass es oftmals schwer ist die Bedürfnisse der User zu erfüllen, aber dass der User dennoch im Zentrum der Software steht. Es ist notwendig ihn zu kennen und zu verstehen, um eine Software zu erstellen, die dessen Bedürfnissen gerecht wird. „Software muss in immer kürzeren Zeiträumen mit immer mehr Anforderungen realisiert werden; die Anforderungen an Qualität steigen dabei kontinuierlich“ [3]. Diese Ziele sind mit alten Vorgehensmodellen, wie zum Beispiel dem Wasserfallmodell, kaum mehr zu erreichen. Bisherige Modelle sind nicht Nutzerorientiert genug und kaum iterativ, beinhalten langwierige und ausgiebige Forschung und Vorarbeit, Dokumentation und Planung und deren strikte Befolgung. In Lean UX wird der komplette Projektzyklus schneller durchlaufen, und damit der heutigen Zeit angepasst. Außerdem werden falsche Annahmen und Anforderungen früher entdeckt und können behoben werden. Die Kernaussage „Reduce waste“ bezieht sich auf diese Reduzierung der benötigten Ressourcen, speziell Zeit, Kapital und Personal, die ansonsten unnötig ver-

braucht werden. Dadurch soll der Prozess effektiver als bisherige Vorgehensmodelle sein.



Abbildung 3: Lean UX Zusammenfassung

Lean UX, grob bereits auf der obigen Abbildung zusammengefasst, ist ein stark auf Visualisierung bauender und kreativer Prozess. Er arbeitet sehr mit gegenseitiger Inspiration und Beteiligung unterschiedlicher Charaktere und Kenntnissbereiche. Diese Zusammenarbeit ist für den Prozess charakteristisch und wesentlich. Durch die Anwendung der Methoden und Denkweisen von Lean UX wird die Entwicklungszeit drastisch verkürzt, ohne aber am Endergebnis zu sparen. Das Erreichen der Ziele, beziehungsweise eines qualitativen Endproduktes, soll durch immer wieder kehrende Meetings und Tests gesichert werden. Prinzipiell baut Lean UX auf Inhalten von drei Elementen auf: Design Thinking, Agile Softwareentwicklung und Lean Startup. Es wurde von den innovativsten Ansätzen der Zeit das Beste zusammen genommen und eine neue Innovation daraus geformt. Im Ganzen beinhaltet Lean UX viele neue Ansätze, aber auch viele altbekannte, die besser in den Softwareerstellungsprozess integriert sind oder besser umgesetzt werden.

- [1] Jeff Gothelf; Lean UX: Applying Lean Principles to Improve User Experience, 10. Auflage, Californien. O'Reilly Media, Inc., 2013, Seite 109
 [2] <http://was-ist-quantified-self.de/>, abgerufen: 25.04.2015
 [3] <http://www.scrum-kompakt.de/grundlagen-des-projektmanagements/vorgehensmodelle-in-der-softwareentwicklung/>, abgerufen: 23.04.2015

Bildquellen:

- Abbildung 1: <https://s-media-cache-ak0.pinimg.com/736x/c9/0a/79/c90a79e1d1d46913dd199fb89df81a6c.jpg>, abgerufen: 03.05.2015
- Abbildung 2: <http://quantifiedself.com>, abgerufen: 03.05.2015
- Abbildung 3: <https://s-media-cache-ak0.pinimg.com/736x/ed/8b/a4/ed8ba40bf47ff9adc35250e28ee7beef.jpg>, abgerufen: 11.05.2015

Entwurf und Realisierung eines Fuzzingtools für kryptographische Protokolle

Stefan Ringhoffer*, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

In der heutigen Zeit spielt das Internet im alltäglichen Leben eine immer bedeutendere Rolle, so zum Beispiel bei der Verwendung von Social Media Plattformen oder der Verwaltung des eigenen Bankkontos per Onlinebanking. Die Software, welche solche Dienste bereitstellt, tauscht über das Internet sensible Daten aus. Damit solche sensiblen Daten nicht von unberechtigten Personen gelesen oder verändert werden können, ist die Benutzung von kryptographischen Protokollen zur sicheren Kommunikation unabdingbar. Doch auch die Software, welche die sichere Kommunikation gewährleistet, kann aufgrund ihrer hohen Komplexität eklatante Schwachstellen aufweisen. Dies zeigen, die in jüngster Vergangenheit aufgetretenen Schwachstellen wie zum Beispiel Heartbleed [3] oder Poodle [4]. Das Fuzzing ist eine mögliche Methode, Schwachstellen in einer Implementierung aufzuspüren.

Der Begriff des Fuzzings fällt unter den Oberbegriff des White-, falls der Sourcecode oder der jeweilige Standard einsehbar ist, oder des Blackboxtesting, falls der Sourcecode oder der jeweilige Standard nicht einsehbar ist (Abbildung 1). Bei dieser Methode werden an eine Eingabeschnittstelle einer zu testenden Implementierung zufällig generierte Daten gesendet. Danach wird die Implementierung auf Fehlverhalten, wie zum Beispiel ein Bufferoverflow, der zum Beispiel durch eine ungesicherte strcpy Funktion ausgelöst werden kann, untersucht. Bei einem Bufferoverflow kann die Rücksprungadresse im Speicher durch den Angreifer verändert werden. Wenn dies gelingt, kann dieser seinen Code mit den Rechten des eigentlichen Programmes ausführen. Ein Fuzzer stellt die folgenden Funktionen bereit: Einlesen der Testfälle, Ausführen der Testfälle, Überwachung der zu testenden Implementierung und Dokumentation der gewonnenen Informationen aus der Überwachung.

Im Rahmen dieser Arbeit werden zuerst die Grundlagen des Fuzzings und eines kryptographischen Protokolles, in diesem Fall das SSH-Protokoll [1], erarbeitet .

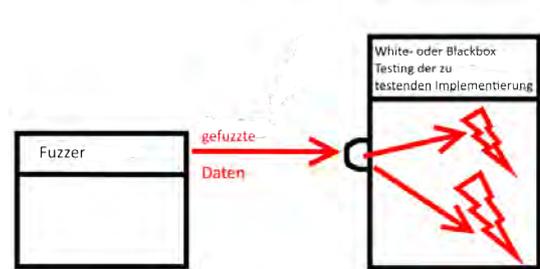


Abbildung 1: Fuzzer als White- und Blackboxtesting

Anschließend wird aus diesen Grundlagen die allgemein gültigen Anforderungen und Möglichkeiten für das Fuzzing eines kryptographischen Protokolles abgeleitet. Die Möglichkeiten ein kryptographisches Protokoll zu fuzzen, lässt sich dabei in zwei Teile trennen. Der erste Teil beinhaltet die Nachrichten, die unverschlüsselt über das Netzwerk geschickt werden. Hierzu zählen die Nachrichten, die den Austausch der Algorithmen und den Schlüsselaustausch bewerkstelligen. Der zweite Teil des Fuzzing eines kryptographischen Protokolles umfasst das Fuzzing der Authentikationsphase. Hier werden Passwörter, Public-Keys und Längfelder mit mutierten Werten verändert. Hier gilt die Anforderung an den Fuzzer, dass die nun verschlüsselten Nachrichten entschlüsselt und nach der Mutation der zu fuzzenden Nachricht mit einem neuen Nachrichtenauthentifizierungscode verschlüsselt werden. Am Beispiel des SSH-Protokolles wären dies das SSH-Transportprotokoll und das SSH-Authentikationsprotokoll auf mögliche Fuzzingparameter zu untersuchen. Dienste eines Protokolles, die auf einer gesicherten Verbindung aufbauen, werden daher beim Fuzzing eines kryptographischen Protokolles nicht berücksichtigt. Dies wäre am Beispiel von SSH das SSH-Verbindungsprotokoll, das es nicht zu berücksichtigen gilt.

Aus den erstellten Anforderungen und Möglichkeiten wird im Zuge der Bachelorarbeit ein Entwurf eines Fuzzers für das SSH-Protokoll erstellt. Als zu testende SSH-Serverimplementierung wird OpenSSH [2] ge-

*Diese Arbeit wurde durchgeführt bei der Firma secuvera GmbH, Gäufelden

wählt. Dieser SSH-Fuzzer wird nach dem Prinzip eines Interceptors aufgebaut. Dabei werden die empfangenen Nachrichten des Clients auf dem Localhost abgefangen und durch den Interceptor an der zu fuzzenden Position der Nachricht verändert und dann an den Server weitergeleitet. Dieses Prinzip ist in Abbildung 3 dargestellt. Desweiteren müssen für den Fall, dass durch das Fuzzing ein Fehlverhalten des OpenSSH-Servers hervorgerufen wird, dem Entwurf des SSH-Fuzzers weitere Funktionsblöcke zur Verbindungsüberprüfung, Ressourcenüberwachung der zu testenden Implementierung und Auswertung des Loggingfiles des SSH-Servers hinzugefügt werden. Diese drei Überwachungsmethoden des OpenSSH-Servers werden jeweils nach dem Ausführen eines Fuzzingtest durchgeführt. Die letzte Funktion des SSH-Fuzzers ist, die Informationen des ausgeführten Testfalls in einem Loggingfile festzuhalten.

Damit der Interceptor weiß, welche Nachricht er abfangen und verändern muss, wird eine Testfalldatenbank diese Informationen dem Interceptor übermitteln. Die Testfalldatenbank wird in einer Konfigurationsdatei hinterlegt. Die Konfigurationsdatei bietet die Möglichkeit, zu einem späteren Zeitpunkt von einem Fuzzingdatengenerator mit beliebig vielen Testfällen gefüllt zu werden.

In Abbildung 2 ist die Veränderung des Interceptors exemplarisch anhand eines Testfalls, der das SSH-Längenfeld der SSH-Nachricht SSH_MSG_KEY_INIT mit dem Hexadezimalwert 0xffffffff fuzzen wird, dargestellt.

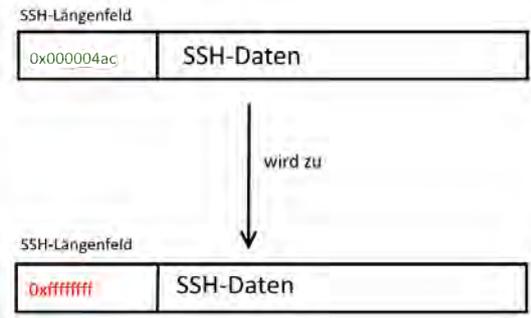


Abbildung 2: SSH-Längenfeld Testfall

Schließlich wird der Entwurf des SSH-Fuzzers in einer Implementierung enden. Hierzu werden die benötigten Funktionsblöcke über die Skriptsprache Perl realisiert.

Zum Abschluss der Bachelorarbeit werden auf Basis der Ergebnisse die Möglichkeiten der Quantisierung der Testfälle und die Möglichkeiten der Performanceverbesserung erarbeitet.

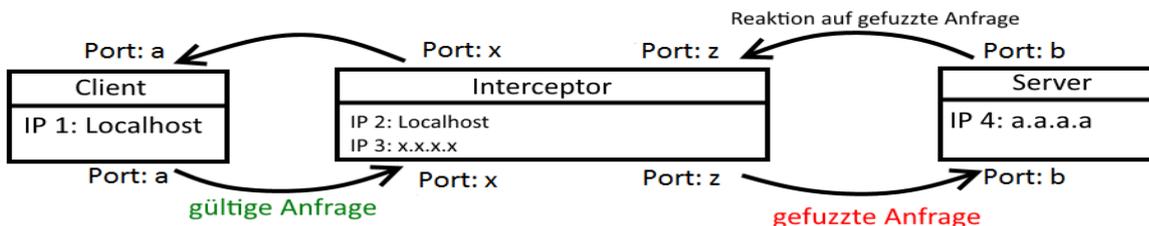


Abbildung 3: Interceptor

- [1] The Secure Shell Protocol Architecture - RFC 4251, T. Ylonen, <http://www.ietf.org/rfc/rfc4251.txt> [03.05.2015]
- [2] OpenSSH Implementierung, <http://www.openssh.com/> [03.05.2015]
- [3] Poodle Exploit <http://www.heise.de/security/artikel/Poodle-So-funktioniert-der-Angriff-auf-die-Verschlusselfelung-2425250.html> [12.05.2015]
- [4] Heartbleed Exploit, <http://www.heise.de/security/artikel/So-funktioniert-der-Heartbleed-Exploit-2168010.html> [12.05.2015]

Bildquellen:

- Abbildung 1: eigene Darstellung
- Abbildung 2: eigene Darstellung
- Abbildung 3: eigene Darstellung

Analyse und Verbesserung der Usability eines Content-Management-Systems

Jens Schmid*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Ein Content-Management-System oder kurz CMS ist eine einfache Pflegeschnittstelle, die es Unternehmen ohne Know-how im Gebrauch von HTML oder im Umgang mit grafischen Editoren erlaubt über HTML-Formulare Inhalte aktuell zu halten. [1]

Das im Laufe dieser Arbeit untersuchte Content-Management-System ist speziell für die Bedürfnisse der Branche Produktionsverbindungshandel (PVH) entwickelt worden. Es wird in erster Linie im B2B Sektor eingesetzt um Firmenwebseiten, Onlinekataloge und Medienservices zu erstellen.

Bei professionellen Content-Management-Systemen ist vor allem der Umstand, dass diese oft eine sehr hohe Komplexität aufweisen, ausschlaggebend für die mangelnde Usability. Komplizierte Funktionen wie Übersetzungsmanagement sowie verschiedene Sprach- und Länderfreigaben der einzelnen Artikel steigern die Komplexität und machen es den Nutzern

schwer bis unmöglich ihre Arbeit mithilfe des Content-Management-Systemes zu erledigen.

Ziel dieser Arbeit ist es, durch eine Usability-Analyse, die Schwachstellen eines Content-Management-Systems aufzuzeigen und die Software dementsprechend zu verbessern. Zum Erkennen der Schwachstellen wird eine Anwenderbefragung, verschiedene Usability Studien sowie ein Vergleich mit anderen Content-Management-Systemen, wie zum Beispiel mit dem Open Source Content-Management-System TYPO3, durchgeführt.

Anhand dieser, durch die Analyse gewonnenen, Erkenntnisse wird eine neue, verbesserte Benutzeroberfläche konzeptioniert und anschließend prototypisch für einen Anwendungsfall überarbeitet. Am Ende dieser Arbeit sollen die getätigten Verbesserungen anhand eines Vergleiches der neuen mit der ursprünglichen Oberfläche evaluiert werden.

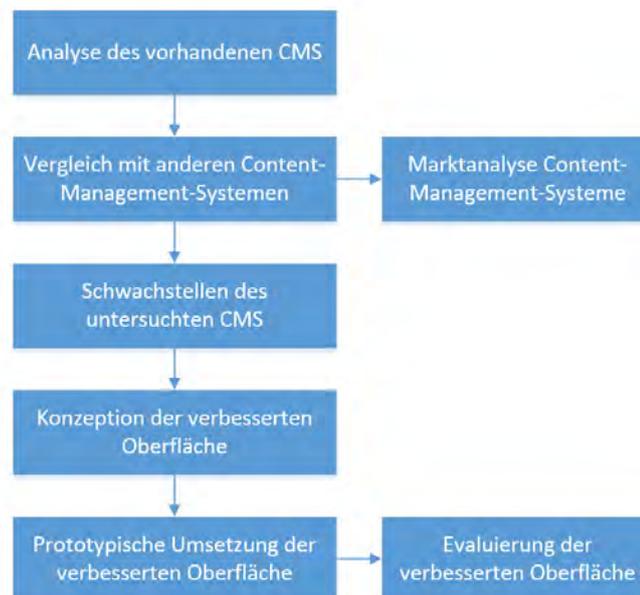


Abbildung 1: Vorgehensweise

*Diese Arbeit wurde durchgeführt bei der Firma crossbase mediasolution GmbH, Böblingen

“Der Begriff Usability oder auch Gebrauchstauglichkeit bezeichnet das Maß der Effektivität, Effizienz und Zufriedenheit, mit der Benutzer mit diesem System vorgegebene Ziele erreichen können.“ (DIN 9241 – Teil 11) [2]

Effizienz
Effektivität **Usability**
Zufriedenstellung

Abbildung 2: Usability

Ist die Usability eines Systems nicht ausreichend, kann dies den Nutzer durch schlecht zu bedienende Eingabemaschinen oder hundert von Pflichtfeldern daran hindern effizient zu arbeiten. Ein solches System frustriert den Nutzer und führt dazu, dass das System, welches den Nutzer eigentlich bei seiner Arbeit

unterstützen sollte, nur als Belastung angesehen wird. Zwischenlösungen und Problemumgehungen um die bestehende Lösung herum sind die Folge. Außerdem führen unzufriedene Nutzer zum Verlust des Vertrauens in die Software und dementsprechend auch zu mangelndem Vertrauen in den Softwarehersteller.

In den letzten Jahren ist der Begriff Usability und vor allem die Notwendigkeit mehr für eine gute Usability zu tun, bei vielen Softwareentwicklern im Sektor B2C angekommen, wie man an Beispielen wie dem iPhone deutlich erkennen kann. Für den B2B Sektor allerdings bleibt der Gedanke Usability häufig auf der Strecke. Für Software die der Endkunde gar nicht zu sehen bekommt sondern mit deren Hilfe Unternehmen ihre internen Vorgänge abwickeln, z.B. Bestellvorgänge aber auch bei den Redaktions- teilen von Content-Management-Systemen für die Pflege der Firmenwebseiten, wird der Usability Gedanke bei der Entwicklung meist zu Gunsten von zu vielen und zu komplizierten Funktionen vernachlässigt.



Abbildung 3: Usability Test Comic

[1] Stefan Jablonski, Christian Meiler. 2002. Web-Content- Managementsysteme: Springer Vieweg Verlag, 2002
[2] Markus Dahm. 2005. Grundlagen der Mensch-Computer-Interaktion: Addison-Wesley Verlag, 2005

Bildquellen:

- Abbildung 1: Eigene Darstellung
- Abbildung 2: <http://www.mittelstand-digital.de/MD/Redaktion/DE/PDF/mdays-mobile-ikt-anwendungen-quesseleit.pdf>
- Abbildung 3: <http://www.bugbash.net/comic/7.html?p=20>

Testautomatisierung von grafischen Oberflächen – Erhebung des Testbedarfs am Fraunhofer–Anwendungszentrum KEIM und prototypische Umsetzung

Marc Schubert*, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Testen ist ein essentieller Bestandteil der Softwareentwicklung. Ein großer Teil des Softwareentwicklungsprozesses wird dem Testen gewidmet, teilweise bis zu 50 % des Zeitaufwandes bzw. 50 % der Gesamtkosten [1] oder gar mehr [2]. Trotz des hohen Stellenwertes des Testens in der Softwareentwicklung wird das Testen von grafischen Oberflächen vernachlässigt [3] [4]. Häufig werden GUIs fast ausschließlich manuell getestet, was sehr aufwendig ist und nur eine geringe Testabdeckung bietet. Gründe für dieses Problem können einige Herausforderungen sein, die GUI-Tests mit sich bringen [3]:

- GUIs werden für Menschen und nicht für Computer entworfen, wodurch eine Automatisierung erschwert wird.
- GUIs sind eventbasiert, d. h. es müssen Benutzerevents für das Testen von GUIs simuliert werden.
- Es müssen andere Testabdeckungskriterien als für sonstige Softwaretests eingesetzt werden.

Auch am Fraunhofer IAO KEIM werden GUIs entwickelt. Da zunächst nicht klar war, in welchem Ausmaß GUIs dort getestet werden, wurde zunächst eine schriftliche Befragung durchgeführt, in der der konkrete Testbedarf ermittelt wurde. Die Umfrage wurde als schriftliche Online-Befragung über Google Docs durchgeführt. Die Entwickler wurden darin u. a. befragt, welche Arten von Anwendungen und Oberflächen sie entwickeln, in welchem Umfang generell getestet wird und welche Bedeutung die Testautomatisierung von GUIs für sie hat. Es war hierbei u. a. wichtig, herauszufinden, ob hauptsächlich manuell oder automatisiert getestet wird. Die Umfrage ergab, dass am KEIM nur selten automatisierte Tests durchgeführt werden, insbesondere GUI-Tests werden bisher fast ausschließlich manuell durchgeführt.

Daher soll im weiteren Verlauf eine prototypische Umsetzung einiger automatisierbarer Testfälle anhand einer ausgesuchten Web-Oberfläche geschrieben werden, die aufzeigen soll, welche Möglichkeiten und Vorteile,

aber auch welche Nachteile und Probleme eine Testautomatisierung von GUIs bietet. Es soll eine Art Vorlage entstehen, mit der die Entwickler am KEIM eine Testautomatisierung für weitere GUIs einführen können.

ALREADY REGISTERED?

If you have an account with us, please log in.

Email Address *

Password *

[Forgot Your Password?](#)

Abbildung 1: Beispiel für Login-Formular, das für GUI-Tests automatisiert getestet werden kann

Die Umsetzung soll mit Selenium erfolgen, einem Softwarepaket für die Automatisierung von Browsern. Selenium unterstützt das Scripting verschiedener Testfälle, welches über unterschiedliche Programmiersprachen erfolgen kann. Für diese Arbeit wurde Python als Programmiersprache für das Formulieren der Testfälle ausgewählt, da Python eine leicht zu erlernende Sprache ist und diese durch ihre Syntax eine kurze und übersichtliche Schreibweise erlaubt. Nachdem die Umsetzung fertiggestellt ist und den Entwicklern am Fraunhofer IAO KEIM vorgelegt worden ist, soll eine Bewertung erfolgen. Dafür werden die Entwickler am KEIM ein weiteres Mal befragt, um ein Feedback zu erhalten. Je nach Ergebnis wird entschieden, ob bzw. in welchem Umfang eine Testautomatisierung von GUIs am KEIM eingeführt werden soll.

Darüber hinaus wird in dieser Arbeit das Grundlagenthema Domain-Driven Design (abgekürzt DDD) behandelt. Domain-Driven Design ist eine Herangehensweise bzw. Denkweise für die Entwicklung komplexer Softwaresysteme. Der Name „Domain-Driven Design“ wur-

*Diese Arbeit wurde durchgeführt bei der Firma Fraunhofer IAO KEIM, Esslingen am Neckar

de geprägt durch das gleichnamige Buch von Eric Evans aus dem Jahre 2003 [5]. Das Domänenmodell als Abstraktion der Geschäftslogik einer Domäne, also des betrachteten Problembereichs, steht bei Domain-Driven Design stark im Fokus. Zu jeder Zeit im laufenden Softwareentwicklungsprozess soll die Implementierung des Domänenmodell umsetzen und umgekehrt soll das Domänenmodell die Implementierung repräsentieren. Zudem fordert DDD eine enge Zusammenarbeit der Entwickler mit den Fachexperten der Domäne. Diese Notwendigkeit ergibt sich daraus, dass die Entwickler ein System mit einer komplizierten Domäne nicht korrekt entwickeln können, wenn sie die Domäne nicht richtig verstehen. Daher müssen die Entwickler ein Verständnis für die Domäne aufbauen und das geht am besten über eine intensive Kommunikation mit den Fachexperten der Domäne. Die Verständigung zwischen allen Beteiligten erfolgt mit der sogenannten „Ubiquitous Language“, welche ein für das Fachgebiet typisches Vokabular enthält.

DDD orientiert sich an den agilen Methoden der Softwareentwicklung. Für DDD wird ein iterativer Entwicklungsprozess empfohlen, indem die Entwickler ein Verständnis für die Domäne entwickeln. DDD schlägt verschiedene Strategien für den Umgang mit komplexen Systemen vor, die es u. a. ermöglichen, das System in mehrere Modelle aufzuteilen, ohne am Ende

Integrationsprobleme zu verursachen. Zudem beinhaltet DDD die Definition mehrerer Bausteine, die für die Modellierung eines Domänenmodells verwendet werden sollen.

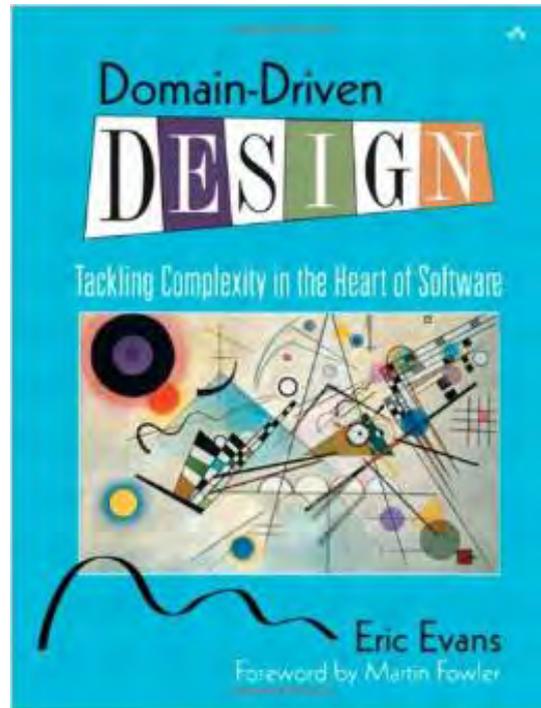


Abbildung 2: Buch „Domain-Driven Design“ von Eric Evans

- [1] G. J. Myers, „Methodisches Testen von Programmen,“ p. VII, 4. Hrsg., München, R. Oldenbourg Verlag, 1991.
- [2] A. Spillner und T. Linz, „Basiswissen Softwaretest,“ p. 15, Heidelberg, dpunkt.verlag GmbH, 2010.
- [3] A. Ruiz und Y. Wang Price, „GUI Testing Made Easy,“ TAIC-PART '08 Proceedings of the Testing: Academic & Industrial Conference – Practice and Research Techniques, p. 99, 2008.
- [4] A. Memon, „GUI Testing: Pitfalls and Process,“ Computer (Volume: 35 , Issue: 8), p. 87, 7 November 2002.
- [5] E. Evans, „ Domain-Driven Design. Tackling Complexity in the Heart of Software,“ Westford, Massachusetts: Addison-Wesley, 2003.

Bildquellen:

- Abbildung 1: Screenshot von <http://demo.magentocommerce.com/customer/account/login/> [Stand 17.05.2015]
- Abbildung 2: www.amazon.de/Domain-Driven-Design-Tackling-Complexity-Software/dp/0321125215 [Stand 17.05.2015]

Grafische Visualisierung von Diagnosedaten

Daniel Secker*, Werner Zimmermann, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Die Anzahl der diagnosefähigen Steuergeräte in Fahrzeugen ist in den letzten Jahrzehnten stetig gestiegen: Während 1985 nur drei in einem Sportwagen verbaut wurden, sind es mittlerweile über 50 [1]. Dies ist unter anderem darauf zurückzuführen, dass immer mehr Elektronik verbaut wird und die Vernetzung zunimmt (siehe Abb. 1), um etwa bessere Getriebesteuerungen oder intelligentere Assistenzsysteme zu ermöglichen. Es ist auch kein Ende dieses Trends in Sicht: Durch Innovationen wie das ‚Connected Car‘ oder sogar autonom fahrende Autos werden wahrscheinlich noch mehr ECUs benötigt.

Mehr Funktionen bedeuten allerdings auch mehr Fehlerpotential und somit mehr Bedarf an Überwachung. Der Einsatz von aufwändigen Diagnosesystemen ist damit unumgänglich. Fahrzeugdiagnose soll eine möglichst kostengünstige Reparatur bzw. Wartung ermögli-

chen. Außerdem sollen sich Fehler ohne tiefes Expertenwissen sowohl lokalisieren als auch beheben lassen, idealerweise über alle Fahrzeuge und Varianten hinweg.

Damit der Datenaustausch geregelt ablaufen kann, werden verschiedene Standards eingesetzt. Auf Kommunikationsebene kommt das Unified Diagnostics Services Protokoll zum Einsatz. Darüber hinaus wird ODX (Open Diagnostic Data Exchange) verwendet. Das auf XML basierende Format sorgt für einen standardisierten Diagnose-Datenaustausch zwischen Automobil-, Steuergeräte und Toolhersteller. Der strukturelle Aufbau der Diagnosebotschaften in ODX ist jedoch herstellerspezifisch, da jeder Hersteller verschiedene Anforderungen hat. So können sich z.B. die verwendeten Parametertypen unterscheiden. Die Strukturen für einen Hersteller werden in sogenannten Autorenrichtlinien definiert.

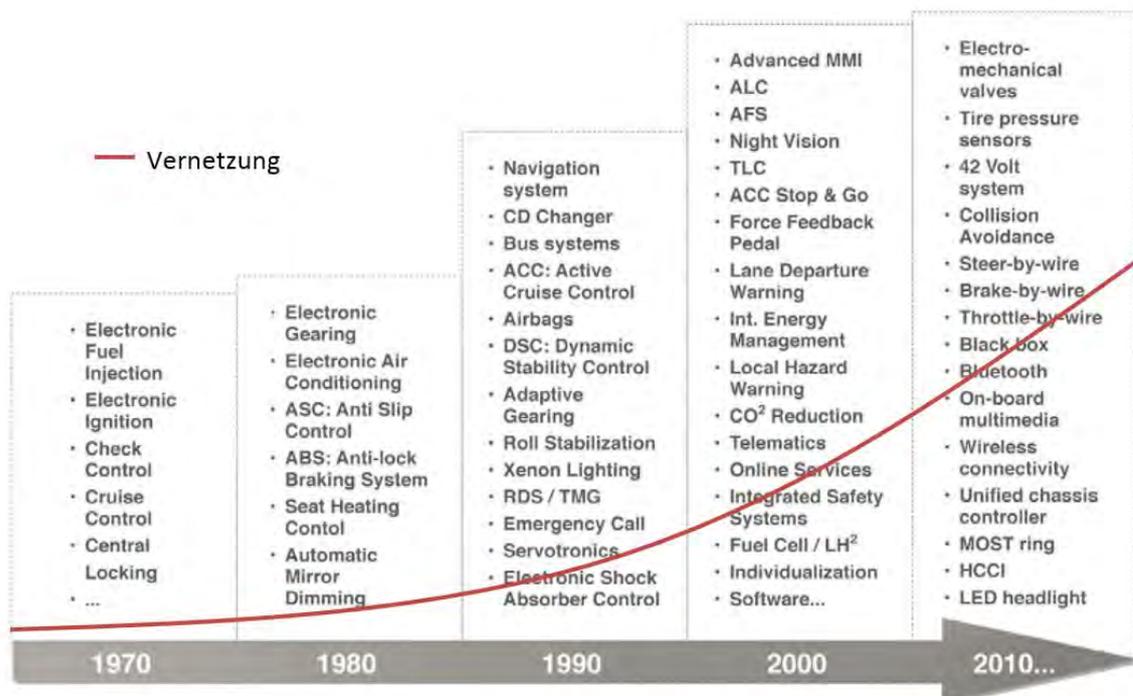


Abbildung 1: Vernetzung im Fahrzeug

*Diese Arbeit wurde durchgeführt bei der Firma Vector Informatik GmbH, Stuttgart

Thema dieser Arbeit ist die automatische Generierung von Diagrammen zur Definition von Servicestrukturen aus ODX-Daten für die Autorenrichtlinien. Dabei sollen im ersten Schritt aus den übergebenen Daten die erforderlichen Parameter und deren Beziehungen zueinander ermittelt werden. Danach wird ein Diagramm erstellt, welches die gefilterten Daten mittels Containern und visualisierten Beziehungen in einer GUI darstellt (siehe Abb. 2).

Bei der Implementierung ist ein modularer Aufbau wichtig, damit später etwa andere Datenformate verwendet können oder unterschiedliche Zieldarstellungen möglich sind. Bei dem Parsen der ODX Daten sollen diese in ein Service Model umgewandelt werden: Ein Service besteht fast immer aus einem Request und einer Response. Die darin enthaltenen Parameter können aber auch Referenzen auf ande-

re komplexe Strukturen enthalten. Der Parser muss also der Verschachtelung folgen und so eine Hierarchie aufbauen.

Das Design (Farbgebung, Layout ...) der Diagramme soll anpassbar sein, um den herstellerspezifischen Anforderungen zu genügen. Das Layout soll möglichst schon ideal erstellt werden, sodass weitestgehend keine Nachbearbeitung notwendig ist. Dabei müssen die Beziehungen berücksichtigt werden, um Überkreuzungen zu vermeiden. Auch die Wegfindung der Verbindungen soll durch geeignete Routen dabei helfen ein übersichtliches Diagramm zu generieren. Falls aber Änderungen im Layout in der Applikation vorgenommen wurden (Container wurden neu platziert), soll es möglich sein bei einer neuen Generierung der Diagramme das modifizierte Layout zu verwenden.

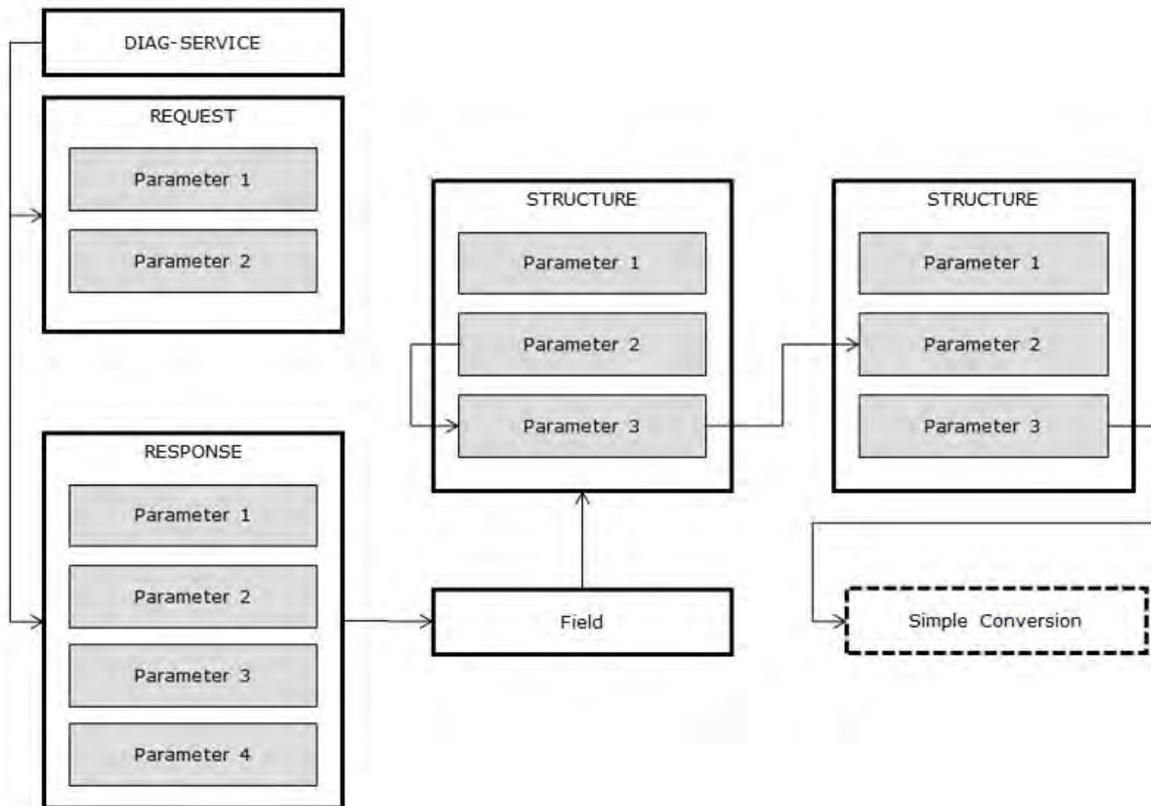


Abbildung 2: Diagramm eines Services

[1] Vector Informatik

Bildquellen:

- Abbildung 1: The Growth of automotive electronics in APAC, Andrew Chong, Infineon
- Abbildung 2: Vector Informatik

Entwicklung eines radarbasierten Notbremsassistenten für ein „Neighborhood Electric Vehicle“

Tobias Thill*, Reiner Marchthaler, Werner Zimmermann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Auf den Automobilmessen werden Fahrzeuge mit immer mehr Fahrerassistenzfunktionen vorgestellt. Diese Fahrzeuge können beispielsweise schon Verkehrsschilder erkennen, die Spur halten, das Fahrzeug stabilisieren, den Abstand zum vorderen Fahrzeug halten und vieles mehr. Dadurch werden dem Fahrer viele Aufgaben abgenommen oder erleichtert. Der Trend geht ganz klar zum autonomen Fahren, was auch die berühmte Mercedes-Benz-Fahrt im August 2013 mit dem S 500 INTELLIGENT DRIVE gezeigt hat. Die Ingenieure von Mercedes-Benz sind die rund 100 Kilometer lange Strecke der ersten Autofahrt von Bertha Benz aus dem Jahre 1888 bei alltäglicher Verkehrsdichte voll autonom nachgefahren. Trotz dieser Pionierleistung gibt es in Sachen autonomes Fahren noch viel zu tun, um in allen Verkehrssituationen autonom fahren zu können. Dabei sind die Sicherheit und die Vermeidung von Unfällen die wichtigsten Meilensteine. [1]



Abbildung 1: Neighborhood Electric Vehicle

Der neueste Trend sind hierbei Notbremsassistenten, die den Fahrer vor einem möglichen Unfall warnen und teilweise auch eine vollautonome Vollbremsung einleiten, um einen Unfall zu vermeiden oder dessen Unfallfolgen zu verringern.

In dieser Bachelorarbeit soll ein radarbasierter Notbremsassistent für ein Neighborhood Electric Vehicle (NEV) entwickelt werden. Das

NEV ist ein gewöhnliches, elektrisch betriebenes, Golfcart vom Modell Freedom RXV der Firma E-Z-GO, das bereits im Rahmen mehrerer Abschlussarbeiten modifiziert wurde. So wurde es beispielsweise um eine by-wire Lenkung und verschiedene Sensoren wie Kamera, Ultraschall oder Radar erweitert. Gesamtziel der NEV-Entwicklung ist es, ein vollautomatisiertes Fahrzeug zu entwickeln, das in einem abgegrenzten Bereich, wie z. B. auf einem Campus oder Golfplatz, autonom fahren kann. Diese Bachelorarbeit soll mit der Fahrerassistenzfunktion „Notbremsung“ einen Teil dazu beitragen. Dazu soll das Fahrzeugvorfeld mit Hilfe des ARS 300 Radars von Continental analysiert und automatisch eine Notbremsung durchgeführt werden, falls ein Hindernis auf der Fahrbahn erkannt wird und eine Kollision droht. Jegliche Art von Kollision soll vermieden werden.

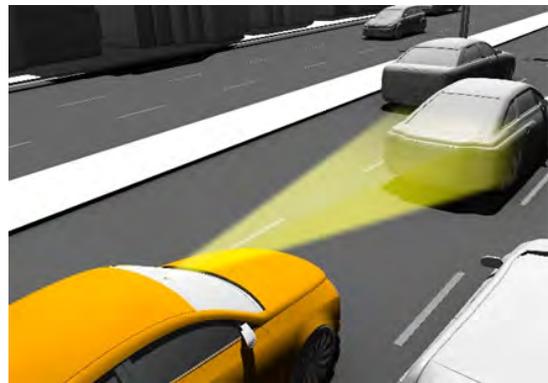


Abbildung 2: Vorfeldscan_Radar

Der Notbremsassistent soll mittels eines Matlab/Simulink Modells realisiert werden. Dazu wird zunächst ein Laptop mit dem NEV über ein CAN-Interface verbunden, um die Messwerte des Radars über CANoe in Matlab/Simulink zu übertragen, wo sie dann zur Laufzeit gefiltert und weiterverarbeitet werden. So können die Messwerte und Ergebnisse der Steuerungen und Regelungen in Echtzeit mitverfolgt werden, um Probleme und Fehler direkt zu erkennen. Erst wenn der Notbremsassistent voll funktionsfähig ist, wird er auf ein Steuergerät der GIGATRONIK Stuttgart GmbH – die GIGA-

*Diese Arbeit wurde durchgeführt bei der Firma GIGATRONIK Stuttgart GmbH, Stuttgart

BOX – geflasht. Der Notbremsassistent ist nach dem EVA-Prinzip (Eingabe Verarbeitung Ausgabe) aufgebaut. Es werden Messdaten vom Radar, wie der Abstand des vorausliegenden Objektes, oder Messdaten vom NEV, wie die Radumdrehung und damit die Geschwindigkeit, eingelesen und daraus berechnet, ob eine Kollision zustande kommen kann. Droht eine Kollision, wird ein Bremsignal mit einer entsprechenden Bremsleistung zur Vermeidung der Kollision ausgegeben. Da die Einlese- und Ausgabefunktionen über CANoe bzw. der GIGABOX unterschiedlich sind, kann durch die EVA-Unterteilung der mit CANoe am Laptop getestete Verarbeitungsblock unverändert für die GIGABOX übernommen werden. Es muss lediglich das Einlesen von Sensorsignalen und die Ausgabe des Bremsignals angepasst werden.

Da das NEV noch nicht autonom fahren kann, wird dem Notbremsassistenten auch eine manuelle Funktion hinzugefügt. Dabei kann man ihn per Knopfdruck aktivieren und mittels eines 6-stufigen Drehschalters den Sicherheitsabstand variabel einstellen. Dabei steht jede Stufe für einen weiteren Meter Abstand zu dem Objekt, vor dem das NEV zum Stehen kommen soll. Der variable einstellbare Sicherheitsabstand hat einen psychologischen Hintergrund, da vermutlich nicht alle

der Technik vollständig vertrauen. Dementsprechend kann man den Sicherheitsabstand etwas größer wählen und merkt so schon deutlich früher, dass die Kollisionsgefahr erkannt wurde und kommt mit dem für sich vertraulichen Sicherheitsabstand zum Stehen. Des Weiteren wird auf dem im NEV verbauten Display kontinuierlich angezeigt, ob der Notbremsassistent aktiviert ist und ob vorausliegende Objekte erkannt werden. Es ist nicht nur wichtig, dass Kollisionen vollständig vermieden werden, sondern auch, dass der Fahrer dem System vertraut und es demzufolge auch einsetzt.



Abbildung 3: GIGABOX

[1] www.mercedes-benz.de

Bildquellen:

- Abbildung 1: GIGATRONIK Stuttgart GmbH
- Abbildung 2: Continental Reifen Deutschland GmbH
- Abbildung 3: GIGATRONIK Stuttgart GmbH

Analyse eines Testsystems für CAN- und FlexRay-Interfaces sowie Implementierung und Aufbau eines Nachfolgesystems einschließlich Migration aller Funktionen

Yannick Uhlmann*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Ein integraler Bestandteil der Produktentwicklung ist die Qualitätssicherung. „Qualitätssicherung ist die Summe aller Maßnahmen, um konstante Produktqualität sicherzustellen“

[1]. Durch sie wird auch im Bereich Informatik die bestmögliche Funktionalität eines Systems erreicht.

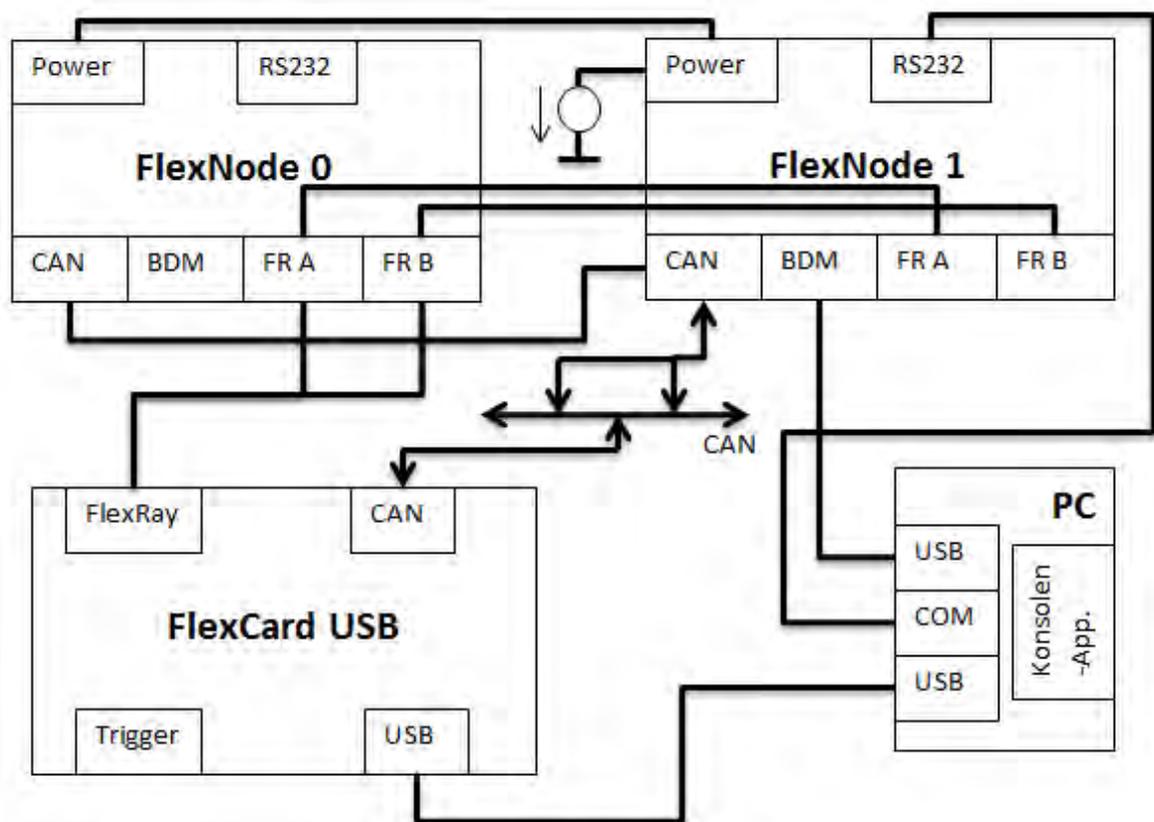


Abbildung 1: Testaufbau mit FlexNodes

Realisiert wird die Qualitätssicherung durch das Testen eines Systems in seinen verschiedenen Entwicklungsstadien. Im Bereich Bussysteme muss daher das Verhalten einzelner Kommunikationsteilnehmer in einem bereits etablierten System getestet werden. Ziel ist es, einen möglichst einfachen Testaufbau zu realisieren, welcher eine möglichst effiziente Anbindung der zu testenden Teilnehmer an das Testsystem zulässt. Die von der

Firma Eberspächer Electronics hergestellten FlexCards sind Interfaces, welche die Analyse und Überwachung von CAN- und FlexRay-Systemen mit Hilfe eines PCs erlauben. Sie werden in einer wie zuvor beschriebenen Umgebung getestet. Der Testaufbau wird zunächst mit zwei Kommunikationsknoten in Betrieb genommen, die die Basis des Testsystems bilden. Um eine FlexCard zu testen, wird diese mit einem PC verbunden und als drit-

*Diese Arbeit wurde durchgeführt bei der Firma Eberspächer Electronics GmbH & Co. KG, Göppingen

ter Knoten zum Testsystem hinzugefügt. Der PC steuert mit Hilfe einer Konsolenapplikation das Verhalten der zu testenden FlexCard. Außerdem kommuniziert er mit den anderen Teilnehmern im Testsystem, damit diese in der Testreihe entsprechendes Verhalten aufweisen. Bisher wurde dieser Testaufbau mit Hilfe von zwei FlexNodes als Knoten realisiert. Bei den FlexNodes handelt es sich jedoch um ein abgekündigtes Produkt, welches nur noch in reduzierter Stückzahl zur Verfügung steht.



Abbildung 2: FlexCard PMC II

Außerdem kommt der auf dem FlexNode verbaute Mikrocontroller (HCS12) bei einigen Testreihen an seine Leistungsgrenzen. Aus diesem Grund soll nun ein neues Testsystem aufgebaut werden. Das ebenfalls von Eberspächer Electronics entwickelte FlexX-Con midget soll dabei die FlexNodes im zukünftigen Aufbau als Testpartner ablösen. Es handelt sich um eine multifunktionale Kontrolleinheit, die als Gateway, zur Restbus-Simulation oder als Datenlogger eingesetzt werden kann. Außerdem liefert es mit seinen beiden Communication-Controllern bereits beide benötigte Knoten des neuen Testaufbaus [2]. Um den Umstieg von den FlexNodes auf das FlexXCon midget zu realisieren, muss das Verhalten des neuen Testaufbaus um einzelne Testfälle erweitert werden. Anschließend wird das Verhalten des neuen Aufbaus während des Tests analysiert und mit dem des alten Aufbaus verglichen. Zur Feststellung eines eventuellen Fehlverhaltens werden die Pakete, die über den Bus des Testsystems verschickt werden, einzeln untersucht.

Stellt man auf diese Weise Verhaltensunterschiede zwischen den beiden Aufbauten fest, liegt der Fehler entweder in der Firmware des neuen Testpartners oder in der Testapplikation, über welche die FlexCard vom PC aus angesteuert wird. Veränderungen an der Firmware werden innerhalb eines bereits bestehenden Frameworks durchgeführt. Die veränderte Firmware lässt sich mit Hilfe des TFTP-Servers auf dem FlexXCon midget einfach per Ethernet auf dieses übertragen. Des Weiteren ist die Testsoftware, welche die FlexCards steuert, ebenfalls zu erneuern. Dazu soll von der seither verwendeten und in C++ erstellten Konsolenapplikation auf eine in C++ mit dem Qt-Framework entwickelte GUI-Applikation umgestiegen werden. In Zukunft sollen die einzelnen Testfälle nicht mehr fest in den Code der Applikation integriert sein, sondern als .dll-Dateien von der neuen Applikation eingebunden werden. Dieser Ansatz erhöht die zukünftige Erweiterbarkeit um neue Testfälle drastisch.

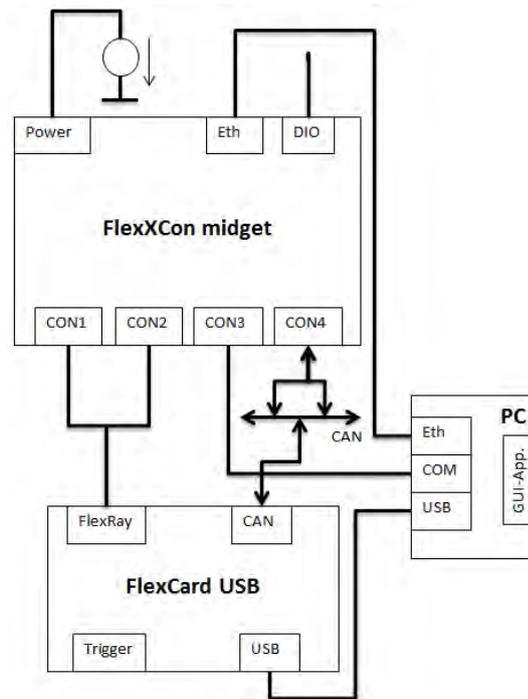


Abbildung 3: Testaufbau mit FlexXCon midget

[1] Wikipedia – Qualitätssicherung <http://de.wikipedia.org/wiki/Qualit%C3%A4tssicherung> Zugriff: 12.05.2015
 [2] Eberspächer Electronics GmbH & Co. KG: FLEXCON MIDGET INSTRUCTIONS FOR USE, 2013

Bildquellen:

- Abbildung 1,3: Yannick Uhlmann
- Abbildung 2: Eberspächer Electronics

Konzipierung und Realisierung des Moduls zur Verarbeitung von Einzelsignalen einer Nachricht für einen CAN-Protokollstack zur Kommunikation zwischen Steuergeräten in Fahrzeugen unter Berücksichtigung einer zeit- oder ereignisorientierten Übermittlung der Signale

Watthana Vilaysouk*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Vor ungefähr zwei Jahrzehnten hat sich das Controller Area Network (CAN) als Kommunikationssystem für Steuergeräte in der Automobilindustrie durchgesetzt und sich seitdem in vielfältigen Anwendungen bewährt. Eine zuverlässige und robuste Arbeitsweise und zugleich kostengünstige Realisierungsmöglichkeiten haben wesentlich zum Erfolg des Systems beigetragen. Noch heute ist in Kraftfahrzeugen CAN als Kommunikationssystem nicht wegzudenken. Die Funktionalitäten reichen vom Versenden und Empfangen von Prozessdaten oder Diagnoseinformationen bis hin zu Mess- und Kalibrierdaten.

Betrachtet man die Softwareimplementierung des CAN in einem Steuergerät, so könnte eine mögliche Softwarearchitektur als Schichtenmodell wie folgt aussehen: Ganz oben in dem Modell befindet sich die Applikationsschicht, in der die eigentliche Anwendung angesiedelt ist. Darunter liegen die Schichten, die die Kommunikation über CAN ermöglichen. Zum einen der Hardware-Abstraktion-Layer und zum anderen die darüber liegende Treiberschicht, welche beide die Hardware bzw. die Zugriffsfunktionen abstrahieren, und darüber die Schicht der High-Level-Protokolle

und sonstigen Automatismen, die eine weitere Abstraktion der Funktionalitäten darstellt [1]. In Abbildung 1 ist eine beispielhafte Architektur dargestellt.

Die Aufgabe der vorliegenden Bachelorarbeit ist es, das Signal-Manager-Modul der in Abbildung 1 dargestellten Architektur zu entwickeln. Die Architektur des Moduls lehnt sich an den Interaction-Layer des Standards OSEK COM an, dessen Aufgabe die Bereitstellung einer API für die Anwendung ist, über die Daten versendet und empfangen werden. Des Weiteren sind die Nachrichteninterpretation, das zyklische und ereignisbasierte Versenden von Nachrichten, die *Byte Order Conversion* und das Puffern von CAN-Nachrichten zusätzliche Funktionalitäten dieser Schicht [2].

Ein wesentlicher Bestandteil des Signal-Manager-Moduls ist eine signalorientierte Datenaufbereitung und die Übermittlung der Daten. Ändert sich ein Datenwert, hier auch Signal genannt, in der Applikationsschicht, so wird die entsprechende CAN-Nachricht ermittelt und zum Versenden weitergeleitet. Umgekehrt werden empfangene Nachrichten in ihre Signale aufgeteilt und diese Signale der Applikationsschicht zur Verfügung gestellt.

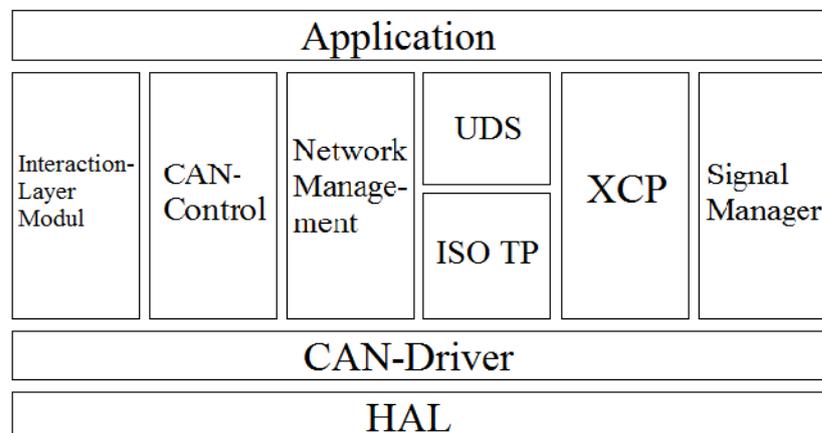


Abbildung 1: Softwarearchitektur eines Basissteuergeräts hinsichtlich CAN-Anbindung

*Diese Arbeit wurde durchgeführt bei der Firma Eberspächer Automotive Controls GmbH & Co. KG, Esslingen

Warteschlangen (Queues) zum Puffern der weiterzuleitenden Nachrichten werden ebenfalls berücksichtigt. Das Beispiel für den Aufbau einer CAN-Nachricht und die Unterteilung des Datenfeldes in Signale ist in Abbildung 2 dargestellt.

Darüber liegt ein weiterer Schwerpunkt der vorliegenden Arbeit auf den zeitlichen Aspekten der Übermittlung der Daten. In Abhängigkeit von den verschiedenen möglichen Sende-

typen werden die Botschaften entweder ausschließlich zyklisch, ausschließlich ereignisgesteuert oder zyklisch und ereignisgesteuert weitergeleitet. Empfangene Nachrichten werden zudem auf rechtzeitiges Ankommen über eine Timeout-Zeit geprüft.

Weitere Aufgaben sind die *Byte Order Conversion* und die Umrechnung der Signale in physikalische Größen.

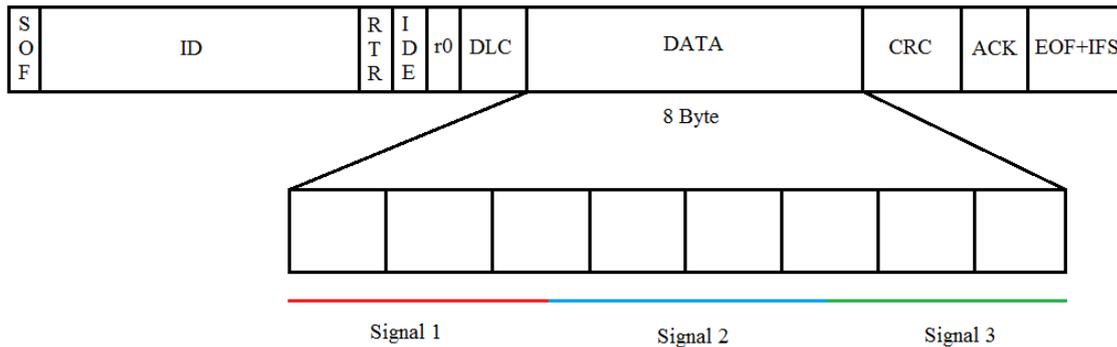


Abbildung 2: CAN-Nachricht (CAN 2.0A) mit Aufteilung des Datenfeldes in einzelne Signale

- [1] F. Rembor, „SRS – Basisteuergerät 1 – Basisentwicklung (0.3.1),“ Eberspächer Automotive Controls GmbH & Co. KG, Esslingen, 2015.
 [2] M. Z. H. G. P. G. A. Di Natale, „Understanding and Using the Controller Area Network Communication Protocol,“ Springer, 2012

Bildquellen:

- Abbildung 1,1: Eigene Darstellung

Umfeldmodellierung durch Belegungsgitter auf Basis von Ultraschallsensorik

Timo Wascheck*, Jürgen Koch

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Heutzutage finden immer mehr Fahrerassistenzsysteme ihren Weg in die Serienreife, um sie somit in Fahrzeugen nutzen zu können. Diese Fahrerassistenzsysteme erleichtern das Fahren und reduzieren dessen Gefahren. Der Aufbau von Fahrerassistenzsystemen ist meist derselbe: über Sensorik werden Messdaten empfangen, auf Grundlage derer eine Umfeldmodellierung realisiert wird. Ergebnis der Umfeldmodellierung ist ein digitales Abbild des Fahrzeugumfeldes. Mithilfe dieses Umfeldmodells ist es möglich, eine Situationsanalyse durchzuführen, um zu erkennen, ob in das Fahrgeschehen eingegriffen werden muss oder nicht. Da durch die Situationsanalyse bekannt ist ob und wo eingegriffen werden muss, kann durch die Aktionsplanung bestimmt werden, wie eingegriffen werden soll. Dieser Plan wird schlussendlich umgesetzt, indem die einzelnen Aktoren angesteuert werden.



Abbildung 1: Skizze Versuchsträger mit Ultraschallsensorpositionen

Das öffentlich geförderte Projekt UR:BAN [1] (Urbaner Raum: Benutzergerechte Assistenzsysteme und Netzmanagement), setzt sich mit Fahrerassistenzsystemen im urbanen Raum auseinander, um vor allem die Sicherheit im städtischen Verkehr zu erhöhen. Hierzu muss wie bereits erwähnt, eine Umfeldmodellierung stattfinden, dessen Komplexität jedoch deutlich höher ist, als die Modellierungen außerorts oder auf Autobahnen. Dies ist der Fall, da viel mehr Faktoren Einfluss auf das System haben können, wie zum Beispiel Fußgänger oder Radfahrer, mit denen auf Autobahnen im Normalfall nicht zu rechnen ist. Des Weiteren ist es wichtig, auf ein, für diese Umgebung geeignetes und stabiles Umfeldmodell zurück-

greifen zu können, wenn man im Stadtverkehr eine Situationsanalyse und die daraus folgende Aktionsplanung durchführt, da eine Fehlentscheidung schnell Leben kosten kann.

Da die Entwicklung von Fahrerassistenzsystemen auch ein Heranarbeiten an das autonome Fahren ist und die Qualität der Umfeldmodellierung im urbanen Raum sehr wichtig ist, ist es notwendig, dass das Umfeld nicht nur auf Basis einer Sensorik modelliert wird, sondern redundant mit verschiedenen Arten von Sensorik.

Für die Umfeldmodellierung um das Fahrzeug wurde ein Belegungsgitter genutzt. Dieses Belegungsgitter abstrahiert die reale Umgebung, durch eine Abbildung auf ein zweidimensionales Gitternetz. Dabei besitzt jede Gitterzelle einen Wert, welcher die Wahrscheinlichkeit angibt, dass diese Zelle belegt oder frei ist.

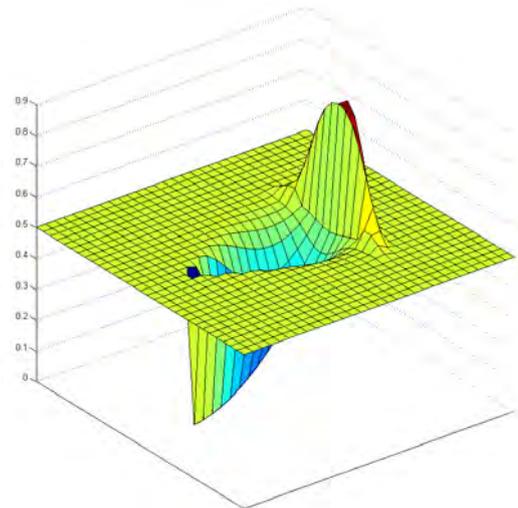


Abbildung 2: Belegungswahrscheinlichkeiten durch Sensormodell

Um diese Umfeldmodellierung für den Nahbereich zu realisieren, soll in dieser Arbeit ein in einem Auto verbautes Ultraschallsystem mit 12 Sensoren, wie es in Abbildung 1 zu sehen ist, genutzt werden. Die Daten dieser Ultraschallsensoren sollen verarbeitet und aufbereitet werden, um sie einem Plugin zur

*Diese Arbeit wurde durchgeführt bei der Firma Robert Bosch GmbH, Renningen

Verfügung zu stellen. Dieses Plugin, dessen Entwicklung Teil dieser Arbeit ist, soll in ein bestehendes Umfeldmodell eingebunden werden. Über definierte Schnittstellen des Plugins muss dabei ermöglicht werden, dass das Umfeldmodell Informationen über Belegungswahrscheinlichkeiten im Nahbereich um das Fahrzeug abrufen kann. Die Umsetzung dieser Arbeit fand innerhalb der Softwareumgebung ADF (Automotive Data and Time-Triggered Framework) statt. Dabei wurden die Sensordaten, welche auf dem CAN-Bus des Fahrzeugs übertragen werden, gefiltert und an eine Vorverarbeitung übergeben. Innerhalb der Vorverarbeitung wurden verschiedene Sensormodelle genutzt. Diese Modelle berechnen anhand der Sensordaten Wahrscheinlichkeitswerte für die Gitterzellen der Fahrzeugumgebung. Solche Belegungswahrscheinlichkeiten für einen Sensor sind in Abbildung 2 zu sehen.

Die durch die Sensormodelle erzeugten Daten werden an ein eigens für diese Arbeit programmiertes Plugin übergeben. Über genormte Schnittstellen dieses Plugins werden von der Software, welche zur Umfeldmodellierung dient, Werte um das Fahrzeug abgegriffen um somit das Umfeldmodell zu generieren. Die Visualisierung solch eines Umfeldmodells ist in Abbildung 3 zu sehen. Zukünftige Arbeiten könnten auf diesen Modellen aufbauen und durch Tests optimale Werte für verschiedene einstellbare Parameter finden. Beispiele hierfür sind der Sichtbereich der Ultraschallsensoren oder die maximale Reichweite, für die mit hoher Wahrscheinlichkeit fehlerfreie Messwerte generiert werden. Des Weiteren wäre eine redundante Umfeldmodellierung im Nahbereich sinnvoll. Dies könnte zum Beispiel erreicht werden, indem zusätzlich zur

Ultraschallsensorik, Kameras zur Umfeldüberwachung eingesetzt werden. Eine Sensordatenfusion der entsprechenden Messwerte sollte im Rahmen einer weiterführenden Arbeit untersucht werden.



Abbildung 3: Visualisierung Umfeldmodellierung

[1] <http://urban-online.org>, Abruf am 03. Mai 2015

Bildquellen:

- Abbildung 1–3: Timo Wascheck

Entwurf und Implementierung eines Herzfrequenz Koinzidenzdetektor Testsystems

Clemens Weißenberg*, Reiner Marchthaler, Karlheinz Höfer

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Eine noch-schwangere Frau bei der Geburt im Kreißsaal. Die Herzfrequenz der Mutter wird über den Sauerstoffgehalt im Blut mit einem Fingersensor gemessen. Die Herzfrequenz des Kindes über Ultraschall mit einem Bauchgurt-sensor ermittelt. Beide Signale werden zur Verarbeitung an einen Überwachungsmonitor gesendet und dort zur medizinischen Überwachung angezeigt. Bei der Geburt rutscht der Bauchgurt über eine Arterie der Mutter: Beide Sensoren messen nun die Herzfrequenz der Mutter. Da weiterhin zwei Herzfrequenzen gemessen werden fällt nicht auf, dass die Herzfrequenz des Kindes nicht weiter überwacht wird. Im schlechtesten Fall könnte das Kind unbemerkt sterben. Dies ist ein Szenario aus der Praxis das leider viel zu häufig eintritt. Um dem zu entgehen wurde der Koinzidenzdetektor entwickelt.



Abbildung 1: Herzfrequenzüberwachung bei Frau und Kind

Der Koinzidenzdetektor soll dem oben beschriebenen Szenario entgegen wirken. Er ist ein Softwaretool, welches auf dem Überwachungsmonitor im Hintergrund läuft. Beim Überwachungsmonitor kommen Herzfrequenzsignal von Mutter und Fötus zusammen. Daher kann der Koinzidenzdetektor einen Vergleich, mittels verschiedener Merkmale, der beiden Signale treffen. Bei Übereinstimmung der Merkmale erkennt er die Deckungsgleichheit der Signale. Vom Überwachungsmonitor folgt eine visuelle und akustische Warnung auf die das Krankenhauspersonal reagieren kann. So kann der Worst Case verhindert werden.

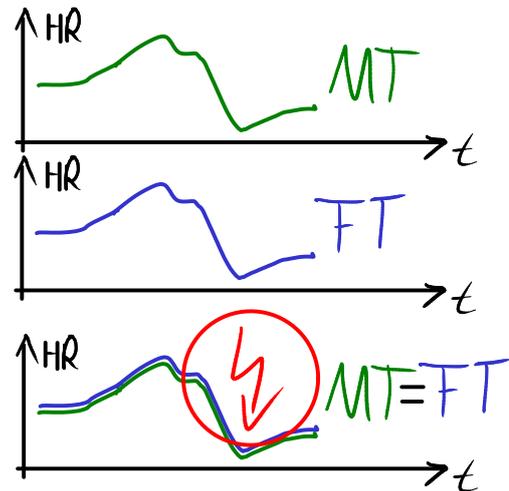


Abbildung 2: Funktionsweise des Koinzidenzdetektors

Bei der Messung der maternalen und der fetalen Herzfrequenz kommt es jedoch zu Laufzeitverzögerungen. Grund dafür sind die physiologischen Verzögerungen im Herzkreislauf, die unterschiedlichen, angewandten Messtechniken als auch die Datenübertragung. Vor allem bei drahtlosen Verbindungen kommen lange Verzögerungen zustande. Diese Verzögerungen können dazu führen, dass eine signifikante Phasenverschiebung zwischen den Signalen auftritt. Überspitzt gesagt kommt Signal eins mehrere Sekunden später als Signal zwei an. Ein direkter Vergleich der Signale wird dadurch erschwert.

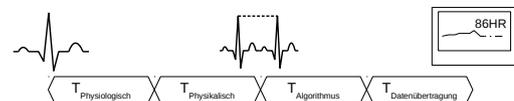


Abbildung 3: Zeitverzögerung bei Herzfrequenzsignalen

Ziel dieser Bachelorarbeit ist es für den erwähnten Koinzidenzdetektor ein Testsystem zu entwerfen und zu implementieren. Es soll möglich sein verschiedene Signale und Sensoren zu simulieren um die Grenzen des Koinzidenzdetektor zu ermitteln und mögliche

*Diese Arbeit wurde durchgeführt bei der Firma Philips Medizin Systeme, Böblingen

Fehlerquellen zu eliminieren. Um einen Patienten möglichst realitätsnah zu imitieren, werden analoge Signale über einen Digital-Analog-Wandler (als D/A gekennzeichnet) generiert. Über Herzfrequenzsimulatoren können die im Krankenhaus üblichen Sensoren,

auch Transducer genannt, die aktuelle Herzfrequenz messen und an den Überwachungsmonitor (als BM gekennzeichnet) leiten. So können alle in diesem System auftretenden Verzögerungen mit berücksichtigt werden.

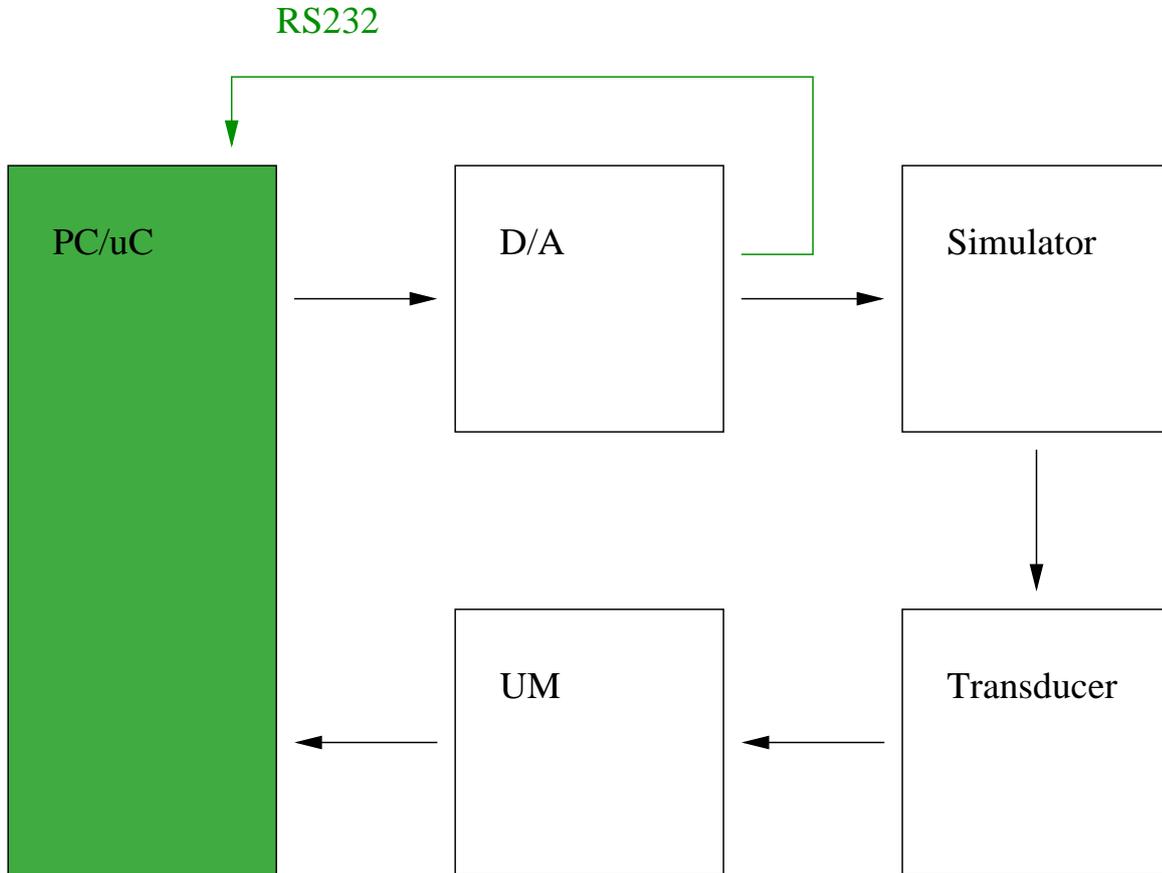


Abbildung 4: Entwurf Systemaufbau

Bildquellen:

- Abbildung 1: <http://www.philips.de/healthcare-product/HC865071/avalon-fm50-fetalmonitor>
- Abbildung 2-4: Clemens Weißenberg

Extreme Programming

Thomas Wild*, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2015

Extreme Programming (Kurzform: XP) nach Kent Beck ist der erste bedeutende, konkrete Ansatz der agilen Softwareentwicklung und Vorgänger des agilen Manifestes. Extreme Programming eignet sich für kleine bis mittelgroße Teams, deren Arbeit vagen oder sich ständig ändernden Anforderungen unterliegt [1].

Erstmalig eingesetzt wurde Extreme Programming beim sogenannten C3-Projekt der Firma Chrysler. Hierbei wurde ein für gescheitert erklärtes Projekt im Jahre 1997 mit den von Kent Beck entwickelten Methoden wieder aufgenommen [2]. Aus den in diesem Projekt gewonnenen Kenntnissen entstand das XP Manifest von Kent Beck, welches 1999 veröffentlicht wurde.

Charakteristisch für XP ist, dass keine Requirements im klassischen Sinne aufgestellt werden, sondern dass der Kunde sogenannte User Stories schildert, welche durch ihn eine Priorisierung erhalten und gemeinsam mit den Entwicklern abgestimmt werden.

Der Fokus agiler Methoden liegt auf einer engen Zusammenarbeit der Entwickler mit dem Kunden [3]. Aus diesem Grund muss immer mindestens ein Mitarbeiter des Kunden verfügbar sein.

Da möglichst viel Overhead im Projekt entfallen soll, wird ein Projekt nicht mit einer aufwendigen Analyse, sondern mit dem Umsetzen der wichtigsten Funktionen gestartet. Die Stories des Kunden werden in kleinere Tasks (Aufgaben) gesplittet. Programmierer nehmen sich dieser Tasks an und setzen diese um (engl. **iteration planning**) [1]. Am Ende eines jeden Tages steht ein lauffähiges System (engl. **continuous integration**) [1]. Dabei verfolgt XP die Vorgehensweise, in Paaren zu programmieren (engl. **pair programming**) [1]. Dies wird nach dem Prinzip, dass zuerst getestet werden muss (engl. **tests first**), umgesetzt. Das bedeutet, dass zuerst die Tests für die Schnittstellen einer Komponente geschrieben werden und dann überprüft wird, ob die Tests erfolgreiche Ergebnisse liefern, bevor die Komponente schließlich implementiert wird [1].

Programmierteams sollen räumlich zusammensitzen, sich jedoch gegenseitig nicht durch hohe Lautstärke behindern. Der erstellte Pro-

grammcode gehört allen Programmierern. Somit kann jeder Entwickler Änderungen durchführen, selbst wenn der Code nicht von ihm selbst stammt. (engl. **collective ownership**) [1].

Die Architektur wird laufend durch sogenanntes **Refactoring** überdacht. Refactoring zielt auf eine Umstrukturierung des Codes ab, wenn für die gleiche Funktionalität eine einfachere Architektur gefunden wird. Die Einfachheit des Codes zeigt sich auch dadurch, dass nicht an Erweiterungen oder eine Wiederverwendung von Code gedacht wird (kein Gold Plating). Es wird stets die minimale Lösung umgesetzt, die den Ansprüchen des Kunden gerecht wird [1]. Besonderheiten von XP sind neben engem Kontakt mit dem Kunden und stetigem Feedback auch schnelle Ergebnisse und ein Fokus auf der Architektur durch Refactoring im Gegensatz zu üblichen prototypischen Vorgehensweisen.

Der Name Extreme Programming mag im erstem Moment verwirrend sein, denn keine der angesprochenen Punkte hat den Anschein, „extrem“ zu sein. Doch bei der Begutachtung, wie XP umgesetzt wird, wird klar, dass Extreme Programming alt bewährte Prinzipien und Verfahren der Softwareentwicklung in extremer Form umsetzt:

- **Code Reviews:** Durch Programmieren in Paaren wird der Code ständig begutachtet.

- **Tests:** Es wird permanent getestet. Sei es von Programmierern durch Komponententests (engl. unit tests) oder durch den Kunden durch Funktionstests.

- **Design:** Die Architektur wird durch sogenanntes Refactoring angepasst und vereinfacht.

- **Einfachheit:** Es wird immer nur die geforderte Funktionalität umgesetzt und zwar mit geringst möglichem Aufwand.

- **Integrationstests:** Es wird täglich mehrmals integriert und getestet (fortlaufende Integration).

- **Iterationszeiten:** Diese sind extrem kurz. Während bei spezifikationsorientierten Vorgehensmodellen die Planungsphase alleine oft über Jahre hinweggehen kann, dauert eine komplette Iteration in XP zwischen einer bis maximal drei Wochen [1].

*Diese Arbeit wurde durchgeführt bei der Firma IT-Designers GmbH, Esslingen

Diese aufgelisteten Vorgehensweisen sollen **Risiken** in der Softwareentwicklung **minimieren**. Durch kurze Iterationszeiten wird eine größere Terminverzögerung oder gar ein Projektabbruch aus Zeitgründen vermieden, da schnell auf Probleme reagiert werden kann. Bei kleinen Schritten zum Ziel ist eine mögliche Projektverzögerung gering. Dadurch wird dem Kunden zu keinem Zeitpunkt gesagt, dass sich zum Beispiel der nächste Schritt ein halbes Jahr verzögert. Neben dem schnellen Aufnehmen von Änderungen dient der laufende Kundenkontakt ebenfalls dazu, die Geschäftsziele nicht zu verfehlen.

Auch aus wirtschaftlicher Sicht bietet XP Vorteile: Kosten für Änderungen werden auf Grund agiler Vorgehensweisen gering gehalten.

XP unterscheidet sich von spezifikationsorientierten Vorgehensmodellen der Softwareentwicklung wie dem Wasserfall- oder dem V-Modell unter anderem in folgenden Punkten:

- **Kurze Zyklen**, welche frühzeitige, konkrete und fortwährende Feedbacks ermöglichen.
- Einen **inkrementellen Ansatz**. Auf den aktuellen, stets funktionsfähigen Stand werden stetig neue Leistungsmerkmale aufgesetzt. Diese Leistungsmerkmale werden so einfach wie möglich umgesetzt.
- Die Fähigkeit, eine **flexible Planung** zu betreiben und somit schnell auf geänderte Anforderungen reagieren zu können.
- XP wird von **Tests** angetrieben, welche von Programmierern und Kunden geschrieben werden. Tests dienen dazu, den Entwicklungsvorgang zu überwachen und frühzeitig Mängel zu erkennen. Getestet wird permanent, nicht ausschließlich in einer Testphase des Projektes.
- Die Struktur des Systems wird in **Tests, Quellcode** und der **mündlichen Kommunikation** festgehalten. Im Gegensatz steht dies zur aufwendigen Dokumentation in spezifikationsorientierten Vorgehensweisen.
- Die **geringen Kosten**, die bei einer Änderung der Anforderungen oder bei Auftritt eines Fehlers anfallen. Auf die Kosten wird im fol-

genden Abschnitt detailliert eingegangen.[1]

Ziel agiler Methoden ist es, eine Flexibilität zu ermöglichen und auf Änderungen reagieren zu können. Es soll des Weiteren nahezu gleich teuer sein, ob ein Fehler früh oder spät im Projekt gefunden wird.

Erreicht wird dies durch Arbeiten in kurzen Zyklen. Ein weiterer Vorteil besteht darin, dass nicht der gesamte Prozess der spezifikationsorientierten Softwareentwicklung durchlaufen werden muss. Ändert sich eine Anforderung im spezifikationsorientierten Vorgehen, wird bei der Neudefinition der Anforderungen begonnen. Die Kostenkurve ist somit exponentiell. Bei XP entfällt eine aufwendige Planung. Ein Nachbearbeiten der Spezifikationen ist nicht notwendig. Nachfolgende Abbildung verdeutlicht diesen Ansatz durch eine flache Kostenkurve bei XP:

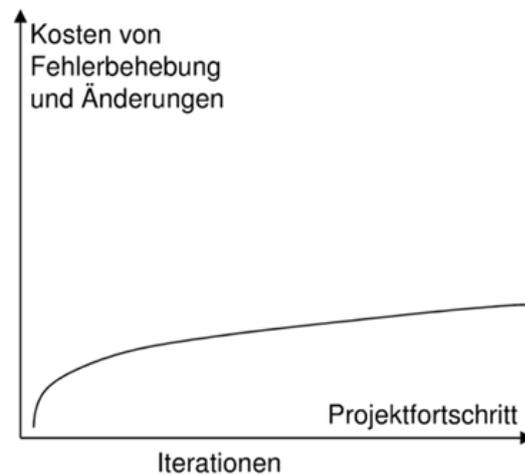


Abbildung 1: Fehlerkostenkurve Extreme Programming

XP legte die Grundsteine für die heute weit verbreitete Methode der agilen Softwareentwicklung. Große Teile der Ideen, welche Kent Beck in seinem Buch über Extreme Programming aufstellt, findet man im Agilen Manifest [3] aus dem Jahre 2001 wieder, welches Kent Beck mit begründete und mit unterzeichnete.

[1] Beck, Kent. 2000. Extreme Programming – Das Manifest. s.l. : Addison-Wesley, 2000. 3–8273–1709–6.

[2] Chrysler. 1998. [Online] Oktober 1998.

[Zitat vom: 24. März 2015.] <http://www.xprogramming.com/publications/dc9810cs.pdf>.

[3] Beck, Kent, et al. 2001. Agile Manifesto. [Online] 2001. [Zitat vom: 10. April 2015.] <http://agilemanifesto.org>

Bildquellen:

- Abbildung 1: Beck, Kent. 2000. Extreme Programming – Das Manifest

