



Informationstechnik

Hochschule Esslingen
University of Applied Sciences

IT-Innovationen

Band 17
Juni 2016

Grußwort des Dekans

Liebe Leserinnen und Leser,

mit der nach oben geöffneten Handfläche in die gewünschte Richtung wischen und das Schiebedach öffnet und schließt wieder, Sitze mit einer Geste komfortabler einstellen: Gestensteuerung sorgt für eine intuitivere und natürlichere Bedienung technischer Systeme und Anlagen. Versehentlich gegebene Befehle werden so vermieden.



Gestensteuerung ist nur eine von vielen Zukunftstechnologien, die wesentlich durch Informationstechnologie getrieben wird. Unsere Absolventen greifen zusammen mit unseren Industriepartnern solche Fragestellungen rund um Innovationen auf und versuchen Antworten auf viele noch offene, brennende technologische Fragestellungen zu finden. So auch in diesem Ihnen vorliegenden Band unserer IT-Innovationen. Neben Beiträgen zur Gestensteuerung finden Sie darin interessante Berichte zur Simulation von Verkehrsflusssteuerung, zur modellbasierten Entwicklung, zu automatisierten Testumgebungen für Software, zu Systemen für die Erfassung, Auswertung und Vorhersage von Emissionswerten, um nur einige zu nennen.

Viel Freude beim Lesen der Beiträge unserer Absolventen
wünscht Ihnen 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

Jordanis Andreanidis	Qualität von Software	1
Waldemar Belikow	Maschinenmanagement 4.0	3
Edis Camović	Analyse und Implementierung von Map-Matching-Algorithmen zur fehlerfreien Abbildung von Telematikdaten auf OpenStreetMap mittels neo4j und Python	5
Frederik Eschmann	Die Entstehung eines Computerspiels – Von der Idee bis zum ersten spielbaren Level	7
Clemens Fabig	Konzeption und prototypische Umsetzung eines Mikrocontroller-basierten Konverters zur Umsetzung der Protokolle verschiedener Feldbussysteme auf Ethernet für eine PC-gestützte Analyse	11
Stephan Fleischer	Aufbau einer Simulation einer Kärcher PKW-Portalwaschanlage für die Evaluation von konzeptionellen Designentwürfen in v-rep	13
Gina Frank	Integration von 3D HMI-Systemen in moderne Industrie 4.0 Automatisierungs-Architekturen	15
Philipp Hornuß	Konzeption und Implementierung einer Konsolenapplikation in C# zur Erfassung von Lasersimulationsdaten in einer SQL-Datenbank einschließlich der auf den Simulationsdaten basierenden Entwicklung von Analysemethoden und Visualisierungselementen	17
Elif Isildar	Erkennung von dynamischen Freiraumgesten mit Hilfe des Maschinellen Lernens	19
Maria Koblova	Die Clean Code Developer-Initiative	21
Manuel Köhler	Entropiequellen zur Erzeugung kryptografisch sicherer Zufallszahlen in eingebetteten Systemen	24
Ernst Lehmann	Analyse von Mechanismen zur Synchronisation von seriellen Kommunikationsdaten und analogen Signalen einschließlich prototypischer Integration in die standardisierte Hardware-/Softwarearchitektur für Prüfsysteme bei Festo	26
David Mändlen	Open Source Case Management	28
Oliver Melchior	Erstellung eines Closed-Loop Systems zur Durchführung automatisierter Tests von Software einer Getriebesteuerung	30
Denis Metzger	Feature Driven Development	32
Chris Manuel Mönch	Konzepterstellung und Realisierung einer konfigurierbaren SENT-Schnittstelle nach SAE J2716 mit optimalem Formfaktor unter Anwendung des Verfahrens Hardware-Software-Codesign	34
Dominik Müller	Konzeption und Implementierung einer App zur Interaktion mit der Produktlebenslaufakte in einer Shopfloor-Umgebung	37
Eugen Muschewski	Die Entstehung eines Computerspiels – Von der Idee bis zum ersten spielbaren Level	7
Matthias Noll	Funktionserweiterung eines Systems für das Condition-Monitoring unter Berücksichtigung von Aspekten der Network Security	39

Philipp Rauscher	Automatisierte Validierung kennfeldbasierter Applikationsdaten	41
Maximilian Röhrle	Konzipierung und Realisierung der Anbindung von Ladesäulen für Elektrofahrzeuge an ein SmartHome-System einschließlich Implementierung des Lastmanagements in C++ unter Berücksichtigung der verfügbaren Anschlussleistung	43
Fatih Emin Sahin	Erweiterung eines Expertensystems zur automatischen Auswertung von Messdaten in der Lenkungsentwicklung	45
Maximilian Schall	Abhängigkeiten reduzieren mit Flow Design	47
Markus Scheider	Was ist Testen?	49
Kerstin Schneider	Auslegung, Implementierung und Inbetriebnahme eines Zustandsreglers sowie eines Fuzzy-Reglers für einen nichtlinearen Prozess mit zwei Stellgrößen zur Positionierung eines Balls in einem Ball-Wippe-System	52
Robin Schotka	Konzipierung und prototypische Realisierung eines generischen Testsystems zum automatisierten Testen von Motorcontrollern für Servomotoren nach dem Prinzip Hardware-in-the-Loop unter Verwendung von dSpace-Komponenten sowie MATLAB Simulink und C# für die Steuerung des Systems	54
David Seddaoui	Entwicklung eines Feature-Trackings zur systematischen Bewertung der Usability eines Lifecycle-Management-Systems	56
Sammy Traxler	Konzeption zur Leistungssteigerung von hochauflösenden Pixel-Scheinwerfern für ein Erprobungsfahrzeug einschließlich prototypischer Realisierung notwendiger Software-Komponenten in MATLAB Simulink	58
Ricardo Zeich	Machine-Learning-Verfahren zur Detektion von Mobilitätsmodi	60

Qualität von Software

Jordanis Andreanidis*, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Softwareprodukte sind in der Regel während ihres kompletten Lebenszyklus Änderungen und Erweiterungen unterworfen. Ist eine Software in ihrer Struktur zu starr oder zu zerbrechlich, so verkürzen diese Änderungen und Erweiterungen die Lebensdauer einer Software erheblich. Im Umkehrschluss bedeutet dies: Nicht allein die äußeren Merkmale einer Software wie etwa die Erfüllung der Anforderungen oder ihre Gebrauchstauglichkeit sind maßgebend für die Qualität einer Software. Doch was bedeutet "Qualität" im Hinblick auf Softwareprodukte und wie misst man sie? Um den Begriff der Softwarequalität definieren zu können, bedarf es zunächst einmal der Betrachtung des allgemeinen Qualitätsbegriffs.

Definition von Softwarequalität

Das Wort Qualität stammt von dem lateinischen Wort "qualitas", welches mit "Beschaffenheit", "Merkmal", "Eigenschaft" oder auch "Zustand" übersetzt wird. Qualität ist zunächst ein abstrakter Begriff, der erst bewertbar gemacht werden muss, um Produkte, also Waren oder Projekte, bewerten zu können. Wenn man abstrakte Begriffe messbar macht, spricht man von einer Operationalisierung.

Für Softwarequalität gibt es verschiedene Definitionen. Eine davon lautet nach der DIN-Norm ISO/IEC 9126 [1]:

"Qualität ist die Gesamtheit der Merkmale und Merkmalswerte eines Software-Produkts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen."

Softwarequalität ist ein grundsätzlich komplexer und vielschichtiger Begriff. Er kann aus verschiedenen Perspektiven betrachtet werden und durch Qualitätsmanagement gesteigert werden. Während ein Benutzer eines Systems besonderen Wert auf die Benutzbarkeit legt, achtet ein Entwickler beispielsweise mehr auf Eigenschaften wie Erweiterbarkeit und Stabilität des Systems. Die verschiedenen Ansichten der verschiedenen Stakeholder können in einer Aufteilung der Qualität, in die sogenannte

innere und äußere Qualität, abgebildet werden.

Innere und äußere Qualität

Verschiedene Rollen von Stakeholdern richten ihre Aufmerksamkeit auf verschiedene Qualitätsziele. Die innere Qualität wird von den Entwicklern beziehungsweise dem Entwicklungsteam in Betracht gezogen. Die äußere Qualität wird aus der Perspektive des Kunden wie z. B. seiner Benutzer bewertet. Abbildung 1 zeigt, dass verschiedene Stakeholder verschiedene Sichten auf die Qualität haben.

Innere Qualität bedeutet:

- Korrektheit der Konstruktion,
- Eleganz des Designs,
- wenig Abhängigkeiten,
- Einfachheit,
- Änderbarkeit und Erweiterbarkeit sowie
- Übereinstimmung mit Standards, Patterns und Konventionen

Äußere Qualität bedeutet:

- Vorhandensein der geforderten Funktionalität und Korrektheit,
- Bedienbarkeit und Eleganz der Benutzeroberfläche,
- kurze Antwortzeiten,
- Verlässlichkeit,
- Stabilität gegen Hard- und Softwarefehler sowie
- Robustheit gegen Fehleingaben.

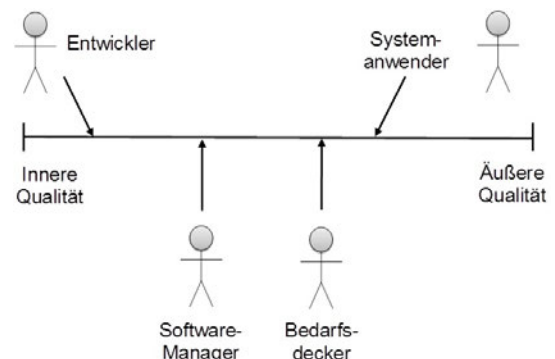


Abbildung 1: Spektrum der inneren und äußeren Qualität

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

Allgemeine Definitionen der Qualität helfen für sich alleine nicht viel weiter. Daher wird die Qualität durch Qualitätsmodelle bewertbar gemacht. Qualitätsmodelle sind strukturierte Zusammenstellungen von Qualitätsmerkmalen, um die Qualität einer Software zu definieren, zu messen und zu beurteilen.

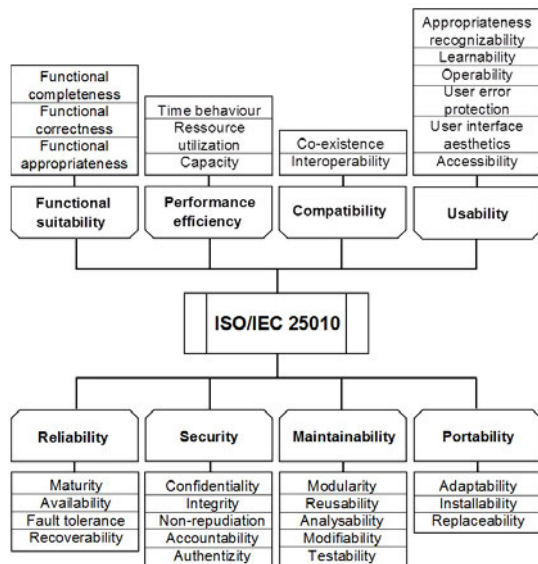


Abbildung 2: Das Qualitätsmodell der ISO/IEC 25010

Die Norm ISO/IEC 25010

Die bereits erwähnte Norm ISO/IEC 9126 ist im Jahr 2011 in der Norm ISO/IEC 25010 (SQuARE – Qualitätsmodell und Leitlinien) [3] aufgegangen und beschreibt den aktuellsten Standard für Produktqualität. In dieser Norm wird unter anderem ein Qualitätsmodell für die Produktqualität durch acht Qualitätsmerkmale beschrieben. Diese acht Merkmale sind jeweils in eine Vielzahl von Teilmerkmalen, welche sich auf die statischen Eigenschaften der Software und die dynamischen Eigenschaften des Computersystems beziehen, aufgeteilt (siehe Abbildung 2). Durch Betrachtung des Qualitätsmodells ISO/IEC 25010 kann Software anhand dieser Kriterien bewertet werden. Dabei muss im Kontext entschieden werden, welche Softwaremerkmale besonders relevant sind.

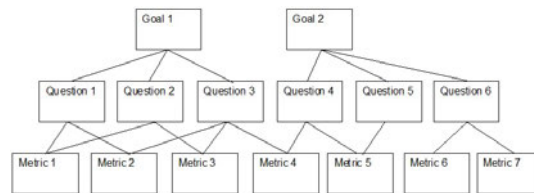


Abbildung 3: Goal-Question-Metric-Ansatz

Der Goal-Question-Metric-Ansatz für projektbezogene Qualitätsmodelle

Beim Goal-Question-Metric-Ansatz von Basili [2] werden Qualitätsziele, die aus den Unternehmenszielen abgeleitet sind, projektbezogen bestimmt und messbar gemacht. Zuerst werden die Qualitätsziele ermittelt. Anschließend ordnet man einem Qualitätsziel Fragen zu. Hierdurch werden die Qualitätsteilmerkmale ermittelt. Die Qualität wird also mit Hilfe der Fragen operationalisiert und durch Metriken (Qualitätsindikatoren) messbar gemacht. Dabei stellt man den Zielen (Goals), also den Qualitätsteilmerkmalen, Metriken gegenüber. Indirekt erreicht man dadurch die Zuordnung von Qualitätsziel zu Metrik. Und damit sind die Qualitätsziele durch Metriken bewertbar gemacht [2]. Abbildung 3 visualisiert den Goal-Question-Metric-Ansatz.

Fazit

Zusammenfassend sind Softwarequalitätsmodelle ein wichtiges Hilfsmittel, um Software zu operationalisieren und bewertbar zu machen. Qualitätsmodelle bleiben nicht konstant, sondern werden, wenn nötig, optimiert und weiterverändert. Ein Beispiel ist der Wandel der ISO Norm 9126 in die ISO-Norm 25010 im Jahr 2011. Generell gilt es zu sagen, dass ein Qualitätsmodell in jedem Unternehmen oder Entwicklungsprozess einer Software so zu interpretieren oder anzupassen ist, dass die Erschaffung und Erhaltung von Software dem maximalen Nutzen aller Beteiligten dient und dass dadurch vor allem die Kundenwünsche und Anforderungen mit höchster Qualität erfüllt werden. Andererseits ist aber auch zu erwähnen, dass sich die gängigen Standards zur Qualität zu sehr an den Kunden orientieren und die innere Qualität, die über die Lebensdauer einer Software entscheidet, wesentlich zu kurz kommt.

[1] DIN ISO/IEC 9126-1:2001 (15.06.2001). Software engineering – Product quality – Part 1: Quality model

[2] J. Goll (2011). Methoden und Architekturen der Softwaretechnik, 1. Auflage, Springer Vieweg

[3] DIN ISO/IEC 25010:2011 (01.03.2011). Systems and software Quality Requirements and Evaluation (SQuARE)

Bildquellen:

- Abbildung 1: Eigener Entwurf. Angelehnt an „Palmer, Felsing (2002): A Practical Guide to Feature-Driven Development. Prentice Hall. S. 14“
- Abbildung 2: Eigener Entwurf.
- Abbildung 3: [2], S. 141

Maschinenmanagement 4.0

Waldemar Belikow*, Jörg Friedrich

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Maschinenmanagement 4.0 ist ein Schlagwort, das hohe Erwartungen an Produktionsabläufe weckt. Dabei sollen die Abläufe möglichst effizient gestaltet werden. Insbesondere liegt der Fokus auf der Erhöhung der allgemeinen Produktivität sowie der Kapazität. Gerade im Bereich der Maschinenproduktion ist Druckluft nicht mehr wegzudenken. Jedoch gilt Druckluft mithin als wenig energieeffizient.

Zur Behebung des Problems bietet Festo das Energieeffizienzmodul (E²M) als Tool an. Dieses Tool dient der Maximierung der Effizienz bei gleichzeitiger Minimierung des Verbrauchs. Ziel ist es, die Verbraucher zu informieren und ihnen einen Überblick über den Energieverbrauch zu verschaffen. Die Verbraucher sollen dadurch präventiv Maßnahmen, wie z.B. die rechtzeitige Wartung von Maschinen ergreifen. Allerdings gibt es noch keine Software für die Zustandsüberwachung des Tools. Um dieses Versäumnis auszuräumen soll ein Lastenheft für die Condition Monitoring Software für das E²M erstellt werden.

Hier knüpft das Projekt der Bachelorarbeit „Maschinenmanagement 4.0: Requirements Engineering im Condition Monitoring und Visualisierung von Energieeffizienz Zeile zur Maschinensteuerung“ an, das bei der Firma Festo AG & Co.KG durchgeführt wird. Im Rahmen der Arbeit werden Kundenbefragungen bezüglich der Zustandsüberwachung durchgeführt, um die Anforderungen an die Software festzulegen. Diese Anforderungen werden anschließend in einem Lastenheft festgehalten.

Der Mehrwert ist dabei deutlich erkennbar, da der Kunde mit dem Kauf des Tools ein übersichtliches Instrument erhält, das keiner weiteren Programmierung bedarf. Die Visualisierung der Energieeffizienz soll dem Betreiber eine bessere Transparenz über den Zustand seiner Anlage verschaffen. Zusätzlich soll es ihm einen besseren Überblick über außergewöhnliche Vorfälle bieten, um ein schnelleres Eingreifen bei Fehlfunktionen von Maschinen zu ermöglichen.

Für die Umsetzung des Projekts sind die sorgfältige Vorbereitung der Kundenbefra-

gungen und anschließende Evaluation der Angaben notwendig.

Condition Monitoring

Unter Condition Monitoring versteht man Kolerus und Wassermann (2014) zufolge die Zustandsüberwachung von Maschinen. Es handelt sich um ein Werkzeug in Form einer Software, welches bei der Analyse der Zustände von Industrieanlagen helfen kann, indem es aufbereitete Daten auf einem Monitor grafisch darstellt.

Die wichtigsten Elemente der Zustandsüberwachung sind Sicherheit und Energieeffizienz. Die Sicherheit wird bei in Echtzeit gemessenen Daten und schnell reagierenden Systemen gewährleistet. Aufgrund der Maschinenüberwachung und aufbereiteter Daten ist eine zustandsorientierte, intelligente Instandhaltung möglich. Des Weiteren dient das Condition Monitoring der Kosteneinsparung [1].

Ziel des Projekts ist, dem Kunden eine intelligente Wartungseinheit mit der zugehörigen Software zu bieten, die Daten aufbereitet. Mit dem im Rahmen des Projekts herausgearbeiteten Anforderungen soll auch das Ziel der Industrie 4.0 erfüllt werden [2].

Das E²M bietet eine Profi-Bus oder eine Profi-Net Schnittstelle, über die die Daten übertragen werden. Es werden Werte wie Druck, Verbrauch oder Durchfluss gemessen. Zudem gibt es die Möglichkeit Parameter einzugeben, anhand derer das Tool reagieren soll (Luftzufuhr Ein/Aus). Bei Fehlfunktion des Moduls wird die Luftzufuhr automatisch aufgedreht, sodass der Produktionsfluss nie durch einen Fehler des Moduls unterbrochen werden kann.

In Abbildung 1 wird der Durchfluss der Druckluft bei einem Produktionslauf dargestellt (die Werte dienen nur zum besseren Verständnis und sind nicht realistisch). Durch die rote Linie wird der Schwellwert festgelegt, bei dem die Maschine nicht aktiv ist. Der Wert wird deshalb nicht auf 0 gesetzt, weil von Leckage

*Diese Arbeit wurde durchgeführt bei der Firma Festo AG & Co.KG, Esslingen/Berkheim

auszugehen ist. Befindet sich der Durchfluss länger unter dem Schwellwert als Δt , wird die Druckluftzufuhr von E²M zugedreht.

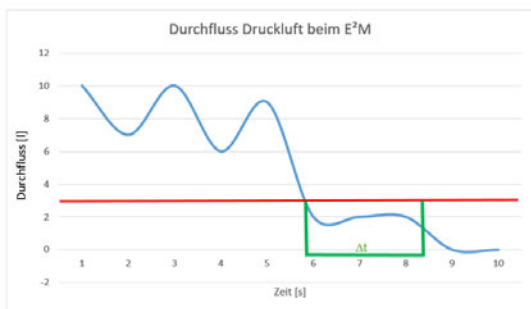


Abbildung 1: Durchfluss-Erfassung der Druckluft durch das E²M

Einige Energieeffizienzmodule sind bereits erfolgreich im Einsatz. Die erste Kundenanforderung ist, die Erstellung einer CSV Datei, in die die gesammelten Daten exportiert werden können, um sie dann in Excel auszuwerten. Jedoch handelt es sich hierbei um keine langfristige Lösung, da eine große Menge an Daten erfasst wird und somit Excel an seine Grenzen stößt.

In Zukunft soll dem Kunden eine Software geboten werden, bei der alle gewünschten Daten aufbereitet in einer grafischen Oberfläche (z.B. Diagrammen) angezeigt werden. Die

Software soll verschiedene Produktionsabläufe und deren Taktungen automatisch erkennen. Außerdem soll der Betreiber Referenzwerte schaffen können und bei Abweichungen benachrichtigt werden.

Eine weitere Möglichkeit ist die Leckage Messung im Stand-By-Modus. Dieses Problem soll in Abbildung 2 verdeutlicht werden. Dabei wird der Luftdruckabfall an der Anlage gemessen. Nichtsdestotrotz wird die Möglichkeit bestehen bleiben die Daten per CSV Datei exportieren zu können.

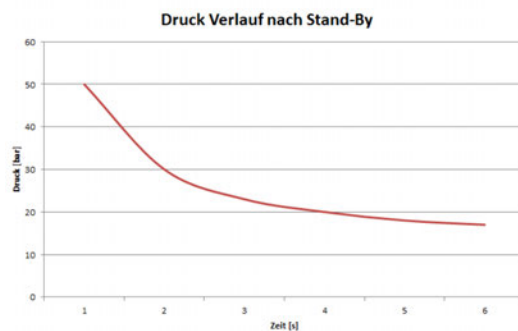


Abbildung 2: Druckverlauf einer Anlage im Stand-By Modus bei Leckage

Die Architektur (Interner Server, Cloud, ...) wird Anlass für weitere Überlegungen sein.

[1] Zustandsüberwachung von Maschinen, Kolerus / Wassermann

[2] An dieser Stelle sei auf die Auflistung der Anforderungen durch das Fraunhofer Institut verwiesen unter: <http://www.iao.fraunhofer.de/langde/images/iao-news/produktionsarbeitderzukunft.pdf> (Zuletzt gesehen am 12.05.2016)

Bildquellen:

- Abbildung 1,2: Selbsterstelltes Diagramm

Analyse und Implementierung von Map-Matching-Algorithmen zur fehlerfreien Abbildung von Telematikdaten auf OpenStreetMap mittels neo4j und Python

Edis Camović*, Reinhard Keller, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Ob es bei der herkömmlichen Navigation im Auto ist oder bei einer Auswertung der Trainingseinheit, digitale Landkarten sind in unserem Alltag kaum mehr wegzudenken. Hilfreich sind diese Karten im Kontext der Fahrtenauswertung, bei denen GPS-Koordinaten aufgezeichnet wurden. Die Genauigkeit dieser Koordinaten hängt aber von vielen Faktoren ab, die nicht immer beeinflussbar sind, wie zum Beispiel der Genauigkeit des Empfängers oder auch das Wetter und der Ort, an dem man sich befindet. Umso wichtiger ist es in diesen Fällen, dass man eine Lösung findet, wie man ungenaue oder fehlende Signale in Fahrten verbessern kann.

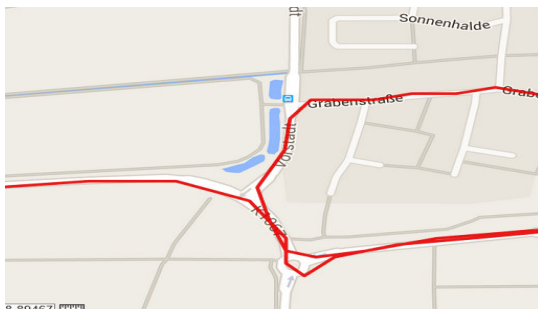


Abbildung 1: Ausschnitt aus einer Fahrt

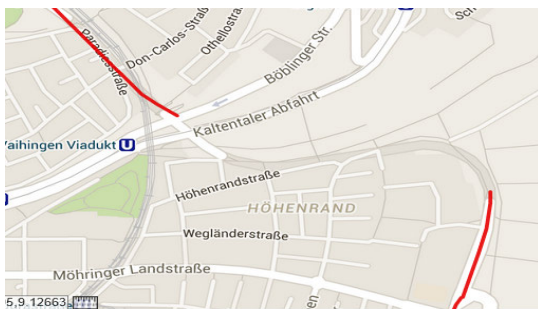


Abbildung 2: Ausschnitt aus einer Fahrt

In Abbildung 1 ist ein Ausschnitt einer Fahrt dargestellt, während der die GPS-Koordinaten alle 5 Sekunden abgespeichert wurden und dann mit Hilfe von Verbindungslinien versucht

wurde, die Fahrt zu rekonstruieren. Schön zu sehen ist, dass es GPS-Punkte gibt, die neben der Straße liegen, auf der falschen Fahrbahnseite verlaufen oder die Linien quer durch den Kreisverkehr gezogen werden, was nicht der originalen Fahrt entspricht. In Abbildung 2 fehlt ein Teil der Strecke, weil das Auto durch einen Tunnel gefahren ist. Mit solchen Problemen haben die Hersteller von Navigationsgeräten täglich zu kämpfen. Im Rahmen dieser Abschlussarbeit mussten Algorithmen gefunden werden, die solche Lücken und Fehler ausbessern.

Map-Matching ist hierfür eine Lösung. Das bedeutet Algorithmen zu finden, um die GPS-Koordinaten auf Straßenverläufe zu projizieren, damit verschiedene Anwendungen im Bereich der Navigation oder Fahrtenauswertung umgesetzt werden können. Das Fraunhofer Institut für Arbeitswirtschaft und Organisation mit dem Anwendungszentrum KEIM, welches September 2012 ins Leben gerufen wurde, hat eine Vielzahl an Elektroautos und sammelt seit 2013 Daten einer Flotte aus Elektroautos, um besseren Service für die Nutzer zu schaffen und die Akzeptanz der Elektromobilität zu verbessern. In der Lösung „EcoGuru“, einem System zur Buchung und Nutzung von E-Fahrzeugen, werden die gesammelten Telematikdaten verarbeitet und für zukünftige Fahrplanungen verwendet. Inhalt dieses Konzepts ist zum einen die Motivation zu ökonomischeren Fahrten zum anderen dem Nutzer unterstützend zur Seite zu stehen.[1]

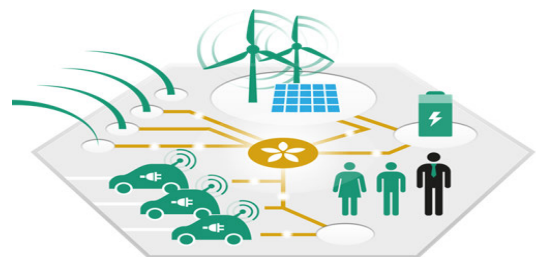


Abbildung 3: Vernetzung Informationen

*Diese Arbeit wurde durchgeführt bei der Firma Fraunhofer IAO | Anwendungszentrum KEIM, Esslingen a.N.

Damit eine optimale Funktionsweise geboten werden kann, wird ein Datenaustausch zwischen den einzelnen Komponenten wie in Abbildung 3 durchgeführt.

Ein Teil davon sind die aufgezeichneten Telematikdaten, die die GPS-Koordinaten der Fahrzeuge beinhalten. Um aus diesen Datensätzen verschiedene Informationen herauszufiltern, zum Beispiel den Einsatzbereich der Fahrzeuge der Flotte und das Verhalten des Fahrer im Bezug auf Akkuverbrauch usw., müssen diese Daten mit Zusatzinformationen zu einem reinen Kartenmaterial angereichert werden.

Im Rahmen dieser Abschlussarbeit wurde ein Konzept entwickelt, welches fehlerhafte oder fehlende Daten mittels eines Algorithmus korrigiert und auf eine digitale Karte abbilden soll. Die Zuordnung der GPS-Punkte zu vorhandenen Straßenpunkten der digitalen Karte ist ein komplexes Thema, da es manchmal mehrere Zuordnungspunkte geben kann, oder auch überhaupt keine, falls ein GPS-Punkt fehlerhaft ist. Problematisch sind auf Landkarten auch Straßen, die direkt nebeneinander oder übereinander verlaufen, da hier eine direkte Zuordnung nicht stattfinden kann.

Bei fehlenden Signalen, wie zum Beispiel in

Tunnelabschnitten oder auf Landstraßen zwischen Wäldern, müssen die Koordinatenpunkte ersetzt und eingefügt werden, damit keine Lücken auf der Karte entstehen.

Das gesamte Straßennetz ist als Graphen-Datenbank gespeichert (Abbildung 4). Die Graphen beinhalten Informationen zu:

- Längen- und Breitengrad des Straßenpunktes
- Verbindungen zu anderen Knotenpunkten
- Kosten zu anderen Knotenpunkten
- Informationen zur Straße (30 Zone, Ampel)

Die aufgezeichneten Fahrten mit GPS-Punkten besitzen auch Informationen aus Längen- und Breitengrad, aus denen man die Position und auch die Geschwindigkeit errechnen kann.

Der Umfang dieser Bachelorarbeit beinhaltet eine Recherche über vorhandene Algorithmen und die Entwicklung von drei geeigneten Möglichkeiten, die Map-Matching-Problematik zu lösen und den aufgezeichneten Fahrten eindeutige Routen auf einer Karte zuzuweisen.



Abbildung 4: Straßennetz als Graphendatenbank und auf einer Map

[1] Fraunhofer IAO | Anwendungszentrum KEIM

Bildquellen:

- Abbildung 1,2,4: Eigene Grafik
- Abbildung 3: Fraunhofer IAO | KEIM

Die Entstehung eines Computerspiels – Von der Idee bis zum ersten spielbaren Level

Frederik Eschmann*, Eugen Muschewski*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Der Inhalt der Bachelorarbeit behandelt die Schritte eines Computerspiels, die von der Ideenfindung bis hin zum spielbaren Level durchlebt werden. Die Arbeit basiert vor allem auf den Erfahrungen, welche die Autoren bei der Entwicklung eines eigenen Spiels während des Abschlusssemesters gesammelt haben. Bei dem Spiel handelt es sich um ein 2.5D-Side-Scroller Rollenspiel, welches mit der Engine Unity3D entwickelt wurde.

Doch was für ein Spiel ist das? Hierzu gibt es zwei Begriffe zu klären: Side-Scroller und Rollenspiel. Als Side-Scroller werden allgemein Spiele bezeichnet, bei denen der Spieler nur von der Seite auf das Spielgeschehen schaut. Klassisch gibt es vor allem "Jump 'n' Runs", "Shoot 'em ups" oder "Beat 'em ups" mit dieser Ansicht. Ein Rollenspiel ist ein Spiel, in dem Spieler die Rollen fiktiver Charaktere oder Figuren übernehmen und selbst handelnd soziale Situationen beziehungsweise Abenteuer in einer erdachten Welt erleben.

Von der Idee zum Konzept – Computerspiele in der Produktionspipeline
Eine Idee findet man überall, egal ob der Ideenfunder dabei schläft, wach ist oder gerade seine Wohnung staubsaugt. Wichtig dabei ist nur, Augen und Ohren immer und überall offen zu halten. Sobald eine Idee gefunden ist, folgt anschließend darauf der kreativ-künstlerische Prozess, diese Idee auszubauen. Durch das intensive Eintauchen in die Idee entstehen mentale Prototypen, Szenarien, Charaktere und Features, die in der Fantasie ausprobiert werden. Marktanalyse und Recherche gehören bei der Ideenfindung auch dazu. Hat man eine Idee, kann im Vorfeld geprüft werden, ob diese schon in ähnlicher Form existiert oder innovativ ist, welche Spiele dies ähnlich machen, warum und wem diese Spiele Spaß machen.

Schon zu Beginn wird also die Zielgruppe festgemacht. Aus der Sicht nach Bartle und Caillois [1, S. 15f] stellen sich 7 charakteristische Spielertypen auf:

- Erkunder – Hauptmotivation im Entde-

cken und Erkunden der Spielwelt oder auch Lösen von Rätseln.

- Wettkämpfer – Lieben es, sich mit anderen Spielern zu messen, besonders durch Kämpfen, Turnieren und Ranglisten.
- Socialicer – Kommunikation steht im Vordergrund. Kennenlernen von anderen Spielern, aber ebenso Augenmerk auf die Interaktion mit der Spielwelt.
- Erfolgsspieler – Streben nach hohen und vielen Erfolgen im Spiel. Beispiele dafür wären, schnell das höchste Level zu erreichen oder bestimmte Missionen in kürzerer Zeit als andere Spieler abzuschließen.
- Glücksspieler – Der Glücksspieler liebt die Wahrscheinlichkeit und den Kick, wenn er nach Abschluss einer Mission mit seltenen Gegenständen belohnt wird.
- Rollenspieler – Die Faszination fängt bei ihm schon bei der Erstellung des Charakters an, ebenso aber auch, wenn er in die Fantasiewelt des Spieles komplett eintauchen kann und die spannenden Abenteuer der Heldenfiguren nachspielen kann.
- Rauschspieler – Typische Genre sind hier schnelle Spiele, darunter Rennspiele oder auch Tanzspiele – Hauptsache rasant. Dieser liebt leichte Schwindelgefühle oder auch das Adrenalin, das beim Spielen entsteht.

Es können in einem Spiel oft mehrere dieser Typen abgedeckt werden, jedoch besteht hier die Gefahr, dann nur eine Schnittmenge der Spielerschaft dieser Spiele zu erreichen. Wird zum Beispiel ein Rollenspiel mit Rennspielementen entwickelt, spricht das weder alle Rollen- noch alle Rennspielespieler an, sondern nur die Rollenspieler, die auch gerne Rennspiele spielen. Dies muss schon bei der Ideenfindung mit berücksichtigt werden. In diesen Fall muss eine sorgfältige Marktanalyse erfolgen. Spiele verbinden diese Typen oft sehr gekonnt, indem sie dem Spieler auch die Möglichkeit geben, bestimmte Inhalte im Spiel

*Diese Arbeit wurde durchgeführt an der Fakultät Informationstechnik

zu überspringen.

Ein Spiel wie zum Beispiel Skyrim bietet eine Variation verschiedener Funktionen an, die der Spieler je nach Bedürfnis nutzen kann. Spieler, die vor allem Action und schnelle Kämpfe erleben wollen, können durch eine Schnellreisefunktion die Wege zwischen den Missionen abkürzen. Entdecker wiederum verzichten meistens auf diese Funktion, da für sie der Weg auf dem Pferd oder zu Fuß selber auch ein Erlebnis ist.



Abbildung 1: Die Produktionspipeline eines Computerspiels

Es folgt daraus das "Mission Statement", das die grundlegende Idee in einem Satz mit den Kernelementen des Spieles beschreibt. Dazu kommt die Erfassung eines Kurzkonzpts, um allen Beteiligten vermitteln zu kön-

nen, um welche Art von Spiel es sich handelt, warum es sich lohnt das Spiel zu spielen, wie lange es ist und wie es sich anfühlt [1, S. 41]. Das Kurzkonzpt stellt im Allgemeinen die Vision des Projektes dar. In der Produktionspipeline wird dieses Konzpt auch als roter Faden verwendet, denn diese darf im weiteren Designprozess an die Anforderungen angepasst, jedoch nicht mehr grundsätzlich verändert werden.

Anhand einer Vision können weitere Details geklärt werden, diese werden dann in einem Game-Design-Dokument festgehalten. Fragen wie: "Welche Level gibt es?", "Welche Charaktere und wie sind sie untereinander verknüpft?", "Story und das Look-and-Feel der Spielwelt", "2 oder 3D?", "Welche Technik kommt zu Einsatz?"- werden hier geklärt und ausführlich dokumentiert.

Parallel zum Game-Design-Dokument entsteht der Projektplan. Dieser definiert, was, wann und wer zu erledigen hat, um bestimmte Ziele und Zwischenziele für erste Demonstrationen zu erreichen. Zeitgleich entsteht der Businessplan, der alle anfallenden Kosten über die Entwicklungszeit und die Zeit nach Veröffentlichung des Spieles beinhaltet.

Zwischen den Dimensionen – Der Einsatz von 2.5D in einem Side-Scroller-Computerspiel

Zu Anfang der Computerspielgeschichte bestand nur die Möglichkeit, Spiele im zweidimensionalen Raum darzustellen. Bekannt aus Klassikern wie dem Side-Scroller "Super Mario" und "Sonic the Hedgehog". Um einen Tiefeneindruck in einem 2D-Spiel zu vermitteln, wird im 2.5D, auch pseudo-3D genannt, gearbeitet. Um dies zu erreichen, wird zum Beispiel das Parallax Scrolling verwendet. Hierzu wird die Spielwelt in mehreren Schichten dargestellt. Somit werden im einfachsten Fall Vorder- und Hintergrund in einem festen Verhältnis zueinander verschoben, sobald sich die Spielfigur bewegt. Da sich der Hintergrund deutlich langsamer bewegt, entsteht ein Eindruck von Tiefe.

Eine weitere Methode, die auch in der Abschlussarbeit zum Einsatz kam, ist die Nutzung einer perspektivischen Kamera. Hierzu wird in der 3D-Welt gearbeitet, wobei die Kamera sich nur in zwei Dimensionen bewegen kann. Alle Elemente, wie Figuren und Objekte, sind in 2D, diese sind aus einer leicht schräg von oben blickenden Vogelperspektive gezeichnet. Durch die richtige Platzierung von 2D-Elementen in der Tiefe des 3D-Raumes entsteht der hybride 2.5D Effekt.

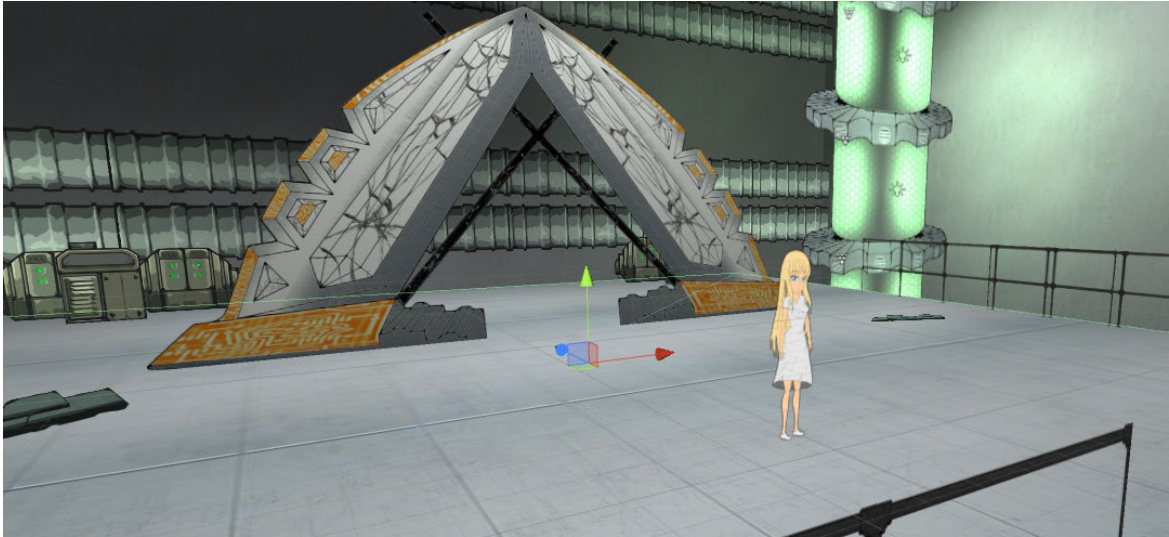


Abbildung 2: 2.5D im Einsatz – Einblick ins Spiel

Die ersten Schritte – Steuerung und Bewegungszustände

Um diese 2.5-dimensionale Welt erleben zu können, steuert der Spieler eine der Hauptfiguren des Spiels. Diese kann durch die verschiedenen Level gesteuert werden und mit der Umwelt, in Form von Objekten und anderen Charakteren, interagieren.

Doch wie bewegt man eine Figur durch einen 2.5-dimensionalen Raum? Dem Spieler wird zur Steuerung die Möglichkeit gegeben, den Charakter in zwei Dimensionen möglichst frei bewegen zu können. Wie schon in klassischen 2D Side-Scrollern, wie zum Beispiel "Super Mario Bros.", kann der Spieler den Charakter nach rechts und links steuern sowie springen oder an machen Stellen auch fallen. Das Bewegen in die Tiefe des Raumes wird nur an wenigen Stellen, bei der Interaktion mit Objekten oder in Sequenzen, freigegeben.

Damit sich der Charakter auch wirklich bewegt, werden die Eingaben des Spielers über die Tastatur oder einen Controller ausgewertet und dem Charakter mittels eines Sets an Bewegungen in Form eines Bewegungs-Zustands-Automaten übergeben. Die horizontale Bewegung wird zum Beispiel mit einem Wert von -1 bis 1 dargestellt, wobei 0 den Stillstand bedeutet. Die meisten Spiele kennen im Regelfall für die Bewegung vier verschiedene Geschwindigkeiten, also Bewegungszustände, für einen Charakter: Stehen, Gehen, Laufen, Rennen.

Im dargestellten Automaten für die Charakterbewegung, des für die Bachelorarbeit entwickelten Spiels, wurden nur die Bewegungszustände Stehen, Laufen und Rennen implementiert. Auch zu sehen ist, dass neben der horizontalen Bewegung auch der Sprung des Charakters im Automaten dargestellt ist. Was bei der horizontalen Bewegung noch nicht direkt ersichtlich ist, wird spätestens bei der Implementierung einer Sprungsequenz wich-

tig: Die Bodenerkennung. Um die Bewegung in manche Richtungen eingrenzen zu können, müssen für manche Objekte, wie für den Boden oder die Wände, Collider definiert werden.

Dem Charakter müssen dann ebenfalls Collider zugewiesen werden, mit denen eine Kollisionabfrage mit der Umgebung möglich ist. In Unity3D kann der Sprung nun über einen hinzugefügten RigidBody ausgeführt werden. Der RigidBody sorgt dafür, dass die Bewegungen einer Komponente, die mit diesem ausgestattet wurde, mit der Physics-Engine von Unity3D verwaltet wird. Für den Sprung muss dem Objekt, in diesem Fall dem Charakter, ein vertikaler Impuls gegeben werden, mit dem dieser ausgelöst wird. Durch definierte Gravitation wird die vertikale Bewegung nun gebremst und schließlich umgekehrt, bis der Objekt-Collider mit dem Boden-Collider zusammentrifft und das Objekt somit landet. In diesem Projekt wurde jedoch der Sprung nicht mit einer einzigen Animation dargestellt, sondern wurde in drei Zustände unterteilt: Absprung, Flugphase und Landung.

Bei den Animationen für den Sprung ist aber auch das Timing ein wichtiger Faktor. So sollte ein Charakter auch erst einen Sprungimpuls auslösen, sobald die Animation "Absprung" beendet ist. Auch bei der Landung kann ein schlechtes Timing für den ein oder anderen komischen Moment sorgen. Hier kommt die Kollisionserkennung zum tragen: Der Charakter muss erkennen, wann der Boden nahe genug ist um die Lande-Animation auslösen zu können. Hierfür werden die Collider des Bodens und die des Charakters auf Überschneidungen geprüft, vorausgesetzt die vertikale Geschwindigkeit des Charakters ist negativ – also, dass der Charakter auch fällt. Mit viel Feinarbeit entsteht so eine flüssige Animationskette.

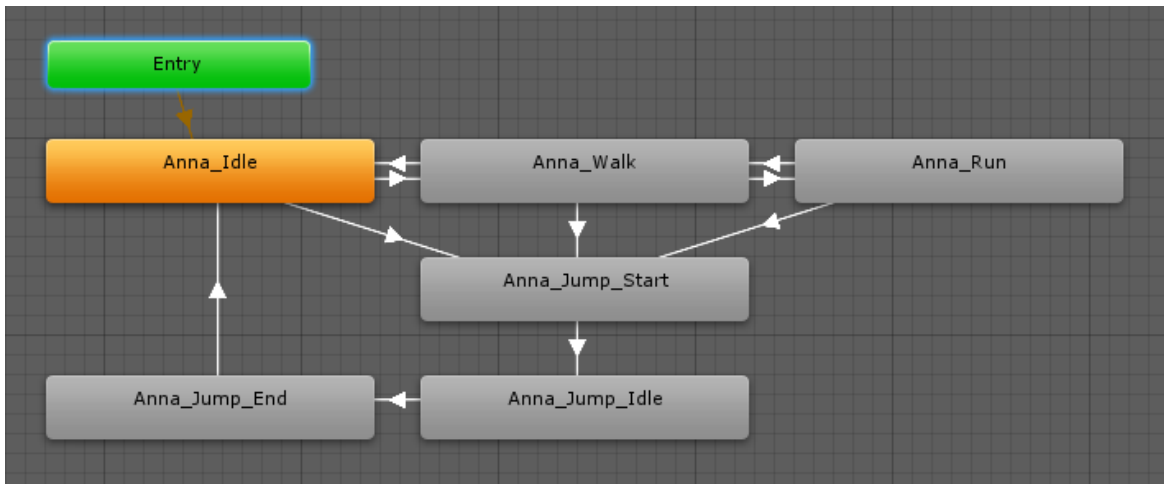


Abbildung 3: Die Bewegungszustände eines Hauptcharakters

Wenn das Spiel zu Denken lernt – Künstliche Intelligenz

In vielen Spielegenres soll eine künstliche Intelligenz das Spiel für den Spieler spannender machen und in manchen Fällen sogar erst ermöglichen. Dabei kann eine künstliche Intelligenz das Spielverhalten eines Gegenspielers simulieren und somit die Gegner-Rolle in beispielsweise einem Strategie- oder Shooter-Spiel übernehmen. Die Rolle der künstlichen Intelligenz in einem Adventure- beziehungsweise Rollenspiel ist meist jedoch die Steuerung der Umgebung. Sie übernimmt somit die Bewegungen und das Denken für alle Nicht-Spieler-Charaktere. Damit der Einsatz von künstlicher Intelligenz jedoch auch einen Mehrgewinn an Spaß in das Spiel bringt, ist es sinnvoll, dass der Computer zum Beispiel nicht perfekt spielt, da dies sonst sehr oft den kompletten Spielspaß zerstört. Doch kann es auch für den Spieler ein ebenso schlechtes Gefühl erzeugen, wenn ein computergesteuerter Gegner etwas extrem unlogisches tut, also beispielsweise endlos gegen dieselbe Wand läuft.

Brian Reynolds sagte zu diesem Thema:

„Beginnen Sie mit einer simplen Routine, die [...] Einheiten über die Karte wandern lässt [...]. Dann spielen Sie das Spiel [...] und beobachten Sie die Spielzüge des Computers. Verfolgen Sie das Treiben auf dem Monitor so lange, bis der Computer etwas wirklich Dummes macht [...]. Nun denken Sie darüber nach, was a) den Computer dazu veranlasst haben könnte, so etwas Dummes zu tun, b) was Sie anstelle des Computers getan hätten und c) welche Informationen ihrer Entscheidung zugrunde lagen. Anschließend arbeiten Sie die gewonnenen Informationen in ihren Algorithmus ein.“ [2, S. 219f]

Brian Reynolds beschreibt damit vor allem die Anfänge der künstlichen Intelligenz in Computerspielen. Heutzutage werden für die Entwicklung von künstlicher Intelligenz erfahrene Entwickler gesucht, mit denen im Voraus Konzepte entwickelt werden.

[1] Rehfeld, Gunther (2014): Game Design und Produktion. Carl Hanser Verlag München.

[2] Saltzman, Marc (1999): Game Design Die Geheimnisse der Profis. München: X-Games, Markt & Technik Verlag.

Bildquellen

- Abbildung 1–3: Eigene Darstellung

Konzeption und prototypische Umsetzung eines Mikrocontroller-basierten Konverters zur Umsetzung der Protokolle verschiedener Feldbussysteme auf Ethernet für eine PC-gestützte Analyse

Clemens Fabig*, Karlheinz Höfer, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Im Rahmen der vorliegenden Bachelorarbeit wurde ein Konzept für eine Embedded Plattform zur Protokollumsetzung von verschiedenen Feldbussystemen auf Ethernet entworfen. Dabei wird der Datenverkehr eines Feldbusses aufgezeichnet und in einem neu konzipierten Protokoll auf Ethernet abgebildet. Ziel ist es, den Datenverkehr auf dem Feldbus ohne Beeinflussung zu analysieren. Die zur Analyse bereitgestellten Daten sollen dann mittels einer PC-basierten Software ausgewertet werden können.

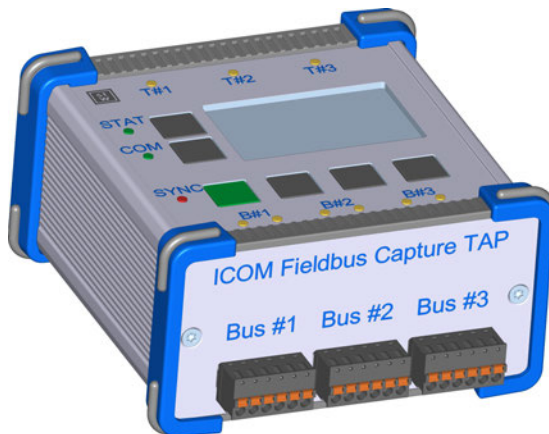


Abbildung 1: CAD-Zeichnung des ICOM Fieldbus Capture TAP

Feldbussysteme

Ein Feldbussystem ermöglicht eine kostengünstige Vernetzung von Sensoren und Aktoren mit einer zentralen Steuerung. Diese werden in der Automatisierungs- und Fertigungstechnik, der Gebäudeautomation und der Fahrzeugindustrie verwendet. Je nach Anwendungsbereich sind die Anforderungen an das Bussystem unterschiedlich. Daher hat sich seit den 80er Jahren eine große Vielfalt an ver-

schiedenen Systemen entwickelt [1].

In den letzten Jahren findet im Zuge der Entwicklung hin zur Industrie 4.0 ein Wechsel von den vorherrschenden Feldbussystemen zu den sogenannten Industrial-Ethernet Systemen statt. Im Gegensatz zu den proprietären Feldbussen basieren diese auf der klassischen Ethernet-Technik. Als Beispiel kann man hier Sercos nennen. Sercos vereint die Vorteile der Ethernet-basierten Kommunikation mit der Echtzeitfähigkeit typischer Feldbusse. Mit der PC-gestützten Software „Sercos Monitor“ existiert bereits ein leistungsfähiges Tool zur Analyse der Sercos-Kommunikation.

Ziel der Arbeit war es, eine Embedded Plattform zu konzipieren und prototypisch umzusetzen, die die Datenkommunikation der etablierten Feldbusse auf Ethernet abbilden kann. Unter Verwendung spezifischer Erweiterungen des Sercos Monitor soll so eine PC-gestützte Analyse der Feldbuskommunikation ermöglicht werden.

Marktanalyse

Die Vielfältigkeit der verwendeten Feldbussysteme machte zu Beginn der Arbeit eine Marktanalyse nötig. Diese sollte klären, anhand welcher Merkmale sich die einzelnen Bussysteme unterscheiden lassen und wo deren Gemeinsamkeiten liegen. Außerdem sollte ein Überblick gegeben werden, welche Feldbussysteme wie stark in welcher Branche verbreitet sind. Als Ergebnis lässt sich festhalten, dass die verwendete Übertragungsart und die Buszugriffsverfahren als klares Identifikationsmerkmal zur Klassifizierung herangezogen werden können. So finden sich Feldbussysteme mit deterministischem Zugriff und solche mit nichtdeterministischem Zugriff. Daneben existieren noch Systeme mit einer rein proprietären Signalübertragung, die sich keinem Standard zuordnen lassen [1], [2].

*Diese Arbeit wurde durchgeführt bei der Firma Steinbeis-Transferzentrum Systemtechnik, Esslingen

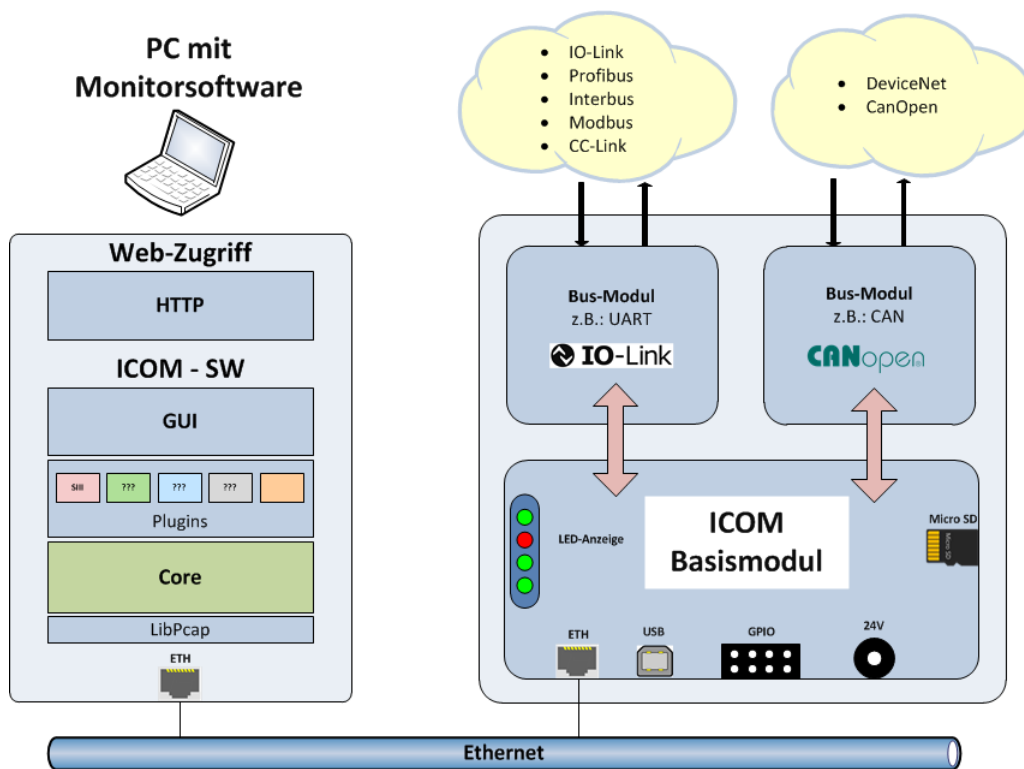


Abbildung 2: Systemübersicht des ICOM Fieldbus Capture TAP

Für die PC-gestützte Analyse existieren auf dem Markt bereits eine Vielzahl von Tools. Diese ermöglichen aber meist nur die Überwachung eines einzelnen Feldbusses. Eine echte Multiprotokolllösung, mit der eine Protokollkonvertierung auf Ethernet möglich ist, konnte bei der Marktanalyse nicht ermittelt werden.

Umsetzung

Unter Berücksichtigung der Fähigkeiten des Sercos Monitors und einer Betrachtung der auf dem Markt erhältlichen Geräte wurden Anforderungen an das Design der Embedded Plattform definiert. In Rücksprache mit den Entwicklern des Sercos Monitors und Partnern aus der Industrie wurden die funktionalen Anforderungen festgelegt. In Abbildung 2 ist ein konzeptioneller Aufbau des *Test Access Points* (TAP) mit seinen Anschlüssen abgebildet. Für die Realisierung der Embedded Plattform wurde durch das Steinbeis-Transferzentrum Systemtechnik eine prototypische Hardware entwickelt, welche in Abbildung 1 dargestellt ist.

Im Rahmen der vorliegenden Arbeit entstand darauf aufbauend eine erste Version der Firmware für das Basismodul. Es sollte untersucht werden, inwieweit die Umsetzung der funktionalen Anforderungen möglich ist. Bei dem zu untersuchenden System fiel die Entscheidung für das IO-Link System. IO-Link ist ein relativ neuer Kommunikationsstandard, über den intelligente Sensoren und Aktoren an die Anlagensteuerung angebunden werden können. Er nutzt als Basis die weit verbreitete UART-Kodierung zur Datenübertragung. Das macht die Ergebnisse dieser Arbeit übertragbar auf weitere Feldbussysteme, die ebenfalls UART-basiert sind.

Bei diesem ersten Entwurf der Firmware sollte getestet werden, wie genau der Zeitstempel für den Empfang einzelner Bytes gesetzt werden kann. Außerdem ist eine Betrachtung der erreichbaren maximalen Übertragungsraten, der Latenz und des Jitters während der Protokollumsetzung von Bedeutung.

-
- [1] Schnell, Gerhard (Hrsg.): Bussysteme in der Automatisierungs- und Prozesstechnik, Vieweg Verlag, Braunschweig, Wiesbaden, 2003
 [2] Klasen, F.; Oestreich, V.; Volz, M.: Industrielle Kommunikation mit Feldbus und Ethernet, VDE Verlag, Berlin, 2010

Bildquellen:

- Abbildung 1: Steinbeis-Transferzentrum Systemtechnik
- Abbildung 2: Eigene Grafik

Aufbau einer Simulation einer Kärcher PKW-Portalwaschanlage für die Evaluation von konzeptionellen Designentwürfen in v-rep

Stephan Fleischer*, Walter Lindermeir, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Reinigung und Sauberkeit nehmen heutzutage einen großen Stellenwert in Industrie, kommunalem Umfeld, Gewerbe und Haushalt ein. Nicht nur eine schnelle und gründliche Reinigung wird von Reinigungsgeräten erwartet, ebenfalls sind optimierte und sparsame Einsätze von Chemikalien und Frischwasser ausschlaggebende Kriterien, um im Markt der Reinigungsgeräte konkurrenzfähig bleiben zu können [1][2]. Die Zukunft der Reinigungsgeräte liegt in der präzisen Erkennung und Analyse von zu reinigenden Objekten. Um dies zu gewährleisten müssen die Aktoren Hand in Hand mit einer optimierten und präzisen Sensorik arbeiten und abgestimmt sein [2]. Abläufe werden prozesssicher gestaltet und sollten sich idealerweise unauffällig in den Alltag von Privatkunden oder gewerblichen Kunden einfinden.



Abbildung 1: Portalwaschanlage Typ CB28

Eine gründliche Fahrzeugpflege und Instandhaltung ist kein Alleinstellungsmerkmal großen Autohäuser und Fleet Managements. Privatkunden greifen schon längere Zeit auf professionelle Reinigung des eigenen Kraftfahrzeuges zurück. Eine PKW-Portalwaschanlage (siehe Abbildung 1) stellt

hierbei eine der häufig genutzten Optionen der Fahrzeugreinigung dar. Durch die technisch bedingte Form gelangte das Waschportal zu Ihrem Namen. Das Waschportal bewegt sich auf am Boden verankerten Schienen und wird über das Fahrzeug bewegt. Dadurch lassen sich die einzelnen Reinigungsmodule wie z.B. Hochdruckreinigung, Einsatz der Waschbürsten und Trocknung mit ein und demselben Portal durchführen.

Die Simulation einer solchen Anlage dient hierbei als gängiges Werkzeug und ist bei der Entwicklung eines Modelles unverzichtbar. Simulationen dieser Art werden durch Simulations-Frameworks ermöglicht. Das Programm virtual robot experimentation plattform, kurz v-rep, wurde speziell für die Simulation im Robotik-Bereich entwickelt [3]. In diesem Framework sind bereits gängige Sensoren implementiert und lassen sich ohne große Anpassungen verwenden. Die kinematischen Vorgänge, logischen Verknüpfungen und physikalischen Verhaltensweisen werden von dieser Software unterstützt und angeboten. So lassen sich die gelieferten Sensorwerte visuell darstellen und Aktoren ansteuern. Für das speziellen Umfeld einer PKW-Portalwaschanlage muss entsprechend die Sensorik angepasst werden. Unterschiedliche Materialien unter ungünstigen Umgebungsbedingungen wie Feuchtigkeit und Schmutz müssen robust erkannt werden. Entsprechend der Wirkungsweise der einzelnen Sensoren muss jeweils eine spezifische Evaluation durchgeführt werden, um eine fehlerfreie Funktion gewährleisten zu können [4].

In kürzester Zeit kann durch solch eine Simulation eine aussagekräftige Veranschaulichung der Resultate erzielt werden. Dies ist die Basis für einen fortschrittlichen Entwicklungsprozess, die vereinfachte Findung neuer Innovationen, aussagekräftige Entscheidungsunterstützung und Weiterentwicklung sowie Veranschaulichung bestehender Konzeptideen.

*Diese Arbeit wurde durchgeführt bei der Firma Alfred Kärcher GmbH & Co. KG, Winnenden

Erste Simulationsentwürfe wurden anhand realer Anforderungen analysiert und Szenarien skizziert. Dadurch ist es möglich eine realitätsnahe Simulation zu erstellen, ohne teure Bauteile für Testzwecke zu ordern und mühsam zu verbauen. Die Projektblöcke wurden anhand der vorhergehenden spezifischen Anforderungen modular entwickelt. Die in v-rep zu Verfügung gestellten Sensoren (siehe Abbildung 2), zeichnen sich dadurch aus, dass sie sich beliebig adaptiert lassen. Dies erlaubt es eigene Sensormodelle zu erstellen, die den auf den Markt verfügbaren Geräten in ihren Eigenschaften identisch sind.

Komplexe Szenarien (siehe Abbildung 3) lassen sich so auf einzelne Teilbereiche vereinfachen und explizit einzelne Problemstellen betrachten und bewerten. Weiterhin besteht die Option das Gesamtprojekt zu betrachten und beliebig komplex zu erweitern. Die Kollektion der unterschiedlichen Module und deren Ergebnisse veranschaulichen den weitreichenden Mehrwert einer Simulation und vermitteln eine objektive Einschätzung über aktuelle Sen-

soren und die damit verbundenen Konzeptideen.

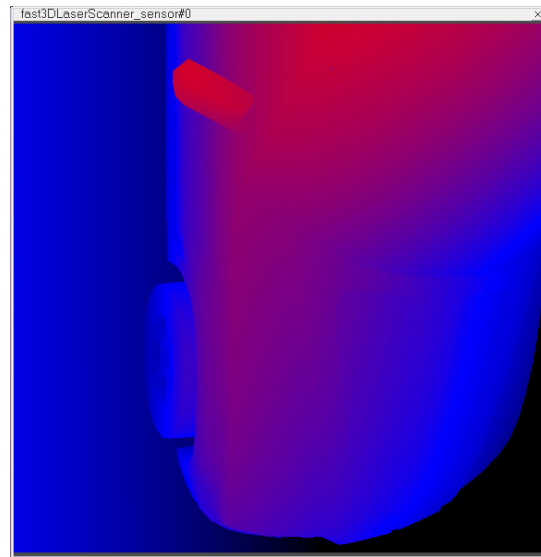


Abbildung 2: Messdaten eines 3D-Laserscanners

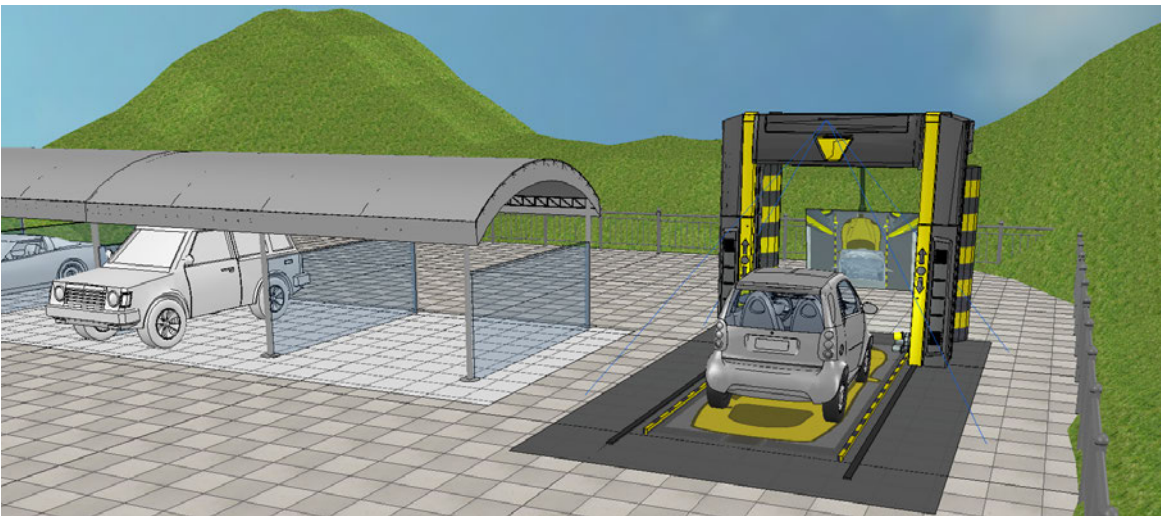


Abbildung 3: Testszenario mit verschiedenen Modulen

-
- [1] Alfred Kärcher GmbH & Co. KG www.kaercher.com/de/professional/hochdruckreiniger/heisswasser-hochdruckreiniger.html Stand 12.05.2016
 [2] Alfred Kaercher GmbH & Co. KG <https://www.kaercher.com/de/professional/waschanlagen.html> Stand 12.05.2016
 [3] Coppelia Robotics GmbH: www.coppeliarobotics.com Stand 02.05.2016
 [4] Jörg Hoffmann – Taschenbuch der Messtechnik 5. Auflage Hanser Verlag, Leipzig 2007, ISBN 3-446-40993-9

Bildquellen:

- Abbildung 1–3: Eigene Darstellung aus v-rep

Integration von 3D HMI-Systemen in moderne Industrie 4.0 Automatisierungs-Architekturen

Gina Frank*, Hermann Kull, Harald Melcher

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Die Bachelorarbeit, die bei der Robert Bosch GmbH in der Abteilung PA-ATMO1/SGS31 erstellt wird, handelt von einer Modernisierung der aktuell verwendeten HMI, mittels der 3D- und Virtual Reality Technik. Eine HMI oder ausgeschrieben „Human Maschine Interface“ ist die Schnittstelle zwischen einem Menschen und der zu bedienenden Maschine. Der aktuelle Stand der HMI im Bereich der Sondermaschinen bei Bosch sieht vor, dass die zu bedienende Maschine mit 2D Bildern auf der HMI abgebildet wird. Die Abbildung der Maschine hat keine nähere Bedeutung, da sie in der HMI nicht detailliert betrachtet werden kann und auch nur eine Seite der Maschine zu sehen ist. In Abbildung 1 ist die Bedienoberfläche der HMI, in der aktuellen Automatisierungs-Architektur von Bosch, zu sehen.

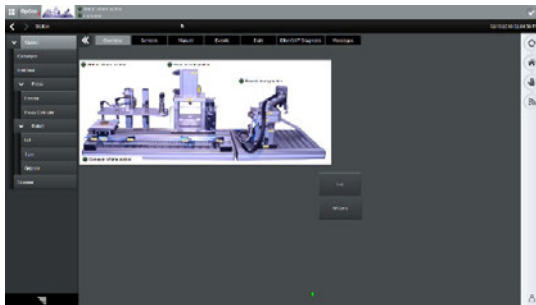


Abbildung 1: Bedienoberfläche der HMI bei Bosch

Es besteht zwar die Möglichkeit mehrere Abbildungen der Maschine in die HMI mit einzubinden, dies ist aber eine unpassende Lösung, da das Display auf der die Bedienoberfläche der HMI dargestellt wird, handlich und somit nicht so groß ist. Zu viele Bilder auf der Bedienoberfläche zerstören die gewünschte Übersicht der HMI. Abbildung 2 zeigt das „mPad“, auf der die HMI abgespielt wird.

Damit die Interaktion mit der HMI leichter und umgänglicher wird, ist das Ziel der Bachelorarbeit eine 3D HMI, die eine bessere Interaktion mit der HMI ermöglicht, zu konzipieren und entwickeln. Die Schwierigkeit hierbei liegt darin, diese als Baukastensystem aufzubauen. Der Vorteil eines Baukastensystems ist, dass der Anwender sich nur die Bausteine aus

dem Baukastensystem auswählt, die für seine 3D HMI von Nutzen sind. Dadurch werden Kosten eingespart, da der Mehraufwand, für jede Maschine eine 3D HMI individuell zu erstellen, wegfällt. Die Bausteine des Baukastensystems umfassen die einzelnen Features, welche mit der 3D HMI realisiert werden können. Dazu gehört beispielsweise die Navigation des 3D Modells mit verschiedenen Finger-Gesten, wie bei einem Smartphone. Mit diesem Navigations-Baustein wird zugelassen, dass der Anwender das Modell beliebig rotieren, transformieren und sich zu dem Modell hin oder weg bewegen kann.



Abbildung 2: „mPad“, Computer auf der die HMI bei Bosch abgespielt wird.

Ein weiterer wichtiger Baustein ist die Fehlervisualisierung einzelner fehlerhafter Objekte der Maschine. Mit diesem Feature ist es möglich, Maschinenteile der realen Maschine, die einen Fehler erzeugen (z.B. ein Roboterarm der sich verklemmt hat), in der HMI am 3D Modell farblich hervor zu heben. Außerdem hat man daraufhin die Möglichkeit, die Ansicht des Betrachters auf das fehlerhafte Objekt in der HMI zu fokussieren. Dies ist ein großer Vorteil, falls sich das entsprechende Objekt innerhalb der realen Maschine befindet und nicht von außen sichtbar ist. Über die virtuelle Fahrt zum Objekt ist es einfacher die Position des gesuchten, realen Objektes zu finden, anstatt Mühsam im Schaltplan der Maschine nach der genauen Position zu suchen. Um die HMI bedienen zu können, ist eine leicht verständliche grafische Oberfläche vorgesehen. Diese soll aus ei-

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

ner „Kopfleiste“ bestehen, die ein Drop-Down Menü enthält, in dem man seine integrierten Bausteine und deren Funktionalitäten auswählen und gegebenenfalls verschiedene Einstellungen durchführen kann. Um die 3D Visualisierung des Modells konfigurieren zu können, wurde als Entwicklungswerkzeug die Game Engine „Unity“ gewählt. Eine Game Engine ist ein Framework für die Programmierung von Computerspielen. Sie stellt Funktionen und Methoden bereit, für die Programmierung der Steuerung des 3D Modells. Der Vorteil von Unity ist, dass der Hersteller eine kostenlose Basisversion anbietet. Dies ermöglicht, dass Anwender ohne größere Anschaffungskosten für das Entwicklungswerkzeug 3D HMI Projekte realisie-

ren können. Unity wird nicht nur genutzt um das Baukastensystem zu entwickeln, sondern wird auch für die individuelle Projekterstellung verwendet. Dazu müssen lediglich ein bestehendes 3D Modell der Maschine und die gewünschten Bausteine aus dem Baukastensystem in ein Unity Projekt per Drag-and-Drop eingefügt werden. Im letzten Schritt wird die so erstellte 3D HMI in die bestehende HMI der Automatisierungs-Architektur von Bosch integriert. In Abbildung 3 ist ein 3D Modell einer Sondermaschine in Unity zu sehen. Ausblickend betrachtet ist diese Arbeit der Prototyp eines 3D HMI-Systems, welches durch das verwendete Baukastensystem einfacher mit neuen Funktionalitäten erweitert werden kann.

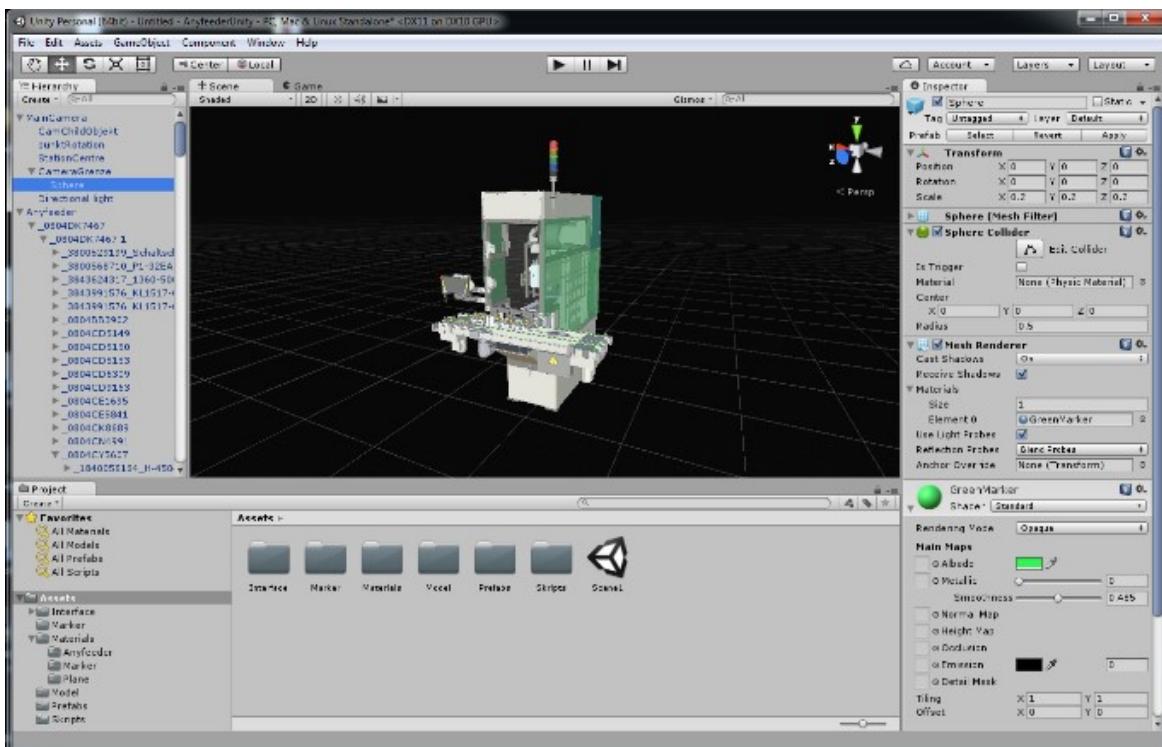


Abbildung 3: 3D Modell in Unity

[1] A. Schreck, „Die Rolle der HMI für die Industrie 4.0,“ VIERNULL Magazin, 2016.

Bildquellen:

- Abbildung 1,3: Gina Frank
- Abbildung 2: Bosch intern

Konzeption und Implementierung einer Konsolenapplikation in C# zur Erfassung von Lasersimulationsdaten in einer SQL-Datenbank einschließlich der auf den Simulationsdaten basierenden Entwicklung von Analysemethoden und Visualisierungselementen

Philipp Hornuß*, Reinhard Keller, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Produkttests dienen dazu, die Sicherheit und die Zuverlässigkeit von Geräten zu gewährleisten. Aus diesem Grund werden Produkte über unterschiedlichste Szenarien entsprechend getestet. Da jedoch die Testdurchführung aus Machbarkeits- und Kostengründen nicht immer in dem gebotenen Umfang möglich ist, werden Tests häufig über die Simulation relevanter Umgebungsverhältnisse durchgeführt. Diese bildet die Testumgebung unter realistischen Bedingungen softwaremäßig nach und liefert jene Daten, die für die Beurteilung der zu prüfenden Produkte erforderlich sind.

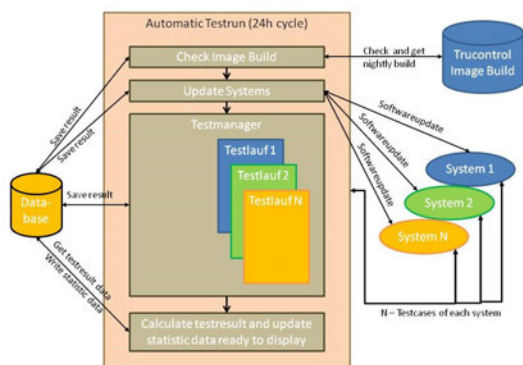


Abbildung 1: Aufbau u. Ablauf eines Testzyklus [1]

Die Aufgabe der vorliegenden Bachelorthesis bestand in der Auswertung und Visualisierung der aufgezeichneten Testdaten, die automatisch von mehreren Testsystemen erzeugt worden waren. Die Arbeit unterteilt sich in die beiden Teilgebiete:

- Parser-Entwicklung und
- Visualisierung der Simulationsdaten

Für die Bewertung der Testdaten ist eine standardisierte und automatisierte Auswertung sowie eine geeignete Darstellung der Ergebnisse erforderlich. Basis der Auswertung und Darstellung sind die Protokolldateien, wobei nur ein Teil der Informationen dargestellt

werden soll. Durch eine Analyse sollten die relevanten Daten ermittelt und dadurch eine Beschränkung und die benötigte Filterung ermöglicht werden.

Parser-Entwicklung

Ein Parser wird in der Informatik zur Zerlegung von Eingangsdaten sowie der Umwandlung in ein geeignetes Format zur Weiterverarbeitung verwendet [2]. Im vorliegenden Fall werden aus den Protokolldateien die Daten über entsprechende Funktionen gefiltert und zur Weiterverarbeitung in einer SQL-Datenbank bereitgestellt. Bezüglich der Datenbereitstellung gilt es im Wesentlichen zu beachten, dass die Daten nicht redundant sind und außerdem keine Inkonsistenzen aufweisen [3].

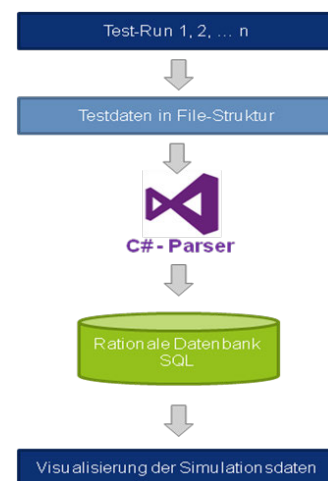


Abbildung 2: Modell zur Visualisierung der Testdaten

Die gefilterten Daten werden in der C#-Entwicklungsumgebung durch Objekte repräsentiert und anschließend in der SQL-Datenbank entsprechend abgelegt. Die Relationen zwischen den einzelnen Objekten wurden nach der ersten und zweiten Integritätsregel konzipiert, so dass die Eindeutigkeit und korrekte Zuordnung der Daten gewährleistet ist. Dieser

*Diese Arbeit wurde durchgeführt bei der Firma TRUMPF Laser GmbH, Schramberg

Aspekt ist hauptsächlich in Bezug auf parallel ausführbare Testläufe relevant. Der Parser *Trumpf.Log2Sql* wurde als Konsolenanwendung konzipiert, so dass er per Kommandozeile aufgerufen werden kann. Mittels der auf Basis des Lastenhefts implementierten Funktionen können über entsprechende Parametrisierung verschiedene Aktionen ausgeführt werden. Unzulässige Eingaben werden von der Software abgefangen und dem Benutzer einschließlich Hilfestellung angezeigt.

Visualisierung der Simulationsdaten

Ziel dieses Teils der Arbeit war die Darstellung der Simulationsdaten über eine universelle und standardisierte Visualisierungsumgebung. Die Datenbank stellt dabei die Basis zur Visualisierung dar. Über die vorhandenen Relationen können die gefilterten Simulationsdaten systematisch ausgewertet und dann entsprechend angezeigt werden. Unterschieden wird im Wesentlichen zwischen der Masterübersicht und den Detailansichten.

Die **Masterübersicht** bietet eine allgemeine Übersicht zu den Testläufen und umfasst neben den zuletzt ausgeführten *Runs* auch statistische Auswertungen und Diagramme.



Abbildung 3: Entwurf der Masterübersicht

Dies soll dem Anwender die Möglichkeit bieten die Testsituation auf den ersten Blick einzuschätzen und zu bewerten. Diagramme sollen dabei zum besseren Verständnis der Testdaten beitragen und das Ableiten von Prognosen unterstützen. Die Masterübersicht soll darüber hinaus den Entwicklern die Fortschritte des betroffenen Entwicklungsprozesses sowie eventuellen Verbesserungsbedarf bezüglich der Testläufe darstellen. So kann auch die Aufteilung der Aufgaben auf diejenige Teams oder Abteilungen eingegrenzt und effizienter gestaltet werden, die zu deren Behebung ge-

zielt einer Fehlerursache nachgehen müssen. In den **Detailansichten** können sämtliche Informationen zu allen Relationen angezeigt werden. Insbesondere in Fehlerfällen erscheint es sinnvoll, zur Lokalisierung von Fehlerursachen weitere Details abrufen zu können. Um in der Detailansicht eine strukturierte und gezielte Suche zu ermöglichen, wurden je nach Detailebene Filteroperationen eingebaut, die die Informationsmenge einschränken.

Abbildung 4: Beispiel: Filteroperator [4]

Bei der Konzeption der Visualisierung, speziell der Masteransicht, spielte die Betrachtung des jeweiligen Anwenders eine entscheidende Rolle. Aus Sicht der Testabteilung ist eine zyklusbezogene Ansicht ausreichend, um einen Überblick über den Status der Zyklen zu erhalten. Bezüglich den abteilungs- oder teambezogenen Anwendern hingegen, sind eher jene Daten relevant, die in die Aufgabengebiete der einzelnen Teams fallen.

Hinsichtlich der Visualisierung wurde in beiden Anwendersichten besonders auf den Usability-Aspekt und auf eine möglichst interessante Darstellung geachtet. Letzteres gilt überwiegend für die Masterübersicht. Ein wichtiges Ziel dabei ist es, die Aufmerksamkeit der Betrachter auf die dargestellten Informationen geeignet zu lenken.

Im Hinblick auf die Usability soll eine intuitive Bedienung auf allen Visualisierungsebenen möglich sein. Im Wesentlichen wird dies durch die Filteroperationen und deren Gestaltung erreicht.

[1] TRUMPF Laser

[2] Wikipedia (Hrsg.): Parser. <https://de.wikipedia.org> (06.05.2016).

[3] Kleuker, S.: Grundkurs Datenbankentwicklung. Vieweg+Teubner Verlag, Wiesbaden, 2. Auflage 2011.

[4] TRUMPF Laser

Bildquellen:

- Abbildung 1,4: TRUMPF Laser
- Abbildung 2,3: Philipp Hornuß

Erkennung von dynamischen Freiraumgesten mit Hilfe des Maschinellen Lernens

Elif Isildar*, Manfred Dausmann, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Mit einer künstlichen Intelligenz ausgestattet, im freien Raum berührungslos Gesten von Sensoren erfassen und darauf reagieren ist ein gewaltiger Schritt in der Evolution der Human Machine Interaction (HMI). Interaktion per Gestik als Bestandteil der HMI ist ein Trend, der in den letzten Jahren zunehmend in der Welt der Consumer Elektronik (CE) zu beobachten ist. Was für ein enormes Potenzial die Gesteninteraktion dabei auch im Innenraum von Fahrzeugen darstellt, wurde auf der CES 2016 in Las Vegas von allen größeren Automobilherstellern mit ihren Showcars vorgeführt. Das Radio berührungslos bedienen, das Schiebedach mit einer Wischgeste öffnen und viele weitere Bedienmöglichkeiten sind keine Zukunftsvisionen mehr[1][2].



Abbildung 1: Der BMW 7 (2015) mit Gestenerkennung

Die zunehmende Verfügbarkeit kostengünstiger und miniaturisierter Sensoren hat zu einer weiteren Verbreitung der Gestensteuerung geführt. Die steigende Akzeptanz der Gestensteuerung beim Anwender und ihre Durch-

dringung in die unterschiedlichsten Anwendungsbereiche wirken sich auf die Kundenerwartungen im Automotiv-Umfeld deutlich aus. Im Center of Competence (CoC) HMI der Robert Bosch GmbH, werden solche neuen Technologien für die Interaktion mit Infotainmentsystemen untersucht. Der Hauptzweck ist es den Fahrer, der sich trotz steigender Komplexität der Systeme im Fahrzeug auf das Geschehen im Straßenverkehr konzentrieren muss, zu entlasten. Dafür wird versucht, alternativ die Bedienung von physischen Elementen abzulösen bzw. zu ergänzen und für einfache Bedienoperationen Gesten zu benutzen, da diese keine große kognitive Belastung für den Fahrer darstellen, intuitiv in ihrer Anwendung sind und ein Blickkontakt zwangsweise nicht mehr notwendig wird.

Diese Bachelorarbeit beschäftigt sich mit der Konzipierung und Implementierung einer Gestenerkennung von dynamischen Freiraumgesten mit Hilfe der Methoden des maschinellen Lernens. Das Ziel der Arbeit ist es, ergänzend zu vorhandenen konventionellen Algorithmen zur Gestenerkennung auch maschinelle Lernverfahren anzuwenden und einen möglichst effizienten, robusten und ressourcenschonenden Algorithmus zu finden. In einem ausführlichen Benchmark sollen diese beiden Ansätze im Anschluss auf Performance, Robustheit, Echtzeitfähigkeit und unter anderem auch auf Anpassungsmöglichkeiten untersucht werden.

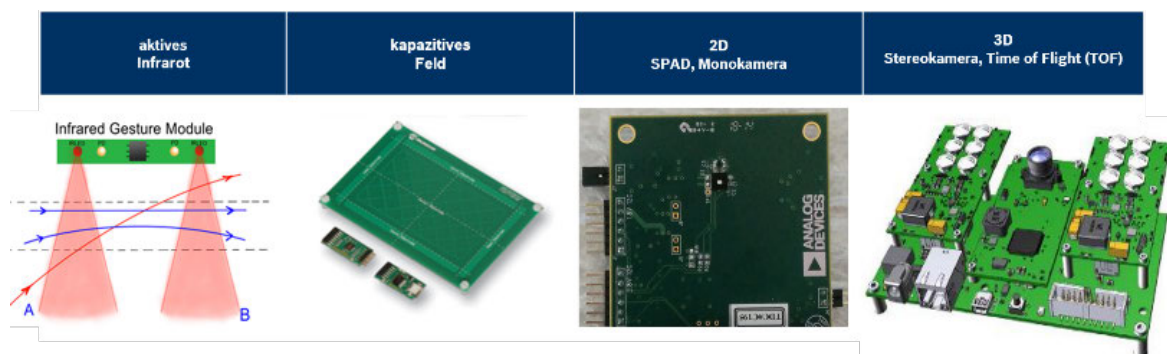


Abbildung 2: Übersicht Sensoren zur Gestenerkennung mit unterschiedlichen Eigenschaften

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

Dazu werden die nach heutigem Stand der Technik bereits im Auto eingesetzte Sensoren zur Gestenerkennung vorgestellt, insbesondere die Proximity-Sensoren im aktiven Infrarot (IR) Bereich. In Abbildung 2 sind unterschiedliche Sensoren abgebildet, die bereits zur Gestenerkennung eingesetzt werden. Weiterhin wird die prinzipielle Signalverarbeitung und Datenerhebung dieser Sensoren erläutert auf denen die beiden Ansätze basieren.

In dieser Arbeit soll die Infrarot Sensorik als eine kostengünstige Alternative der Gestenerkennung untersucht werden. Weiterhin werden verschiedene maschinell lernende Algorithmen analysiert, unter anderem die Template- und merkmalsbasierten Verfahren. Daraus hat sich ergeben, dass in der weiteren Arbeit die merkmalsbasierten Verfahren eingesetzt werden. Um Algorithmen für den IR-Sensor zu entwickeln, bedarf es zuerst der Definition der Gesten. Hierfür wurde ein Gestenkatalog von ausgewählten Gesten erstellt von denen Beispiele mit ihren zugehörigen Signalverläufen in Abbildung 3 zu sehen sind.

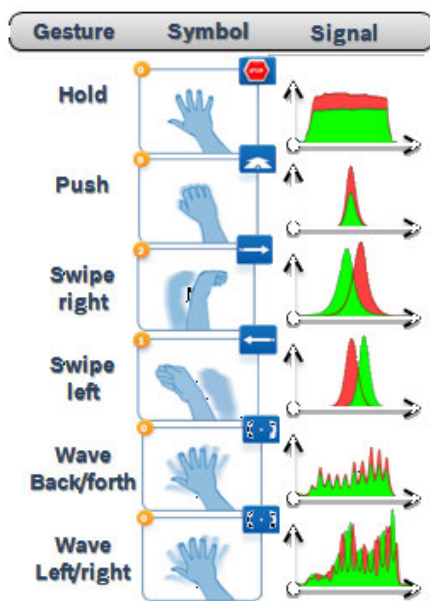


Abbildung 3: Gestenkatalog eines 2-Kanal-Infrarotsensors

Auf Basis der Definition des Gestenkatalogs, werden die Signaldaten mit der System Design Software LabView von National Instruments zur Erkennung aufgenommen und bearbeitet. In einem aufwändigen Datenaufzeichnungsaufbau werden alle Gesten aus dem Gestenkatalog in einem realitätsnahen Mockup-Aufbau von mehreren Probanden ausgeführt und über mehrere Wiederholungen hinweg aufgezeichnet und abgespeichert. Daraufhin werden die aufbereiteten Rohdaten innerhalb eines kontinuierlichen Signals mit Gesten markiert und zur Weiterverarbeitung für die Trainingsdatensätze zur Merkmalsextraktion abgespeichert.

Nach der Merkmalsextraktion können diese Trainingssets nun mit einem ausgewählten Klassifikator trainiert werden. Beim überwachten maschinellen Lernen erfolgt eine Klassifizierung anhand von bereits identifizierten Trainingsdaten um neue, nicht identifizierte Daten einer Bewertung zu unterziehen. Nach dem Trainingsvorgang kann die Genauigkeit und die Zuverlässigkeit des entwickelten Algorithmus evaluiert und eventuell mit Reduktion oder Neustrukturierung der Trainingsdaten angepasst werden. Bei der Evaluierung und der Ergebnisbewertung wird der ursprüngliche Algorithmus des IR-Sensors im Detail mit dem neuen maschinell lernenden Algorithmus verglichen. Es werden für beide Algorithmen dieselben Trainingssets verwendet, um dieselbe Ausgangsbasis zu haben. Bei der Bewertung der Maßstäbe dieser beiden Algorithmen werden unter anderem die Erkennungsgüte, Echtzeitfähigkeit und die Generik untersucht. Wichtig ist es herauszufinden, wie sich ein selbst lernender Algorithmus auf unterschiedliche Anwendungsfälle im Automotive-Umfeld anpassen lässt. Unter anderem spielen Überlegungen eine Rolle, wie ein effizientes Training der Algorithmen bei einer Neupositionierung der Sensoren (z.B. anderes Fahrzeugmodell) oder neue Features integriert werden können.

[1] A Wave Is All It Takes – IAV Automotive Engineering, <https://www.iav.com/us/automotion-magazine/automotion-2014-issue-03/wave-all-it-takes?r=de&n=12655>

[2] VW Golf <http://www.autobild.de/artikel/vw-golf-facelift-e-golf-touch-ces-2016-sitzprobe-5521782.html>, zuletzt zugegriffen am Donnerstag, 12. Mai 2016 09:10:25

Bildquellen:

- Abbildung 1: http://www.focus.de/auto/news/verkehr-mit-einem-wisch-steuern-wir-das-auto-bald-mit-gesten_id_4930933.html, zuletzt zugegriffen am Donnerstag, 12. Mai 2016 09:12:16
- Abbildung 2,3: Robert Bosch GmbH, Center of Competence HMI

Die Clean Code Developer-Initiative

Maria Koblova*, Hans-Gerhard Groß, Joachim Goll

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Wie man professionell entwickelt, ist Inhalt des Buchs „**Clean Code: A Handbook of Agile Software Craftsmanship**“ von Robert C. Martin (2008) [1]. Eine der Zielrichtungen dieses Buches – die Prinzipien und Praktiken für eine erfolgreiche Entwicklung – führte im Jahre 2009 zur Gründung der **Clean Code Developer-Initiative** von Stefan Lieser und Ralf Westphal. Diese beiden versuchen, in Form dieser Initiative die Idee einer professionellen Softwareentwicklung zu beleben und die Einhaltung der Prinzipien und Praktiken für sauberen Code zu erleichtern.

Die Clean Code Developer-Initiative befasst sich speziell mit Verfahren und Techniken zur Erstellung eines guten Quellcodes mit wenigen inneren Abhängigkeiten, damit ein Code wandelbar (änderbar und erweiterbar) ist.

Ein Clean Code Developer folgt einem von Lieser und Westphal festgelegten Wertesystem, welches die in Abbildung 1 dargestellten **Qualitätseigenschaften** umfasst. Diese Qualitätseigenschaften korrelieren nicht mit Qualitätsstandards.



Abbildung 1: Das Wertesystem eines Clean Code Developer

Evolvierbarkeit und Korrektheit sind sogenannte innere Qualitäten der Software. Produktionseffizienz ist eine äußere Qualität und Reflexion eine Praktik des Softwareentwick-

lungsprozesses.

Evolvierbarkeit

Evolvierbarkeit (engl. Evolvability) ist eine innere Qualität der Software, die zum Ziel hat, dass für eine Software eine längere Lebensdauer zu erwarten ist, als dies in der Vergangenheit der Fall war. Diese innere Qualität lässt sich nachträglich nicht mehr erreichen. In der Regel wird sie nur erzieht, wenn die Architektur eines Systems von Beginn an durchdacht ist, insbesondere wenn Abhängigkeiten der verschiedenen Teile stark reduziert wurden. Die Eigenschaft der Evolvierbarkeit beschreibt, mit welchem Aufwand und Erfolg neue Features in eine Software eingebracht werden können oder Änderungen vorhandener Features durchgeführt werden können.

In der Praxis werden Änderungen umso teurer, je später sie realisiert werden. Am Anfang einer Entwicklung sind Änderungen relativ preiswert zu realisieren. Am Ende können Änderungen meist gar nicht mehr realisiert werden. Dies kann einerseits daran liegen, dass die Abhängigkeiten in der Software zu hoch sind, andererseits auch daran, dass der Überblick über die Abhängigkeiten aufgrund ihrer hohen Zahl verloren gegangen ist. Wird in einem solchen Falle die Software an einer bestimmten Stelle geändert, so muss man auf Grund der Abhängigkeiten, durch die viele Teile der Software gekoppelt sind, den Code an vielen weiteren Stellen ebenfalls abändern. Meist entstehen hierbei wiederum neue Abhängigkeiten – das System altert, bis es schließlich dadurch „stirbt“, dass es durch ein neues System abgelöst wird.

Die Evolvierbarkeit einer Software ist umso höher, je einfacher diese Software abgeändert werden kann. Bei guter Evolvierbarkeit kann eine Software eine lange Zeit weiterentwickelt werden.

Korrektheit

Software muss ihre funktionalen und nicht funktionalen Anforderungen erfüllen. Das sind äußere Qualitäten. Aber auch die Korrektheit der Software selbst, also die innere Qualität, muss stimmen.

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

Produktionseffizienz

Nicht nur die Funktionalität in Umfang und Qualität, sondern auch die Entwicklungszeit und die Kosten spielen für den Erfolg eines Projekts eine wichtige Rolle. Damit muss die Produktion von Software effizient sein. Dies bedeutet, dass eine Software über Jahre weiterentwickelbar sein sollte, anstatt frühzeitig durch ein neues Softwaresystem abgelöst werden zu müssen.

Reflexion

Produkte, Prozesse und der Stand des eigenen Wissens sind zu überdenken, damit hierfür Verbesserungen vorgenommen werden können. Dabei ist eine Reflexion auf allen Ebenen erforderlich.

Bewertung des Wertesystems der Clean Code Developer-Initiative

Vorteile des Wertesystems von Lieser und Westphal sind:

- Die Tatsache der schriftlichen Fixierung eines Wertesystems,
- die gemeinsame Methodenwelt und Sprache aller Entwickler,
- die Modularität durch die Einteilung in Grade,
- das besprochene Ziel einer kontinuierlichen Verbesserung und
- die Beachtung der Selbstbestimmung der Mitarbeiter.

Die von Lieser und Westphal herausgegriffenen Werte sind nicht falsch, ihnen fehlt jedoch die wissenschaftliche Basis. Korrektheit und Evolvierbarkeit im Sinne von inneren Qualitäten sind nur für Entwickler interessant. Weder gehen Lieser und Westphal auf andere innere Qualitäten ein, noch auf das Vorhandensein von äußeren Qualitäten.

Produktionseffizienz ist eine wichtige Eigenschaft für den Erfolg eines Projektes. Sie ist eine äußere Qualität und beruht auf einer geeigneten Werkzeugunterstützung. Noch wichtiger ist allerdings die Bedeutung des Menschen für den Erfolg eines Projektes. Auf diesen Faktor gehen Lieser und Westphal nur sehr unvollständig und an sehr untergeordneter Stelle ein.

Auch die Reflexion ist ein wichtiger Bestandteil heutiger Prozesse der Softwareentwicklung. Sie wird jedoch nur an untergeordneter Stelle erwähnt und nur unzureichend begründet. Daher bleibt abschließend festzustellen, dass das Wertesystem der Clean Code Developer-Initiative lediglich eine Orientierungshilfe ist – jedoch ist das Wertesystem anfechtbar.

Tugenden zur Umsetzung des Wertesystems

Die Tugenden zur Umsetzung der Werte eines Clean Code Developers nach Lieser und Westphal umfassen Prinzipielles und Praktisches.

Prinzipielle Tugenden sind:

1. Schätze Variation (Value Variation (VV))
2. Tue nur das Nötigste (Do Only What's Necessary (DOWN))
3. Isoliere Aspekte (Isolate Aspects (IA))
4. Minimiere Abhängigkeiten (Minimize Dependencies (MD))
5. Halte Versprechen ein (Honor Pledges (HP))

Praktische Tugenden sind:

1. Nehme Unsicherheit an (Embrace Uncertainty (EU))
2. Fokussiere (Focus (F))
3. Wertschätze Qualität (Value Quality (VQ))
4. Stelle Dinge fertig (Get Things Done (GTD))
5. Halte Ordnung (Stay Clean (SC))
6. Bleib am Ball (Keep Moving (KM))

Das Wertesystem enthält die Tugenden für die tägliche Arbeit eines Clean Code-Entwicklers. Diese Tugenden sind relativ abstrakt. Die Bausteine, welche die Erfüllung dieser Werte fördern, sind Prinzipien und Praktiken.

Prinzipien unterstützen die Einhaltung des Clean Code-Wertesystems. Sie sind die zugrundeliegenden Gesetzmäßigkeiten für die Strukturierung der Software. Sie sollten nach Möglichkeit eingehalten werden, können aber je nach Situation und Projekt auch in Frage gestellt werden.

Praktiken sind nützliche Regeln für den Alltag in Form von Handlungsanweisungen. Nach der Clean Code-Initiative sieht man dem Code nicht immer an, ob eine Praktik befolgt wird.

Die Clean Code Developer-Grade

Für Clean Code Developer gibt es Grade als Lerneinheiten. Diese Grade werden im Folgenden in aufsteigender Reihenfolge genannt: schwarz, rot, orange, gelb, grün, blau, weiß. Die Grade drücken an sich keinen Wert aus. Die Einführung von Graden ist als didaktisches Hilfsmittel zur Bewältigung der Komplexität der vorhandenen Prinzipien und Praktiken zu sehen, um Lerneinheiten auszuweisen, mit deren Hilfe man sich stufenweise ausbilden kann.

Jeder Interessent (schwarzer Grad) sollte mit dem roten Grad beginnen. Dieser bildet den Einstieg in den Zyklus der Grade. Jedem Grad liegen eigene Prinzipien und Praktiken zugrunde, deren Einhaltung täglich in einer Rückschau geprüft werden sollte. Schafft es ein Entwickler hierbei, 21 Tage ohne Defizit zu arbeiten, so schreitet er zum nächsten Grad voran [2]. Die empfohlene Durchlaufreihenfolge durch die verschiedenen Grade ist im folgenden Bild dargestellt:

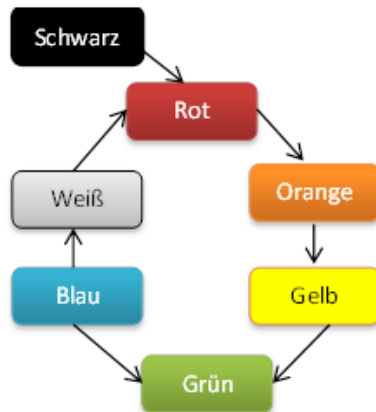


Abbildung 2: Empfohlene Durchlaufreihenfolge

Der Entwickler durchläuft die verschiedenen Grade bis weiß und beginnt dann wieder von vorne. Dies dient der Verfeinerung

der verschiedenen Aspekte eines Grades. Beim nächsten Durchlauf werden die Prinzipien und Praktiken eines Grades tiefer gelernt als beim ersten Mal.

Anforderungen an Professionals

Ein professioneller Softwareentwickler verkörpert Qualität und erzeugt Qualität. Dies zeigt sich in seiner Weitsicht, die das Richtige zu tun erlaubt, seiner Intention, sich fit zu halten, um auch in der Zukunft bestehen zu können, und in dem Wert seiner Produkte. Er reflektiert systematisch nicht nur über sein Produkt, sondern auch über den (automatisierten) Entwicklungsprozess samt Werkzeugen und über sein eigenes Wissen zum Zweck der Fortbildung.

Für professionelle Softwareentwickler wird von der Clean Code Developer-Initiative gefordert:

- **Beherrschung der Grundlagen ihres Fachs** -> Studium
- **Tiefes Wissen in einem erkennbar abgegrenzten Teilbereich** -> erwirbt man überwiegend erst im Beruf
- **Kontinuierliche Fortbildung** -> rasche Weiterentwicklung des Wissens, Anlass für die Existenz von Fachgesellschaften wie ACM oder GI
- **Sorgfalt und verantwortliches Handeln** -> Kunden können sich selbst kaum schützen; Software hat oft Gefährdungspotential.

[1] Martin, Robert C., „Clean Code: A Handbook of Agile Software Craftsmanship“, Prentice Hall, 2008 ISBN 0- 13-235088-2

[2] <http://clean-code-developer.de/> (Aufgerufen: 15.03.2016)

Bildquellen:

- Abbildung 1,2: Eigene Darstellung

Entropiequellen zur Erzeugung kryptografisch sicherer Zufallszahlen in eingebetteten Systemen

Manuel Köhler*, Walter Lindermeir, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Zufallszahlen spielen in der Kryptografie eine zentrale Rolle. Die Sicherheit gängiger Verfahren wie dem RSA-Kryptosystem und dem *Elliptic Curve Digital Signature Algorithm* für die Verschlüsselung und Authentifizierung von Nachrichten basiert auf einer guten Quelle für zufällige Geheimnisse, die zu erraten einem Angreifer nicht möglich sein darf. Um dies zu gewährleisten, existieren Algorithmen, welche basierend auf einem internen Zustand praktisch beliebig viele Werte erzeugen, die statistisch alle Anforderungen an eine zufällige Zahlenfolge erfüllen. Diese Algorithmen werden als *Pseudorandom Number Generators* (PRNGs), also „Pseudozufallszahlengeneratoren“ bezeichnet. Die grundlegende Problematik aus kryptografischer Sicht ist jedoch, dass, während die Ausgabe eines guten PRNG zwar zufällig „aussieht“, diese vollständig vom internen Zustand des Generators abhängt. Dementsprechend sind die so erzeugten Geheimnisse problemlos vorherzusagen, wenn dieser Zustand bekannt ist. Selbst das Wissen über Teile des Zustands kann einen Angriff erheblich erleichtern. Dies kann behoben werden, indem man darauf achtet, dass der anfängliche Zustand des PRNG sich aus Zufallswerten zusammensetzt, die durch einen Angreifer unmöglich bestimmt werden können. Derartige Werte können zum Beispiel aus nicht vorhersehbaren physikalischen Vorgängen stammen. Die Zufälligkeit einer solchen „Entropiequelle“ wird durch die sogenannte „Entropie“ quantifiziert, die eine Angabe darüber macht, mit welcher maximalen Wahrscheinlichkeit das Ra-

ten eines Wertes von dieser Quelle erfolgreich sein wird. Hohe Entropie bedeutet also für eine Quelle, dass ihre Ausgabewerte sehr zufällig verteilt sind und nicht stark zu wenigen, immer gleichen Werten tendieren. Wäre dies der Fall, könnte ein Angreifer mit dem entsprechenden Wissen stets den wahrscheinlichsten Wert annehmen und somit seine Erfolgswahrscheinlichkeit im Vergleich zum blinden Raten unter Umständen erheblich verbessern. Weiter ist Entropie ein rein statistisches Maß, das stets relativ zum Wissen eines Angreifers gesehen werden muss: Der Inhalt eines Pakets besitzt für den Postboten sehr viel Entropie, für den Empfänger aber vermutlich weniger und für den Absender überhaupt keine. So kann es sein, dass gewisse Entropiequellen von außen gemessen werden können oder mit anderen, dem Angreifer bekannten Größen zusammenhängen.

In Computern wird häufig der Nutzer als Entropiequelle verwendet. Zum Beispiel besitzen die Zeitabstände zwischen Tastendrücken, Mausbewegungen und Klicks eine oft schon hinreichende Menge an Entropie. Auch Verzögerungszeiten bei Festplattenzugriffen, die durch deren kontinuierliche Rotation Variationen aufweisen, sind eine gute Quelle für Entropie. In eingebetteten Systemen sind all diese Quellen nicht vorhanden. Es müssen also Vorgänge gefunden werden, die einerseits hohe Entropie besitzen und andererseits nicht durch einen Angreifer manipuliert oder aus anderen Informationen, die einem Angreifer zur Verfügung stehen, bestimmt werden können.

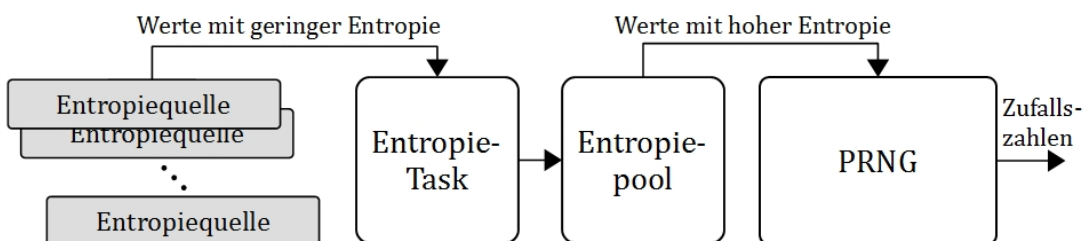


Abbildung 1: Vereinfachter Aufbau des Softwaremoduls zur Erzeugung kryptografisch sicherer Zufallszahlen

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

Diese Arbeit befasst sich zum einen mit dem Entwurf und der Implementierung einer Architektur zur Ansammlung von Entropie aus einer beliebigen Zusammensetzung von Quellen und der Erzeugung von Zufallszahlen aus dieser. Zum anderen geht es um die Bestimmung und Bewertung von Vorgängen, die üblicherweise in Mikrocontrollern, also den Prozessoren in eingebetteten Systemen, vorhanden sind und als Entropiequellen verwendet werden können.

Abbildung 1 zeigt einen vereinfachten Aufbau der entwickelten Software. Ein Hintergrundprozess (Entropie-Task) fragt zyklisch die vorhandenen Entropiequellen ab und fügt deren Ausgaben einem „Entropiepool“ hinzu, in dem die Entropie der verschiedenen Quellen gesammelt wird. Der Grundgedanke dabei ist, dass der Inhalt des Pools immer schwieriger zu bestimmen wird, je mehr Werte er enthält. Wird nun der eigentliche PRNG genutzt, wird der Inhalt des Entropiepools in den Zustand des PRNG überführt und gewährleistet so die notwendige kryptografische Sicherheit.

Zusätzlich wurde eine Suite aus Tests zusammengestellt, mit denen die Entropie einer Quelle aus einer Reihe ihrer Werte statistisch bestimmt werden kann. Dabei sind neben den üblichen Entropiemaßen nach [1, 2] auch weitere vom *National Institute for Standards and Technology* (NIST) entwickelte Tests (vgl. [3]) implementiert, die nicht nur die bloße Verteilung der Werte, sondern auch zeitliche Abhängigkeiten mit einbeziehen.

Speziell in Mikrocontrollern existiert dank der Möglichkeiten zum direkten Zugriff auf deren Hardwarefunktionen eine Vielzahl an Vorgängen, die als Entropiequellen nutzbar sind. Ein Beispiel hierfür sind die Werte von Speicherzellen (SRAM) direkt nach dem Start des Systems. Diese kennen nur zwei stabile Zustände („0“ und „1“), behalten ihren Wert jedoch nicht, wenn sie die Spannungsversorgung verlieren. Dementsprechend „kippt“ die Zelle beim Anlegen der Spannungsversorgung in einen der beiden Zustände, wobei aufgrund der großen Menge an möglichen Einflüssen auf diesen Vorgang und der Komplexität der Speicherarchitektur eine Vorhersage praktisch unmöglich ist. Somit steht, schon bevor das System überhaupt aktiv wird, eine schier unbegrenzte Menge an Entropie zur Verfügung. In Abbildung 2 ist eine Reihe von Werten für einen Ausschnitt des Speichers eines Mikrocon-

trollers dargestellt. Dabei sind jeweils 8 Zellen, also ein Byte, zu einem Bildpunkt zusammengefasst, wobei der Bildpunkt schwarz ist, wenn alle Zellen den Wert „0“ haben und weiß, wenn alle eine „1“ enthalten. Eine Spalte entspricht einer Reihe von Werten für ein bestimmtes Byte aus mehreren Startvorgängen. Es ist deutlich zu erkennen, dass manche Bytes immer den gleichen Wert besitzen, während andere unregelmäßig zwischen zwei oder mehr Werten wechseln. Man kann nun beliebig viele Speicherzellen im Entropiepool zusammenfassen, wobei diese Unregelmäßigkeiten erhalten bleiben. Werden dabei genug Zellen verwendet, erhält man eine Quelle mit sehr hoher Entropie, die aufgrund ihrer physikalischen Grundlage nicht vorherzusagen ist.

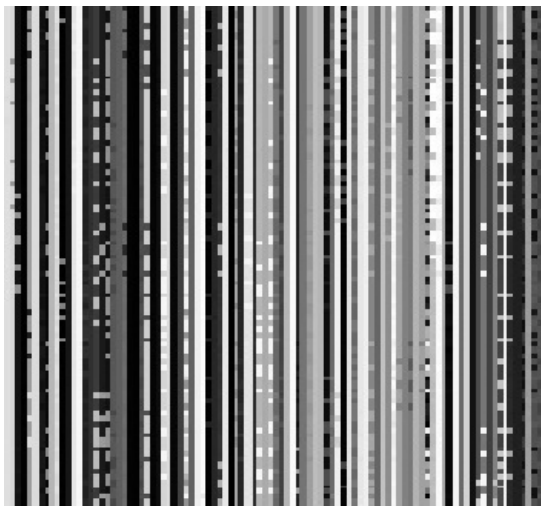


Abbildung 2: Reihe von Werten für verschiedene Bytes im SRAM eines Mikrocontrollers direkt nach Anlegen der Versorgungsspannung

Weiter lassen sich beispielsweise Ungenauigkeiten der Oszillatoren, die den Prozessoren als Taktsignal dienen, als Entropiequelle nutzen. Diese sollen zwar so konstant wie möglich arbeiten, weisen aber trotzdem teils zufällige Variationen in ihrer Frequenz auf, die sich, wenn entsprechend gemessen, auch hervorragend als Entropiequelle eignen. Auch Rauschen in Sensormessungen besitzt Potential für diese Anwendung, wobei aber darauf geachtet werden muss, dass ein Angreifer dieses nicht maßgeblich beeinflussen kann.

[1] Claude E. Shannon. Communication Theory of Secrecy Systems, In: Bell Systems Technical Journal 28 Nr. 4, S. 656–715, 1949

[2] Alfred J. Menezes, Paul C. van Oorschot. Handbook of Applied Cryptography, CRC press, 1996

[3] Meltem S. Turan, Elaine Barker, John Kelsey, Kerry A. McKay, Mary L. Baish, Mike Boyle. Recommendation for the Entropy Sources Used for Random Bit Generation, NIST Special Publication 800–90B (second DRAFT), 2016

Analyse von Mechanismen zur Synchronisation von seriellen Kommunikationsdaten und analogen Signalen einschließlich prototypischer Integration in die standardisierte Hardware-/Softwarearchitektur für Prüfsysteme bei Festo

Ernst Lehmann*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

In der klassischen Messtechnik werden physikalische Größen als analoge Werte, als digitale Pulsfolgen wie z.B. bei PWM-Signalen oder als Gruppen von Impulsen wie bei AB-Signalen abgebildet. Diese können mit gängigen Datenerfassungskarten zeit-synchron erfasst bzw. erzeugt werden. Bibliotheken für den gesamten Funktionsumfang sind in der standardisierten Hardware-/Softwarearchitektur TEaM (Test Environment and Modules) von Festo enthalten. In der Automatisierungstechnik werden jedoch viele Größen als digitale Werte einerseits produktübergreifend über Feldbusse wie z.B. CAN, IO-Link oder ein Industrial-Ethernet-Protokoll, andererseits produktintern über SPI, I2C oder proprietäre Busse von Festo übertragen. Die benötigten Master- oder Monitorkomponenten sind am Markt verfügbar oder beruhen auf Eigenentwicklungen.

Ziel der vorliegenden Bachelorarbeit war eine allgemeingültige Analyse der Möglichkeiten zur Synchronisation dynamischer Signale verschiedener digitaler und analoger Systeme, sowie deren prototypische Integration in die bestehende TEaM-Architektur zur Stimulation, Erfassung und Darstellung der Signale.

Signale und Module

Zur Abbildung der physikalischen Signale im Messrechner werden sogenannte virtuelle Signale verwendet. Diese setzen sich aus einzelnen Sample-Punkten, welche je einen Signalwert und einen eindeutigen Zeitstempel besitzen, zusammen. Jede Hardware, die die Möglichkeit zur Zeitstempelung besitzt, bezieht alle ihre Signale auf die eigene interne Uhr, wodurch diese zueinander synchronisiert sind. Werden jedoch mehrere Module im Prüfsystem verwendet, kann es zu Zeitfehlern durch Zeitabweichungen der Moduluhren kommen, woraufhin sich virtuelle Signale nicht korrekt auf einer gemeinsamen Zeitachse darstellen lassen. Des Weiteren sind der Verbund aus Hardwaremodulen und der zugehörige Messrech-

ner über ein Kommunikationssystem verbunden, das ebenfalls zeitlichen Ungenauigkeiten unterliegt. Dies soll jedoch nicht zu zeitlichen Fehlern bei Mess- und Stimulationssignalen führen, was eine Synchronisation der Module erforderlich macht, die auf unterschiedliche Weise realisiert werden kann.

Einbindung des Prüflings

Eine entscheidende Rolle bei der Synchronisation spielt die Einbindung des Prüflings in das Prüfsystem. Der Prüfling wird vom Prüfsystem umschlossen, d.h. durch Stimulation und Erfassung der Ausgangsgrößen kann die Reaktion des Prüflings erfasst und ausgewertet werden. Die Signale können hierbei in physikalischer Form, als Nachricht über ein Busprotokoll oder gemischt vorliegen.

Zeitfehler im Prüfsystem

Bei der Stimulation und Erfassung von Ausgangsgrößen eines Prüflings durchlaufen die Signale, wie bereits beschrieben, mehrere Soft- und Hardwareschichten. Beim Durchlaufen der Schichten können je nach Art der verwendeten Hard- und Software sowohl konstante Verzögerungen (Delay) als auch inkonstante Verzögerungen (Jitter) auftreten, was zu vielfältigen Zeitfehlern in den Signalen führen kann.

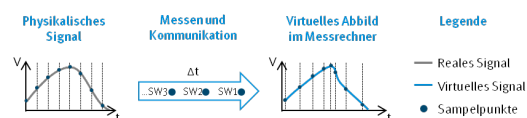


Abbildung 1: Signalverzerrung durch Jitter

Die Klassifizierung der Zeitfehler erfolgte anhand der Auswirkungen auf die Signale, sowie über die Ursache ihrer Entstehung:

- Allgemeine Zeitfehler
- Bei Systemen mit Zeitstempelung
- Zeitfehler bei Buskommunikation

*Diese Arbeit wurde durchgeführt bei der Firma Festo AG & Co. KG, Esslingen

Mechanismen zur Synchronisation

Zur Realisierung der Synchronisation stehen je nach verwendeter Hardware verschiedene Möglichkeiten zur Verfügung. Eine Einteilung in Hardwaremodule mit und ohne Zeitstempelung bildet die beiden Hauptkategorien.

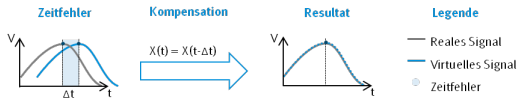


Abbildung 2: Wirkung der Synchronisation bei Verzögerung im Signalpfad

Mittels eines *Test Access Point* (TAP) wird der Datenfluss einer Ethernetkommunikation voll-duplex aufgezeichnet und die Prozesswerte zeitgestempelt, ohne dabei aktiv in die Kommunikation einzugreifen. Die in Verbindung mit TAPs verwendete Software zur Auswertung des Datenflusses wird als Monitor bezeichnet. Ihre Aufgabe ist es, die in den Nachrichten befindlichen Prozesswerte zu filtern und in eine nutzbare Form zu bringen. Durch TAPs wird die Problematik der starken Verzögerung und dem teilweisen Verlust von Nachrichten ungepufferter PC-Buskarten behoben.



Abbildung 3: Wandeln der Prozessdaten zu Samplepunkten im Monitor

Bei der Stimulation eines Prüflings ist das Ausgangssignal des Stimulators meist mit einer zeitlichen Verzögerung belastet. Durch Rückmessung dieses Ausgangssignals zusammen mit den Ausgängen des Prüflings führt diese Verzögerung zu keinem Zeitfehler und es können genauere Aussagen zur Reaktionszeit des Prüflings gemacht werden. Um welche Art von physikalischer Größe es sich handelt, ist dabei ohne Bedeutung, solange ein Sensor zur Erfassung der Größe bereitsteht.

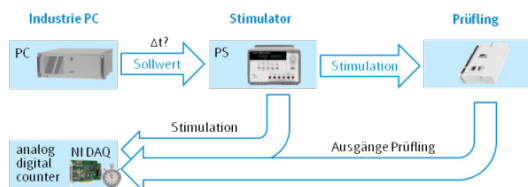


Abbildung 4: Rückmessung bei Stimulation durch physikalische Größen

Ein weiteres wichtiges Kriterium zur Bewertung einer Methode ist ihre herstellerübergreifende Verwendbarkeit. Hierbei hat sich ein digitales Triggersignal als einfach umsetzbarer und häufig ausreichender Mechanismus herausgestellt, da bereits viele Module digitale Ein- und Ausgänge besitzen und der Aufbau beliebig erweiterbar ist.

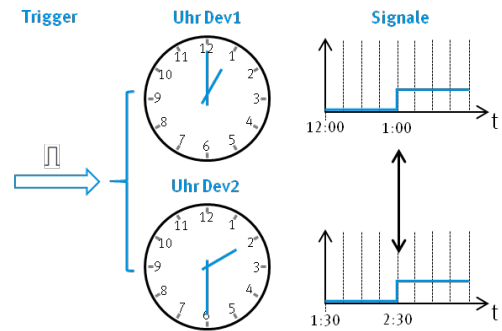


Abbildung 5: Zusammenführung zweier Zeitachsen durch Triggerung

Herstellerspezifische Mechanismen zur Synchronisation wie beispielsweise der *Real-Time System Integration Bus* (RTSI) von National Instruments zur Synchronisation von PC-Messkarten können Verwendung finden, bieten jedoch weniger Flexibilität im modularen Prüfsystemen.

Durch spezielle Netzwerkprotokolle wie dem *Precision Time Protocol* (PTP) und *Network Time Protocol* (NTP) können die hardwareeigenen Uhren mittels Ethernet-basierender Kommunikation synchronisiert werden. Dabei erfolgt zunächst ein Zeitabgleich aller Uhren auf eine Masteruhr. Das Auseinanderlaufen der Uhren durch Taktungenauigkeiten wird durch zyklisches Abgleichen verhindert.

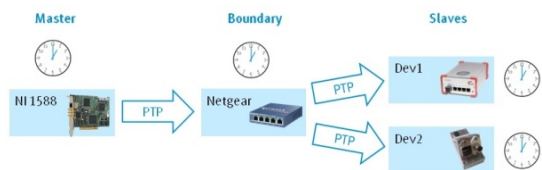


Abbildung 6: Synchronisation der Uhren per Precision Time Protocol

Auswertung

Durch die detaillierte Analyse aller Zeitfehler und Mechanismen zur Synchronisation war es möglich, Empfehlungen für zukünftige Testfälle auszusprechen, sowie Mechanismen prototypisch in die Hardware-/Softwarearchitektur zu integrieren.

Bildquellen:

- Abbildung 1–6: Eigene Abbildungen

Open Source Case Management

David Mändlen*, Harald Melcher, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

"It is not enough for code to work" [1]

Diese Bachelorarbeit wurde für und in Zusammenarbeit mit der Abteilung Digitale Forensik und Incident Response (DFIR) der SySS GmbH durchgeführt. Ziel war die Erstellung einer Software zur Vereinfachung der Fallverwaltung.

Bei der anfänglichen Bestandsaufnahme der bestehenden Lösungen ergab sich, dass die bisherige Umsetzung der Fallverwaltung ineffizient und fehleranfällig war. So wurden beispielsweise händisch laufende Nummern für Fälle vergeben, die Verwaltung der Beweismittel erfolgte auf Papierbasis und die Anlage des Analyse-Setups erforderte mehrere komplexe Arbeitsschritte, um sicherzustellen, dass die notwendigen virtuellen Maschinen erstellt wurden und genutzt werden konnten. Insbesondere die fehlende Datenbankunterstützung sowie die mühsam händisch anzulegenden Forschungsstrukturen erschwerten die schnelle, qualitativ hochwertige Durchführung von Ermittlungsarbeiten in erheblichem Maße.

Die hier erarbeitete Lösung hat den Anspruch, durch die eingebaute Datenbankunterstützung sowie durch die zusammengefassten Arbeitsschritte diese Probleme zu lösen.

Dazu wurden zunächst die bestehenden Prozesse der SySS GmbH analysiert und die vorhandenen Datenbestände geprüft. Aus diesen Informationen wurde in Zusammenarbeit mit der Fachabteilung ein Datenmodell in UML erstellt, das als Vorlage für die weiteren Arbeitsschritte diente. Aus der Ableitung dieses Datenmodells in einen Rohentwurf der Systemarchitektur wurden dann durch feingranulare Iterationen die wesentlichen Bestandteile und Anforderungen der geplanten Software nochmals präzisiert.

Die Umsetzung bereits vorhandener bash-Skripte in Python-Code erfolgte nach der Erstellung der Kernapplikation, da diese Skripte lauffähig waren und nur konvertiert wurden, um eine einheitliche Code-Basis zu haben. Der Vorteil dieser einheitlichen Code-Basis liegt vor allem in der vereinfachten Wartbarkeit sowie einer weitergehenden Flexibilität

der Anwendung.

Auf der Grundlage des Datenmodells sowie der Systemarchitektur wurde die Anwendung als Django-Applikation entwickelt, was vor allem in der Datenbankarchitektur eine erhebliche Erleichterung darstellte. Auch die Sicherheit der web-basierten Bestandteile der Anwendung wurde dadurch deutlich erhöht, durch das Django-Framework entfiel beispielsweise die Notwendigkeit für selbst geschriebene SQL-Statements sowie die Einbindung dieser in die Web-Applikation.

Durch diese Unterstützung konnte die vorhandene Zeit genutzt werden, um die Anwendung präzise auf die Bedürfnisse der Fachabteilung zuzuschneiden und wesentliche Bestandteile um wertvolle Features zu erweitern.

Die Kernbestandteile der neuen Architektur sind die Fallverwaltung einerseits, sowie die Beweisverwaltung andererseits, wobei die Fallverwaltung sich nochmals unterteilt in das Analyse-Setup, das die Anlage der notwendigen virtuellen Maschinen übernimmt, sowie den Datenbank-Anteil, wo die Ermittlungsergebnisse gespeichert werden.

Die beiden Kernbestandteile können unabhängig voneinander existieren und können somit auch getrennt weiterentwickelt werden, sollte sich hier eine Notwendigkeit ergeben. Zur Weiterentwicklung des Tools sind derzeit einige Features in der Diskussion, beispielsweise eine ReST-API zur automatisierten Auswertung der Fallbearbeitungsdauer und ähnlicher Kennzahlen.

Die gesamte Anwendung wurde nach aktuellem Stand der Softwareentwicklung erstellt und richtet sich an Current Best Practices aus, wie sie beispielsweise Robert C. Martin in „Clean Code: A Handbook of Agile Software Craftsmanship“ beschreibt.

Das Ergebnis ist eine plugin-basierte, flexible Lösung, die alle nötigen Fähigkeiten mitbringt, die Forensikfälle der SySS GmbH durch eine integrierte Bedienungsoberfläche einfach und bequem zu verwalten und als freie, quell-offene Lösung weltweit zur Nutzung und zur Verbesserung zur Verfügung steht.

*Diese Arbeit wurde durchgeführt bei der Firma SySS GmbH, Tübingen

Hierbei ist wichtig, dass die Definition von Freiheit sich auf die „Four Freedoms“ [2] der FSF stützt:

- The freedom to run the program as you wish, for any purpose (freedom 0)
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1)
- The freedom to redistribute copies so you can help your neighbor (freedom 2)
- The freedom to distribute copies of your modified versions to others (freedom 3)

Eine, zugegebenermaßen etwas flapsige, dennoch aber treffende, Umschreibung dieser vier Freiheiten stammt von Richard Stallman selbst: „To understand the concept, you should think of “free” as in “free speech,” not as in “free beer”.“

Da diese Lizenzierung für kommerziell eingesetzte Software nicht selbstverständlich ist, bedankt sich der Autor hiermit nochmals bei der SySS GmbH für deren vorbildliches Entgegenkommen in dieser Hinsicht.

Die neue Lösung wurde auf Basis von Django 1.9.6 in Python 3.4 erstellt und ist damit

offen für zukünftige Weiterentwicklungen.

Durch die verwendete Plugin-Architektur können Erweiterungen und Anpassungen schnell und einfach vorgenommen werden, ohne das Grundsystem zu beeinträchtigen oder es verändern zu müssen. Gleichzeitig ist aber die Verwendung im Rahmen der von der SySS GmbH verwendeten Prozesse bereits ohne Einschränkungen möglich. Da die Software bereits während der Erstellung mit der Apache 2.0-Lizenz ausgestattet und transparent und offen auf github entwickelt wurde, steht einer Nutzung sowie einer Verbesserung durch interessierte Kreise nichts im Wege. Interessierte Leser sind herzlich eingeladen, sich an einer gemeinschaftlichen Weiterentwicklung zu beteiligen. Der Autor hofft auf eine große interessierte Community, um dieses Tool weiterzuentwickeln.

Zusammengefasst lässt sich sagen, dass die neue Anwendung insbesondere in der SySS GmbH, aber auch in anderen Unternehmen, die im Bereich Digitale Forensik tätig sind, eine deutliche Vereinfachung und Beschleunigung der Arbeitsabläufe mit sich bringen wird.

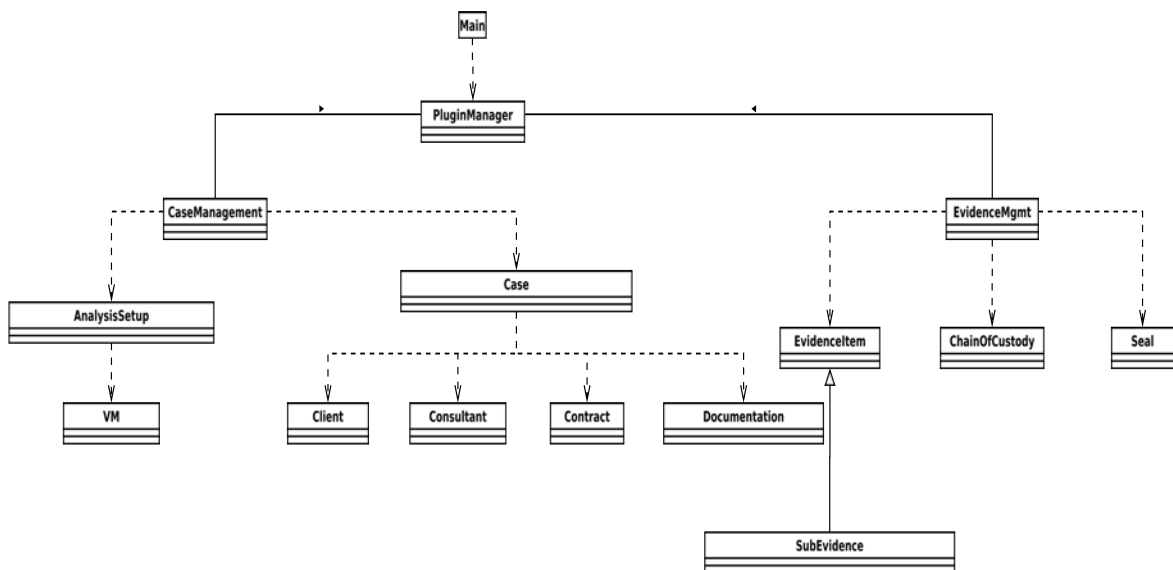


Abbildung 1: Architektur (Übersicht)

[1] Robert C. Martin, Clean Code:
A Handbook of Agile Software Craftsmanship
[2] www.gnu.org/philosophy/free-sw.en.html

Bildquellen:

- Abbildung 1: David Mändlen

Erstellung eines Closed-Loop Systems zur Durchführung automatisierter Tests von Software einer Getriebesteuerung

Oliver Melchior*, Hermann Kull, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Der Produktbereich Powertrain entwickelt kundenspezifische Systemlösungen für den gesamten Antriebsstrang. In diesem Fall geht es nicht nur um Verbrennungsmotoren, sondern um viele verschiedene neue Generationen der Motoren. Hierbei reicht es von der Elektrifizierung des Antriebsstranges bis hin zu Wasserstoffzellensystemen mit samt Getriebe [1].

Heutzutage werden zu den Motorentypen ständig neue Varianten von Getrieben entwickelt. Sie dienen dazu, dass die einkommende Eingangsdrehzahl durch den Motor in entweder eine höhere oder niedrigere Ausgangsdrehzahl umgewandelt wird. Notwendig ist es die Motordrehzahl in die Drehzahl der Räder, die für einen normalen Fahrbetrieb notwendig ist umzuformen. Die unterschiedlichen Übersetzungen (Gänge) werden durch verschiedene Kombinationen von unterschiedlichen Zahnrädern bestimmt.

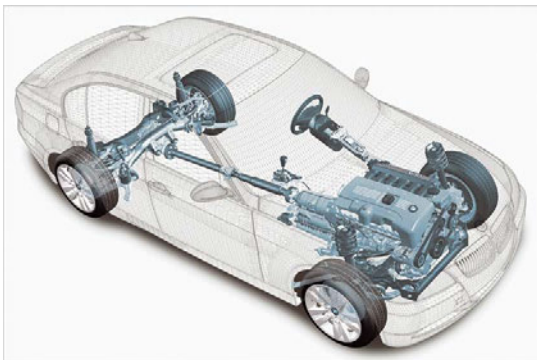


Abbildung 1: Antriebsstrang eines PKW (Heckantrieb)

Heutzutage gibt es zwei unterschiedliche Getriebearten mit jeweils firmeneigenen Abwandlungen und Namensgebungen. Diese sind:

1. Manuelle Schaltgetriebe

Dies ist die erste Generation von Getrieben, die heute immer noch in vielen Fahrzeugen verbaut ist. Der Fahrer muss dabei selber über die Kupplung die Verbindung vom Motor mit dem

Getriebe trennen, sodass die Wellen bei ungefähr gleicher Drehzahl frei drehen und über den Schalthebel neue Zahnräder mit der Ausgangswelle des Getriebes, die die Kraft an die Räder weitergibt, verbunden werden können.

2. Automatisches Schaltgetriebe

Dies ist die neueste Generation, bei der die Getriebesteuerung den optimalen Zeitpunkt zum Schalten berechnet. Daraufhin automatisch auskuppelt, den neuen Gang über hydraulische oder elektromechanische Aktoren einlegt und wieder einkuppelt. Dies sorgt für einen erhöhten Fahrkomfort und geringeren Kraftstoffverbrauch mit reduzierten Emissionen. Der Fahrer kann hierbei nur zwischen verschiedenen Fahrmodi umschalten, um die Schaltzeitpunkte zu verändern (Komfort, Sport, ECO, Manuell) [2].

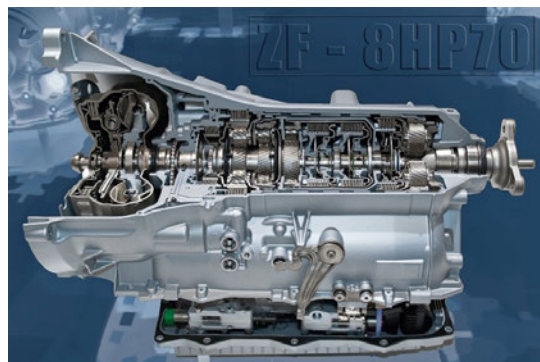


Abbildung 2: Automatisches Getriebe ZF-8HP70

Da einerseits die Softwareanforderungen der Getriebesteuerungen von automatischen Getrieben von Kunde zu Kunde abweichen und andererseits neue Varianten von Getrieben, wie das Doppelkupplungsgetriebe eingesetzt werden, muss diese jedes Mal ausreichend getestet werden. Die Besonderheit des Doppelkupplungsgetriebes ist, dass das Getriebe aus 2 Teilgetrieben besteht, wobei das eine aktiv ist und das andere inaktive den nächsten Gang schon voreinlegt, sodass nur die eine Kupplung geöffnet und die andere geschlossen werden muss. Die Vorteile hierbei sind

*Diese Arbeit wurde durchgeführt bei der Firma Bosch Engineering GmbH, Abstatt

geringere Schaltzeiten und besserer Fahrkomfort.

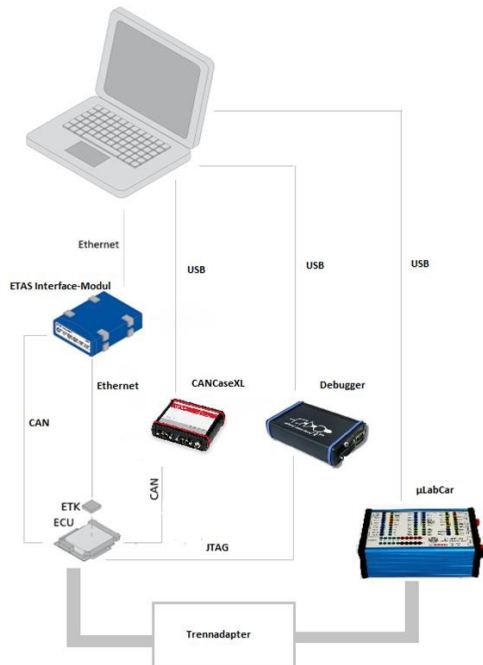


Abbildung 3: Aufbau der Testumgebung
Im Hinblick auf automatisiertes Testen, erstel-

le ich im Rahmen meiner Bachelorarbeit eine Testumgebung mit der es möglich ist verschiedene Hardware, wie ETAS-Module oder CANalyzer anzusprechen und untereinander kommunizieren zu lassen und Testszenarios für die Steuerung eines Doppelkupplungsgetriebes automatisch durchzuführen (Abbildung 3).

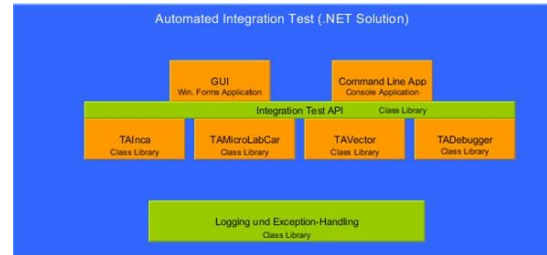


Abbildung 4: Architektur des Testtools

Um verschiedene Szenarios zu testen, haben wir über ein eigenes Tool (Abbildung 4) die Möglichkeit sämtliche Konfigurationen der Hardware und Arbeitsumgebungen zu laden und selbst in Python geschriebene Skripte auszuführen. Darüber können in INCA gesteuerte Parameter des Steuergeräts geändert, sämtliche Signale und Fehlerspeicher ausgelesen werden. Das Tool liefert daraufhin Ergebnisse und signalisiert an welchen Stellen die Tests nicht erfolgreich waren.

[1] Inside Bosch Homepage – BEG-PT Powertrain 2016

[2] Robert Bosch GmbH; Kraftfahrtechnisches Taschenbuch, Springer Vieweg, 2014, 28. Auflage

Bildquellen:

- Abbildung 1: www.auto-news.de
- Abbildung 2: <https://de.wikipedia.org/wiki/Fahrzeuggetriebe>
- Abbildung 3,4: Bosch Connect – [Tool] Automated Integration Test

Feature Driven Development

Denis Metzger*, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Einführung

Feature Driven Development (FDD) wurde von Jeff De Luca im Jahre 1997 erstmals in einem Projekt eingeführt und im Buch „Java Modeling in Color with UML“ von Peter Coad, Eric Lefebvre und Jeff De Luca im Jahre 1999 erstmals beschrieben. [3]

FDD ist ein domänengetriebener Prozess zur Entwicklung eines Softwaresystems, welcher agil ist. Dabei liegt ein besonderes Augenmerk auf einem Upfront High Level Model, welches als Leitfaden für die Entwicklung dient. Jeff De Luca unterstreicht dessen Bedeutung durch die Worte: „[...] there has to be some informational/analytical activity at the start to give us the knowledge to set a baseline that we can travel and report against [...]. FDD is the only agile method that gets this part right.“ [2]

FDD gilt als agile Methode, da die Entwicklungszyklen der sogenannten Features, welche jeweils einem zu realisierenden Arbeitspaket entsprechen, auf maximal zwei Wochen festgelegt sind. Durch die kurze Dauer eines Entwicklungszyklus ist eine schnelle Reaktion auf veränderte Anforderungen möglich.

FDD besitzt dabei ein eigenes Prozessmodell, welches die Entwicklung mit FDD vollständig beschreibt.

Der Begriff des Features

Bei der Entwicklung mit FDD steht der Begriff des Features im Mittelpunkt der Entwicklung. Ein Feature ist im Sinne von FDD eine überschaubare Entwicklungsleistung, welche im Auge des Kunden nützlich ist [2].

Die Aufteilung des Projekts in Features erfolgt dabei im Rahmen der Prozesse von FDD. Wichtig ist hierbei, dass bei der Planung des Systems bereits die maximale Implementationsdauer von zwei Wochen pro Feature beachtet wird. Ein Vorteil der geringen Implementationsdauer und der übersichtlichen Entwicklungspakete ist die sehr gute Schätzbarkeit des Zeitaufwandes [1]. Auf Basis der einzelnen Features lässt sich das Projekt leicht schätzen.

Rollen in FDD

FDD schreibt in seinem Prozessmodell verschiedene Rollen für die einzelnen Projektmitglieder vor. So ist der „Chefmodellierer“

für die fachliche Konvergenz der Teilsysteme und des Gesamtsystems verantwortlich. Ein „Chefprogrammierer“ steht an der Spitze eines Entwicklungsteams, welches in FDD „Feature-Team“ genannt wird. Dabei besteht ein Feature-Team aus dem Chefprogrammierer und den Verantwortlichen für die beteiligten Klassen eines Features (Klassenverantwortliche). Ein Feature-Team besteht maximal für die Dauer der Realisierung des entsprechenden Features, welche auf zwei Wochen begrenzt ist [1].

Prozesse FDD ist in fünf Prozesse unterteilt. Die ersten drei Prozesse dienen zur Erstellung des Upfront High Level Model und der Gliederung des Projekts in Features. Die letzten zwei Prozesse sind iterativ und spiegeln die Implementation eines jeden Features wieder. Im Folgenden werden diese Prozesse detailliert betrachtet (vgl. [1]).



Abbildung 1: Die fünf Prozesse des FDD

Entwickle Gesamtmodell

Die Entwicklung mit FDD beginnt mit der Erstellung eines High Level Upfront Model. Als Einstieg stellt ein Fachexperte den Problembereich des Systems vor. Teile des Modells werden zunächst in kleineren Gruppen entwickelt und anschließend zu einem Gesamtmodell zusammengeführt. Dabei werden Dokumente, welche den Problembereich näher beschreiben, mitberücksichtigt. Ebenso stellt ein Fachexperte im jeweiligen Team die Domäne der Features vor und schafft somit ein gutes Verständnis der sogenannten Problem Domain. Nach der Zusammenführung der Teilmodelle beginnt der erste Prozess auf Basis der gesammelten Ergebnisse von vorne, bis ein zufriedenstellendes Gesamtmodell entstanden ist. Durch die iterative und inkrementelle Erstellung des High Level Upfront Model entsteht ein detailliertes und klares Bild des Problembereichs und von dessen Anforderungen.

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

Erstelle eine Feature–Liste

Im Anschluss wird das Gesamtmodell in Features gegliedert und eine Feature–Liste mit allen Features, die implementiert werden sollen, erstellt. Dabei soll pro Feature die maximale Implementationsdauer von zwei Wochen eingehalten werden. Die Einhaltung der Implementationsdauer ist essentiell für den Erfolg eines Projekts mit FDD. Durch eine gute Schätzung des Zeitaufwands pro Feature ergeben sich genaue Termine für die Lieferzeiten und Fertigstellungsdaten einzelner Teilbereiche und somit auch des gesamten Projekts.

Plane in Features / je Feature

Vor der eigentlichen Implementierung der einzelnen Features gilt es jetzt noch, die Entwicklungsreihenfolge sowie die Zuständigkeiten für Klassen und Features zu klären. Dazu werden die Features in Geschäftsaktivitäten und diese wiederum in Fachgebiete eingeordnet. Anschließend wird die Entwicklungsreihenfolge der Fachgebiete sowie die der darin enthaltenen Features bestimmt. Durch die Festlegung der Entwicklungsreihenfolge und durch die maximale Dauer der Entwicklung pro Feature lassen sich die Fertigstellungsdaten besser schätzen, was zu einer besseren Planbarkeit des Projekts beiträgt. Geschäftsaktivitäten werden einem Chefprogrammierer zugewiesen, der für die Implementierung der darin enthaltenen Features verantwortlich ist. Diese Klassenverantwortlichen sind Entwickler oder Chefprogrammierer. Durch die Zuweisung der Zuständigkeiten ergeben sich klare Verantwortlichkeiten für entwickelten Code.

Entwerfe je Feature

Nach der Festlegung der Entwicklungsreihenfolge wird mit der Implementierung der einzelnen Features begonnen. Dazu wählt der Chefprogrammierer ein Feature aus einer Geschäftsaktivität aus und stellt das Feature–Team zusammen. Das Feature–Team besteht dabei aus den Entwicklern, die für die beteiligten Klassen verantwortlich sind. Im Anschluss werden Sequenzdiagramme des Features angelegt und die Methodenprologe, welche Rückgabewerte, Übergabeparameter und mögliche Ausnahmen umfassen, der erforderlichen Methoden von den Klassenverantwortlichen festgelegt. Erst nach erfolgreicher Inspektion durch das Feature–Team kann mit der Implementierung begonnen werden.

Implementiere je Feature

Die Implementierung des jeweiligen Features beinhaltet nicht nur das Programmieren, sondern auch die Tests der implementierten Methoden. Ebenso können für das gesamte Feature noch Tests angelegt werden. Dabei wird das Feature erst nach erfolgreichem Bestehen der Tests und einer Codeinspektion in den aktuellen Stand des Projekts eingepflegt. Dabei betont Palmer [1] die Bedeutung des Einpflegens. Ein Feature ist erst abgeschlossen, wenn alle notwendigen Dokumente bezüglich des Features auf einem finalen Stand sind. So ist vermieden, dass nicht oder unvollständig dokumentierte Features als fertiggestellt gelten. Bei allen Teilprozessen von FDD werden Notizen zu den Dokumenten oder den vorgeschlagenen Varianten geschrieben und in einem geeigneten Tool abgelegt. Dadurch ist gewährleistet, dass Alternativen zu dem ursprünglichen Design nicht verloren gehen. Ebenso wird der Status der einzelnen Features festgehalten. Dadurch ist es jederzeit möglich, den aktuellen Entwicklungsstand und den verbliebenen Restaufwand abzurufen.

Domain Object Modeling

Als Domain Object Modeling wird eine Modellierung des Systems im Problembereich bezeichnet. Durch die starke Einbindung der Fachexperten in der Planung des Projekts ist das Prinzip des Domain Object Modeling in FDD sehr stark vertreten. Ein Vorteil des Domain Object Modeling ist das klare und einheitliche Verständnis des abzudeckenden Problembereichs. Durch das Richten der Aufmerksamkeit auf den Problembereich wird zudem verhindert, dass falsche Annahmen gemacht werden und zusätzliche Arbeit durch Missverständnisse wird somit vermieden. [1]

Bewertung

Durch das eigene Prozessmodell wirkt FDD trotz der kurzen Lieferzyklen von zwei Wochen etwas schwergewichtig. Andererseits ist FDD jedoch sehr gut strukturiert und durchdacht. Die Messbarkeit des aktuellen Entwicklungsstandes ist durch die Aufteilung in Features sehr einfach und genau. Ebenso ist FDD gut auf große Projekte anwendbar, da die Aufteilung in Features nahezu beliebige Projektgrößen erlaubt. Je größer das Projekt, desto größer die Menge der Features.

-
- [1] Palmer, Stephen R. und Felsing, John M. 2002. A Practical Guide to Feature–Driven–Development. New Jersey : Prentice–Hall, Inc., 2002. ISBN 0–13–067615–2.
 [2] De Luca, Jeff. 2007. Jeff De Luca on Feature Driven Design. it–agile. s.l. : akquinet, März 2007.
 [3] Peter Coad, Eric LefeBvre, Jeff de Luca. 1999. Java Modeling in Color with UML. New Jersey : Prentice–Hall, Inc., 1999. ISBN 0–13–011510–X.

Bildquellen:

- Abbildung 1: vgl. Wolf Henning. 2007. it–agile.de.
http://www.it-agile.de/fileadmin/videos/FDD-Vorstellung_XP_Days2007.pdf Zugriff am 25.02.2016

Konzepterstellung und Realisierung einer konfigurierbaren SENT-Schnittstelle nach SAE J2716 mit optimalem Formfaktor unter Anwendung des Verfahrens Hardware-Software-Codesign

Chris Manuel Mönch*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Durch *Smart Sensors* werden in der Automobilbranche neue Anforderungen an die Datenübertragung von Sensoren zu Steuergeräten gestellt. Die Verfahren der analogen und PWM-basierten Übertragungen können diese nicht erfüllen und die vorhandenen standardisierten digitalen Schnittstellen sind zu kostenintensiv, erfordern einen hohen Realisierungsaufwand oder sind nur für kurze Reichweiten geeignet. Zur Ablösung der analogen Schnittstellen werden daher neue offene und standardisierte Schnittstellen mit hoher Übertragungsrates benötigt, die kostengünstige und robuste Alternativen zu *CAN* und *LIN* darstellen, ohne die Komplexität des Systems zu erhöhen. Dabei müssen folgende Anforderungen der *Smart Sensors* durch die digitalen Schnittstellen erfüllt werden:

- Konfiguration über die Schnittstelle
- Steigerung der Datenintegrität
- Erweiterte Diagnosefähigkeit
- Clusterbildung, Sensor- und Datenfusion

Eine digitale Schnittstelle, die diese Anforderungen erfüllt, ist das durch die *Society of Automotive Engineers (SAE)* standardisierte Protokoll *Single Edge Nibble Transmission (SENT)* mit der Bezeichnung *SAE J2716*. *SENT* ist eine

unidirektional, asynchrone und digitale Punkt-zu-Punkt-Schnittstelle für die Verbindung von Sensoren und *ECUs*, die speziell für die Anforderungen der Automobilindustrie entwickelt wurde und bei Positions-, Winkel-, Druck- und Temperatursensoren Verwendung findet. Die Drei-Draht-Schnittstelle besteht aus einer Versorgungsleitung, einer Masseleitung und der Signalleitung [2].

Die Codierung und Decodierung der als 4-Bit-Wert übertragenen Information, auch *Nibble* genannt, erfolgt zwischen zwei *Falling Edges (Pause-Pulse-Modulation)*. Die variable Zeitdifferenz zwischen zwei *Falling Edges* wird als Vielfaches der sogenannten *Ticktimes (TT)* angegeben, welche die Zeitbasis einer Übertragung darstellt. So entsprechen $12TT$ dem Wert $0x0$ und $27TT$ dem Wert $0xF$. Übliche Werte eines *TT* sind $3\mu s$ bis zu $90\mu s$ mit einer Toleranz von bis zu $\pm 20\%$. Für das Generieren einer *Falling Edge* werden logische Nullen mit einer Dauer von $5TT$ gesendet, gefolgt von $7TT$ bis $22TT$ Anzahl von logischen Einsen. Innerhalb dieser *TT* muss garantiert sein, dass das Signal den High- bzw. Low-Pegel erreicht, um die *Falling Edge* selektieren bzw. generieren zu können. Aus der *Raising-* und *Falling-Time* des Systemes ergeben sich die maximal mögliche realisierbaren *TT* des Systemes.

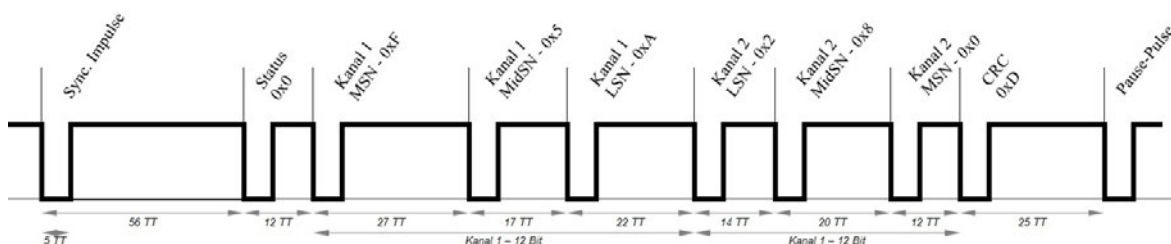


Abbildung 1: FDC Standard Frame

Die primären Daten werden über einen *SENT*-Frame (*Fast Data Channel, FDC*) und die sekundären Daten werden über die sogenannte *Slow Data Message (SDM)* übertragen. Die Daten des *SDM* werden aus *FDC*-Frames zusammengesetzt. Ein *FDC*-Frame (siehe Abb. 1) besteht aus einem Synchronisationsimpuls,

der immer die konstante Länge von $56TT$ hat. Aus diesem Impuls kann die verwendete *TT* der Übertragung für die Decodierung bestimmt werden. Zu Beginn wird ein Status-Nibble gesendet, das *SDM*- und applikationsspezifische Informationen beinhaltet. Anschließend folgen bis zu 6 Daten-Nibble, die sich auf zwei Kanä-

*Diese Arbeit wurde durchgeführt bei der Firma ARRK | P+Z, Stuttgart

le aufteilen, gefolgt von einer CRC-Prüfsumme mit der Länge 4 Bit. Für die Sicherstellung einer konstanten Übertragungsrates kann optional noch ein *Pause-Pulse* eingefügt werden.

Die Übertragung im *FDC*-Frame bietet spezielle Modifikationsmöglichkeiten für eine schnellere oder sichere Übertragung. Der *Fast Channel High Speed* beschränkt die Anzahl der Kanäle auf einen und reduziert die Größe eines Datenelementes von 4 Bit auf 3 Bit. Dadurch verringert sich die maximale Übertragungszeit und die Zeitvariation einer Nachricht wird begrenzt. Der Modus *12 Bit Single Secure Message* benötigt drei Daten-Nibbles zur Darstellung eines *8 Bit Rolling Counter* und für die Spiegelung des *Most Significant Nibble* des *Counters*. Durch diese Modifikation können die Anforderungen an die Übertragungen von Signalen erfüllt werden die durch die ISO26262 gefordert werden.

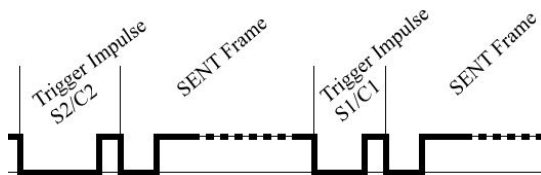


Abbildung 2: SPC Trigger

Die Daten des *Slow Data Message (SDM)* werden aus den beiden höchstwertigen Bits des Status Nibbles mehrerer *FDC*-Frames zusammengesetzt. Damit werden Daten übertragen, die über eine geringe Änderungsrate verfügen oder konstant sind. Je nach ausgewähltem Modus besteht eine *SDM*-Nachricht aus 16 oder 18 *FDC*-Frames. Der Standard Modus besteht aus 32 Bit, beinhaltet eine ID der Länge 4 Bit, Daten der Länge 8 Bit und eine *CRC* der Länge 4 Bit. Das *MSB* des Status-Nibble markiert mit einer 1 den Beginn einer neuen *SDM*. Der *Extended Modus* (siehe Abb. 3) besteht aus 36 Bit, beinhaltet eine *CRC* der Länge 6 Bit, Daten der Länge 12 Bit bzw. 16 Bit und ID mit der Länge von 8 Bit bzw. 4 Bit [1].

Tabelle 1: Vergleich Kerndaten von SENT mit SPC

	SENT	SPC
Topologie	P2P	Bus/P2P
Zugriffsverfahren	Broadcast	Master-Slave
Datenübertragung	Uni-Direktional	Bi-Direktional
Datenübertragung	Asynchron	Synchron
Vcc	5V	3.3 oder 5V

Die Erweiterung *Short PWM Code (SPC)* erhöht die Funktionalität der *SENT*-Schnittstelle (siehe Tab.1). Primär wird eine bidirektionale Verbindung zwischen Sensor und *ECU* ermöglicht. Zusätzlich zur Punkt-zu-Punkt-Verbindung unterstützt *SPC* auch eine Verwendung des Systemes als Bus mit mehreren

Sensoren an eine Interface. Die *ECU* übernimmt die Rolle des Masters im System und leitet die Kommunikation der Slaves über einen Trigger-Impuls (siehe Abb. 2) ein. Typische Werte für das *SPC*-Format sind nachfolgend in Tabelle 2. dargestellt. Der jeweils angesprochene Slave beginnt dann mit der Übertragung des üblichen *SENT*-Frames. Die Bedeutung des Triggersignals wird über die Zeitdauer des Low-Impulses codiert. Die Erweiterung *SPC* erlaubt folgenden Betriebsarten [2]:

- Synchroner Modus, Polling eines Slaves über ein kurzes Signal
- Command/Bus Modus, Senden kurzer Befehle oder Triggern eines Slaves
- Long Command Modus, Senden langer Befehle zum Sensor

Im Rahmen der vorliegenden Bachelorarbeit wurde eine Schnittstellenkarte für ein *Automotive Embedded System* zur Erweiterung einer bestehenden Kommunikationsplattform entwickelt um die *SENT*-Schnittstelle und dessen Erweiterung, dem *Short PWM Code*, sowie den Modifikationen des *Fast Data Channels* und des *Slow Data Message*. Dazu wurde eine Platine entworfen, die sowohl als Sender wie auch als Empfänger einsetzbar ist.

Tabelle 2: Typische Zeiten des SPC

<i>TT</i> [μ s]	1.5	90
Betriebsart	Low-Time	
	Min.	Max.
Sync. [<i>TT</i>]	1.5	4
ID0/C0[<i>TT</i>]	9	12
ID1/C1[<i>TT</i>]	19	23
ID2/C2[<i>TT</i>]	35.5	40.5
ID3/C3[<i>TT</i>]	61.5	67.5

Für die Realisierung wurde das *V-Modell* verwendet. In einer Anforderungsanalyse wurden Lasten und Pflichten an die Hardware und Software des Systems definiert und beschrieben. Für eine parallele Umsetzung der Hardware und Implementierung der Software wurde auf das *Hardware/Software Codesign* Verfahren zurückgegriffen. Mit diesem Verfahren wird die Aufteilung der Funktionen in Hardware und Software festgelegt.

Für die Demonstration der Funktionalität, zur Analyse und Auswertung des Systems wurde eine Sensorkarte mit *SENT*- und *SPC*-Sensoren aufgebaut.

Die Hardware unterstützt, aufgrund der verfügbaren Pins und des begrenzten Platzes, drei konfigurierbare *SENT/SPC* Interfaces. Die Implementierung der Software wurde mittels der Programmiersprache *C* in der *Schichtenarchitektur* umgesetzt.

Die zu übertragenden *FDC*- und *SDM*-Daten werden im Slave-Modus, der einen Sensor simuliert, unbearbeitet über *CAN* versendet.

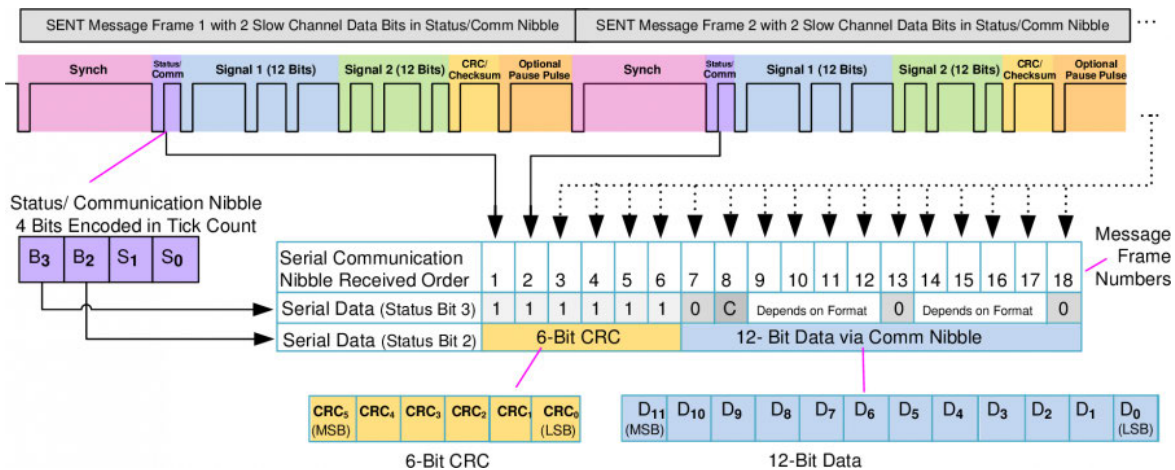


Abbildung 3: Aufbau einer SDM-Nachricht im Extended Format

Der Treiber wandelt diese in das jeweils konfigurierte Format um. Die Bits im Status-Nibble werden automatisch für die *SDM*- und applikationsspezifischen Daten gesetzt. Der Slave sendet Informationen über *CAN*, sobald dieser ein Trigger-Signal empfängt oder eine *SDM*-Nachricht komplett versendet wurde. Die CRC-Prüfsummen werden nach dem *SENT*-Standard *SAE J2726* oder nach der Konfiguration berechnet. Bei der Übertragung eines *Rolling Counter* wird dieser automatisch inkrementiert.

Konfiguriert als Master werden die erhaltenen *SENT*-Frames decodiert und je nach Modifikation in die jeweiligen Daten gewandelt. Die CRC-Prüfsummen werden für *FDC*-Frame und *SDM*-Nachricht überprüft. Falls es sich um ein *FDC*-Frame des Types *12 Bit Secure Message* handelt, wird der *Rolling Counter* und das invertierte Nibble überprüft. Per *CAN* werden die empfangenen *FDC*-Daten mit den jeweiligen Flags zur Daten-Integrität und sobald eine

vollständige *SDM*-Nachricht vorliegt die *SDM*-Daten versendet. Die am Bus befindlichen Slaves dürfen eine variable *TT* besitzen, müssen jedoch mit einer Toleranz von 20% eindeutig über den ID-Trigger unterscheidbar sein.

Die erstellte Platine war in Betrieb zu nehmen und insbesondere hinsichtlich des Umschaltens der Versorgungsspannung, des Master und Slave Verhaltens sowie der Generierung und dem Decodieren von Frames zu testen.

Die erreichbaren *TT*-Zeiten, die *Latenzen* und die *Jitterzeiten* des Systems sind mit Hilfe der Sensorkarte zu analysieren.

Mit dem realisierten *SENT*-Modul, das als *SENT/SPC* Master/Slave verwendet werden kann, werden *Falling Times* von $1,95\mu\text{s}$ und *Rising Times* von $6,54\mu\text{s}$ sichergestellt. Damit lassen sich im dem System *TT* von bis zu $1\mu\text{s}$ realisieren. Die *Latenzen* des *SENT*-Moduls betragen maximal $0,8\mu\text{s}$, gemessen vom Sensor bis zum Pin der *MCU*.

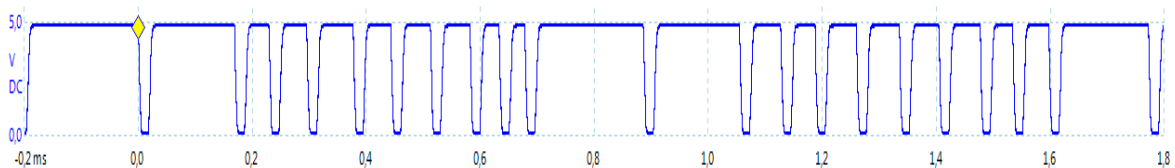


Abbildung 4: SENT-Signal des Sensors MLX90366

In der Abbildung 4 sind zwei aufeinanderfolgende *SENT*-Frames des Sensors *MLX90366* mit 6 Datennibbles, die über das *SENT*-Modul mittels Oszilloskop ausgewertet wurden, dar-

gestellt. Die *TT* beträgt $3\mu\text{s}$. Der *Pause-Pulse* wird verwendet, um eine Übertragungsrate von $36\frac{\text{kBit}}{\text{s}}$ sicherzustellen.

[1] SAE International; SENT – Single Edge Nibble Transmission for Automotive Applications, 27.01.2010

[2] Bearenaud, Laurent; Short PWM Code: A Step towards Smarter Automotive Sensors, in: Advanced Microsystems for Automotive Applications 2009: Smart Systems for Safety, Sustainability and Comfort, hrsg. v. Meyer, Gereon; Valldorf, Jürgen, Springer Berlin Heidelberg, 2009, S. 383 – 395, 978-3-642-00745-3

Bildquellen:

- Abbildung 1,2,4: Eigene Darstellung
- Abbildung 3: Tim White, Zentrum Mikroelektronik Dresden

Konzeption und Implementierung einer App zur Interaktion mit der Produktlebenslaufakte in einer Shopfloor-Umgebung

Dominik Müller*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Sämtliche wichtigen Produktionsvorgänge, sowie verbauten Teile in den Powertrain-Werken der Daimler AG werden in der Produktlebenslaufakte (PLA) mit allen Abhängigkeiten und Verbau-Stufen dokumentiert. Je nach resultierendem Aggregat und dessen Dokumentationspflicht, wie Motor, Getriebe oder Achse, werden die Daten der verbauten Teile bis zu 15 Jahre lang archiviert. Alle qualitätsrelevanten Daten über produzierte Teile können also der Produktlebenslaufakte entnommen werden, womit sie einen produktionsentscheidenden Faktor darstellt. Da die PLA derzeit nur über Desktop-Geräte optimal zu erreichen ist, liegt im mobilen Bereich noch einiges an Potenzial.



Abbildung 1: Designkonzept für die App

Gegenstand der hier vorgestellten Arbeit ist die Konzeption und Entwicklung einer Smartphone-optimierten Applikation für eine bereits bestehende Desktop-Anwendung der Produktlebenslaufakte PLA. Besonderes Augenmerk wird auf die Usability der App gelegt, da selbige in einer Shopfloor-Umgebung zur Anwendung kommt. Aus diesem Grund werden einige Bedienkonzepte verglichen und auf Sinnhaftigkeit überprüft. Wichtige Gründe für die Entwicklung sind Sichtprüfungen der Ag-

gregate im Werk, welche sofort in der App vermerkt werden können und somit zu kürzeren Laufwegen verholfen wird, da nicht an den stationären Arbeitsplatz zurückgekehrt werden muss. Dabei spielt Zeit und die gesteigerte Produktivität sowie eine höhere Effektivität eine entscheidende Rolle.

Zur Konzeption der App werden zunächst die Stakeholder erfasst, Risiken und Probleme dargestellt und Use Cases erstellt. Daraus wird abgeleitet, wie die Entwicklung vonstattengehen soll. Um die App zu entwickeln muss zwischen einer nativen, hybriden oder Web App verglichen werden, wobei Vor- und Nachteile der einzelnen Methoden herausgearbeitet werden. Dabei stellt sich heraus, dass die Entwicklung einer Web App, aufgrund der Abhängigkeiten von Software Development Kits der jeweiligen Hersteller bei der nativen und hybriden Entwicklung, in vielen Fällen Sinn macht.

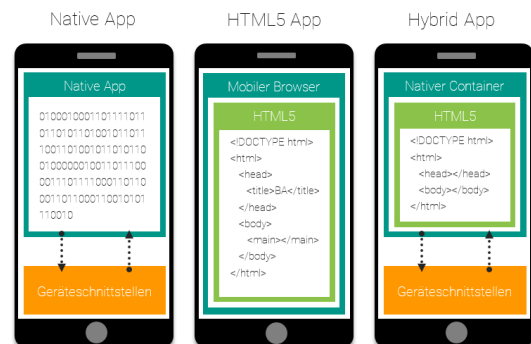


Abbildung 2: Vergleich zwischen Nativer App, HTML5App und HybridApp

Je nach Methode wird ein passendes und voll funktionsfähiges Tool verwendet, welches im Vorhinein an ähnlichen Tools gemessen wurde. Auftretende Probleme sind hierbei die Systemkompatibilität, sowie die späteren App-Berechtigungen auf den Endgeräten und die Barriere der verschiedenen Programmiersprachen, welche bei der nativen Programmierung zur Anwendung kommen. Zusätzlich muss beachtet werden, wie Übertragungsfehler und Netzabbrüche behandelt werden sollen. Verbindungstechnisch beachtenswert ist hierbei

*Diese Arbeit wurde durchgeführt bei der Firma Daimler AG, Esslingen-Mettingen

ebenfalls die Synchronisierung zwischen verschiedenen Geräten, wobei relevante Datenänderungen auf einem Gerät in Echtzeit auch auf anderen Geräten angezeigt werden müssen.

Um die App tauglich für den Einsatz im Shopfloor zu machen, wird sie nicht nur funktionstechnisch, sondern auch mit starkem Fokus auf die Usability entwickelt. Bei der Usability wird ein besonderes Augenmerk auf Steve Krugs Regeln der Usability [1], insbesondere auf die erste Regel „Don't make me think“ [2] gelegt. Das Ziel muss sein, dass die App intuitiv, aktiv und gerne genutzt wird. Um die Usability zu maximieren, müssen sämtliche Zustände mit Mockups (Abbildung 2) abgedeckt werden, aus denen für Testzwecke ein sogenannter Click-Dummy erstellt wird. Der Click-Dummy wird mit den Werkern getestet und aus den gewonnenen Erkenntnissen wird das finale Layout und Design erarbeitet.

Um der App die nötige Relevanz im Unternehmen zu geben, soll nach Fertigstellung erfasst werden, inwiefern eine Zeitersparnis eingetreten und sich die Effektivität erhöht hat. Außerdem könnte es möglich sein, die App an externe Mitarbeiter weiterzureichen, sodass diese für die Nacharbeitung von fremdvergebenen Aggregaten und Teilen ebenfalls die App benutzen können. Dadurch ist es möglich, einiges an Kommunikationsverkehr und Komplikationen, die auf dem herkömmlichen Weg vorhanden sind, einzusparen.



Abbildung 3: Frühes Mockup der App (Einzelfreigabe)

[1] Krug, S. (2006). Don't Make Me Think! A Common Sense Approach to Web Usability, Second Edition. Berkeley, CA: New Riders

[2] Krug, S. (2006). Chapter 1: Don't make me think! In S. Krug, Don't Make Me Think! A Common Sense Approach to Web Usability, Second Edition (S. 11–19). Berkeley, CA: New Riders

Bildquellen:

- Abbildung 1,3: Eigene Darstellung
- Abbildung 2: Eigene Darstellung, inspiriert durch Worklight, 2011a

Funktionserweiterung eines Systems für das Condition-Monitoring unter Berücksichtigung von Aspekten der Network Security

Matthias Noll*, Reinhard Keller, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Die Digitalisierung aller Dinge unter dem Schlagwort *internet of things* und die damit verbundene Vernetzung schreitet immer weiter voran. Damit nimmt auch die Kommunikation zwischen den einzelnen netzwerkfähigen Geräten immer mehr zu. Was im privaten Gebrauch schon gang und gäbe ist, hält jetzt auch unter dem Schlagwort „Industrie 4.0“ Einzug in die Industrie. Die Kommunikation von Maschinen mit einer Überwachungseinheit für die vorbeugende Wartung wird als Condition-Monitoring bezeichnet. Damit kann die betroffene Anlage melden, falls beispielsweise ein Verschleißteil einen hohen Abnutzungsgrad aufweisen sollte oder ein anderes Problem wie etwa eine Überhitzung auftreten sollten.

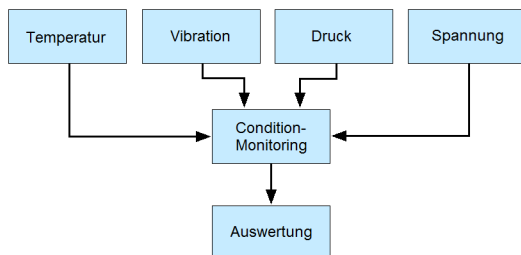


Abbildung 1: Schema des Informationsflusses des Condition-Monitoring

Außerdem kann durch eine zentrale Auswertung der Daten beispielsweise ein Zustandsverlauf für jede zu überwachende Maschine anlegt werden. Die Auswertung kann dann zu einem späteren Zeitpunkt oder auch im Fehlerfall verwendet werden, um Fehlerbilder besser zu erkennen und um die sich abzeichnenden Ausfälle im Vorfeld zu verhindern. So ist es möglich, kostenintensive Maschinenstillstandszeiten zu minimieren. Aus diesem Grund ist die visuelle Darstellung der erfassten Daten eines der größeren Aufgabengebiete der vorliegenden Abschlussarbeit, was als Ziel die Darstellung der Messwerte der für Condition-Monitoring eingesetzten Sensoren über der Zeit hat. Dies ist sowohl mit einem eigens dafür entwickelten Tool als auch mit einem Überwachungsprogramm möglich, dessen Funktionen um eine automatische Daten-

auswertung und das Erzeugen und Speichern von Systemzuständen erweitert wurden.

Um den Mehrwert nutzen zu können, muss zu Beginn die passende Infrastruktur in Form von geeigneten Kommunikationswegen geschaffen werden. Die Verwendung einer kabelgebundenen Übertragung ist meist aufwendig, da die Leitungen in Fertigungshallen gut geschützt verlegt werden müssen, insbesondere wegen der mechanische Belastung, aber auch hinsichtlich der elektromagnetischen Verträglichkeit. In einigen kabelgebundenen Systemen ist daher sogar eine Ausfallerkennung implementiert. Neben der leitungsgebundenen Kommunikation gibt es weitere Möglichkeiten.

Eine der Optionen ist die drahtlose Kommunikation, deren Vorteil die äußerst flexiblen Einsatzmöglichkeiten sind, sei es an schwer erreichbaren Stellen oder in mobilen Geräten und Maschinen. Allerdings ist unbedingt zu beachten, dass die Signalausbreitung leicht durch andere Objekte, wie beispielsweise eine Mauer, einen Menschen oder allgemein durch Gegenstände beeinflusst werden kann. Zusätzlich ist die drahtlose Kommunikation empfindlich gegenüber elektromagnetischer Einstrahlung, ein potentieller Schwachpunkt des Systems ist auch der vereinfachte physische Netzzugriff.

Wie aber kann ein Netzwerk effizient und angemessen geschützt werden? Vorab ist der Begriff der Sicherheit genauer zu umreißen: Im Deutschen beschreibt das Wort auch im Kontext von Software sowohl den Schutz vor mutwilliger Manipulation (*Security*) als auch das Verhindern eines Schadens durch Fehlfunktionen (*Safety*). Relevant hier sind die Aspekte der *Network Security*.

Eine der Herausforderungen in der vorliegenden Abschlussarbeit war es, eine geeignete Sicherheitsmaßnahme für die Übertragung der Daten zu finden. Basis hierfür ist ein fundierter Überblick über die grundlegenden Sicherheitskonzepte, denn die *Security* umfasst viele Aspekte wie beispielsweise die Verschlüsselung und die Authentifizierung. Beide sollen hier kurz umrissen werden. Die Verschlüsselung kann weiter in symmetrische und asymmetrische Verfahren differenziert werden.

*Diese Arbeit wurde durchgeführt bei der Firma Festo AG & Co. KG, Esslingen-Berkheim

Die symmetrischen Verfahren sehen vor, die Nachricht mit einem Schlüssel zu codieren und mit demselben Schlüssel auch wieder zu decodieren. Ein Beispiel hierfür ist die Cäsar-Chiffre, bei der die Buchstaben des Textes um eine bestimmte Anzahl an Stellen im Alphabet verrückt werden.

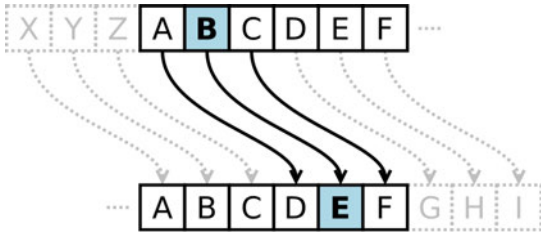


Abbildung 2: Cäsar Chiffre

Durch diese Methode ist der Inhalt der Nachricht nicht ohne Decodierung ersichtlich und somit geschützt. Symmetrische Methoden werden für die schnelle Verschlüsselung größerer Datenmengen genutzt. Vorteilhaft ist, dass selbst kurze Schlüssel schon einen relativ hohen Grad an Sicherheit bieten. Das Sicherheitsrisiko liegt hier vor allem in der Übergabe des Schlüssels.

Diese Sicherheitslücke ist bei den asymmetrischen Verschlüsselungsverfahren geschlossen, da hier jeder Nutzer über ein Schlüsselpaar, bestehend aus einem privaten und einem öffentlichen Schlüssel, verfügt. Soll eine Nachricht verschlüsselt werden, wird diese mit dem öffentlichen Schlüssel des Empfängers codiert. Nur der Empfänger selbst kann die Daten mit seinem eigenen privaten Schlüssel wieder decodieren. Ein Nachteil der Methoden ist, dass das Schlüsselpaar durch einen Algorithmus miteinander verbunden sein muss. Die Schlüsselerzeugung ist hier kompliziert, da der private aus dem öffentlichen Schlüssel nicht oder möglichst schwer zu rekonstruieren sein darf. Diese Problematik erfordert Schlüssel, die bei gleicher Sicherheitseinstufung ein Vielfaches länger sein müssen als vergleichbare symmetrische Schlüssel. Dies macht die eigentliche Datenverschlüsselung über die asymmetrischen Methoden sehr

zeitaufwendig.

Eine weitere Methode ist die hybride Verschlüsselung, die die Vorzüge beider vorher genannten Verfahren vereint. Hier wird nur der Schlüssel der symmetrischen Methode mit einem asymmetrischen Verfahren verschlüsselt und anschließend versendet.

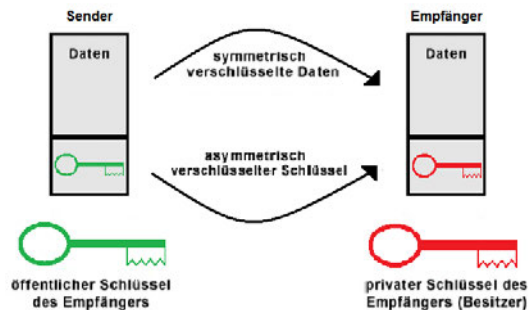


Abbildung 3: Schematische Darstellung der Hybriden Verschlüsselung

Zur Authentifizierung und zum Sicherstellen der Integrität von Datenpaketen werden digitale Signaturen verwendet. Beim sogenannten Signieren wird von der zu sendenden Nachricht ein Referenzwert berechnet. Dieser Wert wird mit dem privaten Schlüssel des Senders verschlüsselt und dem eigentlichen Dokument angehängt. Der Empfänger kann ebenfalls den Referenzwert der Nachricht berechnen und den gesendeten Referenzwert mit dem öffentlichen Schlüssel des Senders decodieren. Wenn beide Referenzsummen übereinstimmen, dann ist sowohl gesichert, dass die Nachricht nicht nachträglich verändert worden ist, als auch, dass sie vom Sender des Schlüsselpaares stammt[1].

Die Evaluierung der verschiedenen Optionen der Codierung war die Grundlage für die Entscheidungsfindung über die zu verwendende Methode oder zu verwendende Kombination von Methoden der Verschlüsselung. Ziel hierbei war eine bestmögliche Abdeckung der Kriterien Rechenzeit und Sicherheitsstufe durch die verwendete Lösung.

[1] it-Wissen (Hrsg): Digitale Signatur, www.itwissen.info/definition/lexikon/Digitale-Signatur-DSig-digital-signature.html [Zugriff am 11.05.2016]

Bildquellen:

- Abbildung 1,3: Matthias Noll
- Abbildung 2: Wikipedia (Hrsg): Caesar-Verschlüsselung, www.wikipedia.org [Zugriff am 11.05.2016]

Automatisierte Validierung kennfeldbasierter Applikationsdaten

Philipp Rauscher*, Walter Lindermeir, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Der Fahrkomfort eines Fahrzeugs ist in hohem Maße abhängig von einem ruckfreien Momentenaufbau. Gerade deshalb stellen der positive und der negative Lastwechsel im Hinblick auf das Komfortempfinden eine besonders wichtige Situation dar, da hier das Potential für lastwechselinitiierte komfortmindernde Antriebsstrangschwingungen besteht. Vereinfacht dargestellt lässt sich der gesamte Antriebsstrang als ein schwingendes, über Dämpfer mit dem Fahrzeug verbundenes Feder-Massesystem beschreiben. Befindet sich ein Fahrzeug im Schubbetrieb, wird der Antriebsstrang aufgespannt und das Fahrzeug mitgeschleppt. Auf die Komponenten

des Antriebsstrangs wirkt ein vom Getriebeausgang herrührendes Moment. Das Betätigen des Fahrpedals hat zur Folge, dass die auf die spielbehafteten Teile wirkenden Kräfte nun entgegengesetzt angreifen. Daraus resultiert, dass sich die verbindenden Kräfte abbauen und sich die Komponenten im Momentennulldurchgang voneinander lösen. Zu diesem Zeitpunkt sind die Bauteile ohne Kraftschluss. Dann schlagen die spielbehafteten Teile an den gegenüberliegenden Komponenten an und verursachen so spürbare Schläge, die Schwingungen anregen. Der Übergang von Schubbetrieb in den Zugbetrieb ist in Abbildung 1 schematisch dargestellt [1][2].

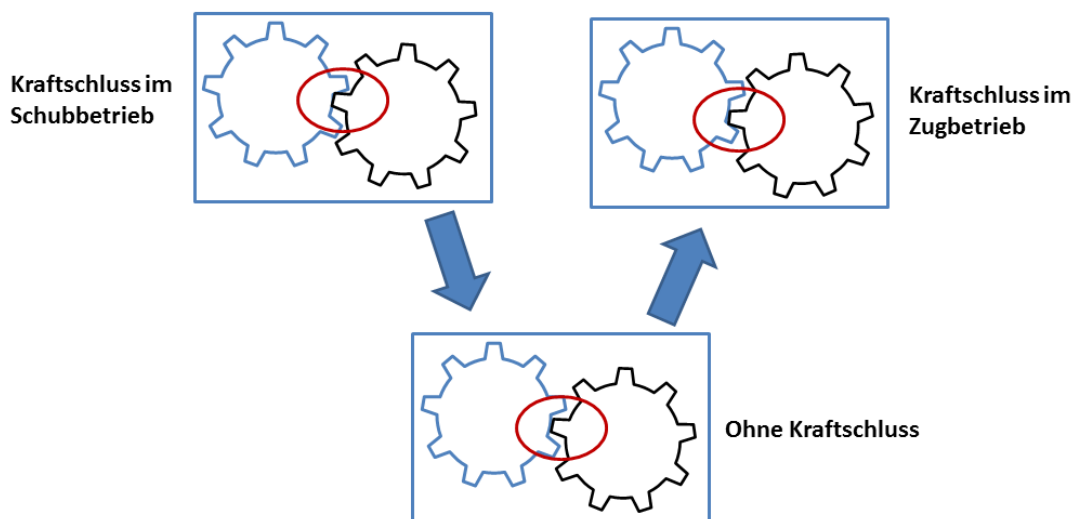


Abbildung 1: Umlegen der Spiele im Antriebsstrang

Um diesem gerade bei Hybridfahrzeugen aufgrund ihrer drehmomentstarken elektrischen Maschinen besonders relevanten Phänomen entgegenzuwirken, findet hier bei der Daimler AG eine Lastschlagdämpfungsfunktion Anwendung. Diese Funktion ist eine Softwarelösung, die aus dem Fahrerwunschmoment eine im Sinne der Lastschlagdämpfung optimierte Momentenvorgabe für den Antriebsstrang formt

Die Lastschlagdämpfungsfunktion greift

zur individuellen Anpassung an die Baureihe, das Fahrprogramm und die Motorisierung auf Kennfelder zurück. Die Bedatung dieser Kennfelder erfolgt händisch. Aufgrund des manuellen Vorgehens besteht an dieser Stelle ein Fehlerpotential und damit die Notwendigkeit, den Aufwand bei der Fehlersuche durch die Entwicklung einer Möglichkeit, Anomalien in Kennfeldern frühzeitig zu erkennen, deutlich zu reduzieren.

Die steigende Anzahl hybridisierter Baureihen, die zum einen durch die weltweit verschärf-

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

ten CO₂-Emissionsvorgaben und zum anderen durch die Forderung der Bevölkerung nach emissionsärmeren Fahrzeugen getrieben ist, zieht einen erhöhten Arbeitsaufwand bei der Applikation solcher Funktionen nach sich.

Um Fehlbedatungen überhaupt feststellen zu können, wurden Applikationsdatensätze der als Versuchsträger dienenden Lastschlagdämpfungs-Funktion analysiert. Die Applikationsdaten wurden hierfür aus DCM (Data Conservation format)-Container-Files ausgelesen und in Klassen mit unterschiedlichen Anforderungen unterteilt. Eine Erkenntnis daraus ist, dass eine mathematische Betrachtung der Kennfeldbeschaffenheiten Rückschluss über ihren Verlauf und ihre Auffälligkeit liefern kann. Mit den hierdurch erlangten Informationen wurden zwei vorbereitende und vier bewertende Kriterien entwickelt. Diese Kriterien liefern Kennwerte, die es erlauben, in ihrer Gesamtheit Rückschlüsse über die Auffälligkeit von Kennfeldern zu ziehen. Die Kriterien bewerten unter anderem Steigungsänderungen, Monotonie sowie die Nachbildbarkeit eines Kennfeldes durch eine Funktion definierten Grades. Anhand einer statistischen Analyse der ermittelten Kriterienkennwerte können die im Kontext eines Applikationsdatencontainer-Files auffälligen Kennfelder über dynamische Merkmalsgrenzen identifiziert werden. Diese dynamischen Grenzen ergeben sich aus einer Quantilanalyse. Auffälligkeiten werden folglich immer relativ in Bezug zu den bewerteten Kennfeldern ermittelt. Eine Einschränkung, die sich hieraus ergibt, ist, dass bei der Analyse von Datensätzen ohne mathematische Grundgesamtheit keine verwertbare Aussage über deren Auffälligkeit getroffen werden kann.

Da die im Rahmen dieser Arbeit entwickelte Methode vom direkten Bezug der Funktion entkoppelt ist, sind die Auffälligkeiten rein mathematischer Natur. Folglich können Kennfelder mathematisch auffällig sein, in der Funktion aber dennoch das gewünschte Ergebnis liefern.

Die erarbeitete Methode wurde in einer Applikation zur automatisierten Validierung kennfeldbasierter Applikationsdaten umgesetzt (siehe Abbildung 2), die es dem Anwender erlaubt, schnell die ermittelten Kriterien auf gewünschte Kennfeldcontainer-Files anzuwenden.

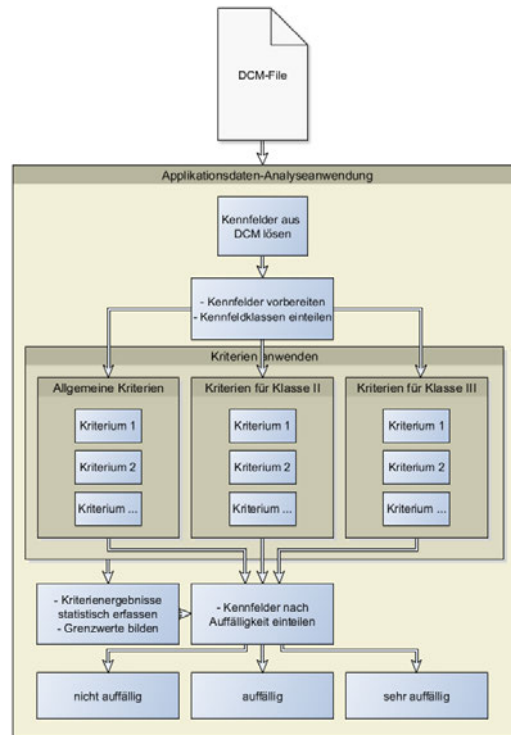


Abbildung 2: Architektur der Applikation

Diese Anwendung erlaubt es dem Benutzer, automatisiert eine grafische sowie tabellarische Auswertung der Merkmale und Auffälligkeiten im PDF-Format zu generieren und legt den Grundstein für eine automatisierte Erkennung von Fehlbedienungen. So kann sie durch das Einbringen der vorgeschlagenen Erweiterungen langfristig die Entwicklung der Lastschlagapplikation durch Zeitersparnis vorantreiben.

[1] BIERMANN, J.-W.; REITZ, A.; SCHUMACHER, T.: The Clonk Phenomenon a Load Change Reaction to be Balanced in Terms of Comfort and Engine Response. Seoul. 2000. FISITA World Automotive Congress.

[2] DRESIG, H.; FIDLIN, A.: Schwingungen mechanischer Antriebssysteme. Berlin Heidelberg: Springer. 2014.

Bildquellen:

- Abbildung 1: Eigene Abbildung
- Abbildung 2: Eigene Abbildung

Konzipierung und Realisierung der Anbindung von Ladesäulen für Elektrofahrzeuge an ein SmartHome-System einschließlich Implementierung des Lastmanagements in C++ unter Berücksichtigung der verfügbaren Anschlussleistung

Maximilian Röhrle*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Elektromobilität wird in der heutigen Gesellschaft immer wichtiger. Mehr und mehr Menschen steigen auf die umweltfreundliche Variante der Fortbewegung um. Als Folge steigt auch der Bedarf an Ladeinfrastruktur weiter an. Ein privater Haushalt stößt schnell an die Leistungsgrenze des Hausanschlusses, wenn über diesen mehr als ein Elektrofahrzeug gleichzeitig geladen werden soll. Aus diesem Grund ist ein Lastmanagement, welches die

Leistung der Ladesäulen regelt, sehr wichtig. Für den Aufbau des Lastmanagements zu Ladesäulen benötigt das betreffende System weitere Informationen über das versorgende Gebäude. Daher erscheint es sinnvoll, das Lastmanagement in das SmartHome-System der Enisyst GmbH zu integrieren, zumal die benötigten Daten zu einem großen Teil in dem System verfügbar sind.

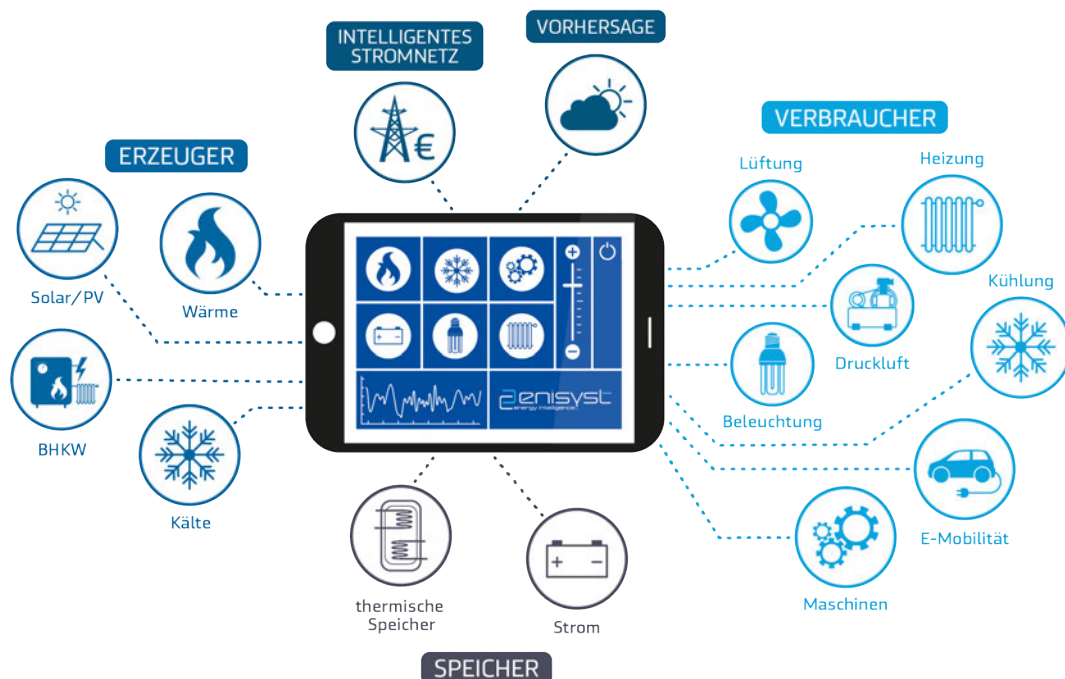


Abbildung 1: Systemansicht des SmartHome-Systems enisyst

Die Kommunikation mit dem Elektrofahrzeug übernimmt die Ladesäule selbst. Sie teilt dem Elektrofahrzeug mit, wie viel Leistung zum Laden zur Verfügung steht. Die Übertragung der Daten erfolgt in Europa bei üblichen Steckverbindern Typ 2 über eine Pulswei-

tenmodulation. Die Aufgabe des SmartHome-Systems besteht also darin, der Ladesäule nur eine limitierte Leistung zur Verfügung zu stellen, und so ein erfolgreiches Lastmanagement zu implementieren. Diese benötigte Limitierung erfolgt über einen veränderbaren Wider-

*Diese Arbeit wurde durchgeführt bei der Firma Enisyst GmbH, Pliezhausen

stand im Steckverbinder der Ladesäule. Über diesen Widerstand wird die maximale Leistung kodiert. Die Ladesäule misst den Widerstand und begrenzt, je nach Codierung, die maximale Leistung.



Abbildung 2: Beispiel einer Ladesäule

Ein erfolgreiches Lastmanagement muss zudem automatisch erkennen, ob ein Fahrzeug angeschlossen ist und mit welcher maximalen Leistung dieses geladen werden kann. Hier gibt es sehr große hersteller- und modellspezifische Unterschiede. Der Renault Twizy kann zum Beispiel nur mit einer Leistung von 2,3 kW geladen werden [1]. Das sind lediglich 10 Ampere einphasig, was zu einer Ladezeit von circa 10 Stunden führt. Ganz anders hingegen ist es bei dem Tesla Model S. Das Fahrzeug kann mit bis zu 22 kW an 400 VAC geladen werden. Dies entspricht 32 Ampere dreiphasig [2].



Abbildung 3: Typ 2 Ladestecker

Es ergibt sich also ein sehr großer Unterschied zwischen einzelnen Elektrofahrzeugen. Um einen möglichst kurzen Ladevorgang der Elektrofahrzeuge zu erreichen, muss also immer ermittelt werden, wie viel Leistung welches Fahrzeug nach Anschluss an die Ladesäule aufnehmen kann. Dies wird erreicht, indem man die für die Ladestation verfügbare Leistung in definierten Schritten erhöht und misst, wie viel Leistung tatsächlich von dem jeweiligen Elektrofahrzeug aufgenommen wird. Insgesamt führt dies zu einer Klassifizierung der angeschlossenen Fahrzeuge, mit der das Lastmanagement seine Ladestrategie optimieren kann. Zudem müssen auch verschiedene Ladeszenarien beherrscht und verschiedene Strategien zur Leistungsverteilung auf die Ladesäulen eingesetzt werden. Das wohl häufigste Szenario für das Laden eines Elektrofahrzeugs ist folgendes: Ein Fahrzeug wird zum Pendeln genutzt und muss nach dem Arbeitstag über die Nacht bis zum nächsten Morgen geladen werden. In diesem Szenario genügt es, das Elektrofahrzeug mit einer relativ geringen Leistung zu laden, da ja sehr viel Ladezeit zur Verfügung steht. So können in diesem Fall andere Fahrzeuge mit weitaus höherer Leistung geladen werden, wenn diese zum Beispiel schon nach einer Stunde Ladezeit wieder betriebsbereit sein sollen. Dies führt zwingend zu einer weiteren Konstellation: Ein Fahrzeug soll so schnell wie möglich wieder betriebsbereit sein und für eine bestimmte Kilometerleistung geladen sein. In diesem Fall ist es besonders wichtig, dass das Lastmanagement richtig arbeitet und, falls möglich, dieses Fahrzeug entsprechend priorisiert. Das priorisierende Szenario bestimmt zwangsweise die Strategie des Lastmanagements. Denn bei einer ausschließlich „fairen“ Verteilung der Leistung durch das Lastmanagement unter den zu ladenden Fahrzeugen ist eine Schnellladung nicht möglich. Im Rahmen der Arbeit wurde ein Lastmanagement in das SmartHome-System enisyst integriert, sowie die Schnittstellen des Systems zu den Ladesäulen definiert und in C++ implementiert. Das Lastmanagement verfügt über verschiedene Strategien zur Regelung der Leistung, um den oben genannten Ladeszenarien gerecht zu werden.

[1] Wikipedia: Renault Twizy, <https://de.wikipedia.org> (Stand: 11.05.2016)

[2] Wikipedia: Tesla Model S, <https://de.wikipedia.org> (Stand: 11.05.2016)

Bildquellen:

- Abbildung 1,2: Enisyst GmbH
- Abbildung 3: Wikipedia, <https://de.wikipedia.org>

Erweiterung eines Expertensystems zur automatischen Auswertung von Messdaten in der Lenkungsentwicklung

Fatih Emin Sahin*, Werner Zimmermann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Durch die Modernisierung und Entwicklung der Lenksysteme in der Automobilwelt, stoßen Ingenieure immer wieder auf neue technische Hindernisse, die zu bewältigen sind. Eine elektromechanische Servolenkung, welche heutzutage eingesetzt wird, ermöglicht eine genauere Parametrisierung der Lenkung als eine elektrohydraulische Servolenkung, die eine Generation zuvor zum Einsatz gekommen ist.

Die elektromechanische Servolenkung ist im Wesentlichen eine geschwindigkeitsabhängig geregelte elektrische Hilfskraftlenkung. Die elektromechanische Lenkkraftunterstützung ermöglicht es, durch entsprechende Parametrisierung das Lenkgefühl hoch dynamisch regeln zu können, so dass Sport- bzw. Komfortlenkungen realisiert werden können. Die wachsenden Ansprüche an die Technik setzen voraus, dass mehr Aufwand und Ingenieurs-Know-how in die Entwicklung zu investieren ist.

In diesem Zusammenhang wird ADAE (Automatische Datenauswertung zum Expertensystem) im Rahmen einer Bachelorarbeit entwickelt. ADAE ist ein mit VBScript (ähnlich zu VBA) im DIAdem-Umfeld geschriebenes Programm, welches bei der Bewertung von nahezu 400 Fahrmanövern zum Einsatz kommen wird. ADAE setzt auf das kontinuierlich weiterentwickelte ADAL (Automatische Datenauswertung Lenkung) auf und ist somit eine Erweiterung bzw. eine Ergänzung zu ADAL. Die Hintergründe vor dem ADAL und ADAE entwickelt werden, ist, dass bisher die Lenkungsabteilung manuell Versuche geplant, durchgeführt, gemessen, ausgewertet, beurteilt, dokumentiert und freigegeben hat. Dieser aufwendige Prozess soll durch ADAL und ADAE vereinfacht werden.

Die Fahrer der Lenkungsabteilung haben langjährige Erfahrungen und können auch mit einer subjektiven Einschätzung (z.B. des Lenkgefühls) Messungen beurteilen bzw. einordnen. Der Trend geht jedoch dahin, dass die Messungen der Fahrmanöver auf HiL (Hardware in the Loop)-Systeme übertragen werden

sollen. Dadurch fällt der Bezug zwischen Lenkgefühl und Messwerten weg. Eine systematische Weiterentwicklung einer Lenkung setzt voraus, dass mittels Messwerten Bewertungen getroffen werden, die nicht mehr von subjektiven menschlichen Empfindungen abhängen. Deshalb soll ADAE Bewertungen des Lenkgefühls in Zahlen fassen, in dem es von Messung zu Messung gezielt Erfahrungswerte statistisch auswertet und in eine Datenbank aufnimmt.

Ziel von ADAE ist das Lernen aus den lenkungsspezifischen Vergleichswerten aller laufenden Messungen. ADAE soll auf Basis menschlicher Bewertungen einer Messung nach einem Lernprozess Grenzwerte anpassen können. Dabei gibt es harte sowie weiche Grenzwerte, die zu berücksichtigen und individuell für jedes Fahrmanöver speziell durch den Lernprozess zu ermitteln sind.

Im Konkreten sieht das ganze folgendermaßen aus: Der Fahrer fährt ein Fahrmanöver und misst mit seinem Messsystem alle lenkungstechnisch relevante Größen wie das Lenkmoment, Lenkradwinkel, Querbewegung, Gierrate, Fahrzeuggeschwindigkeit usw. Im Anschluss bewertet der Fahrer im ADAL-Umfeld die Reaktion der Lenkung auf das Fahrmanöver mit grün (für gut), gelb (für befriedigend) oder rot (für schlecht). ADAE identifiziert relevante Kennwerte und Kurvenverläufe sowie Grenzwerte und bildet Verknüpfungen zwischen menschliche Bewertungen und mathematisch erfassbaren Erkenntnissen. Mit der Zunahme von grün, gelb oder rot bewerteten Fahrmanövern kristallisieren sich mathematische Zusammenhänge heraus, die dazu beitragen, weitere Messungen nun ausschließlich auf Basis dieser mathematischen Zusammenhänge zu bewerten. Somit ist das Programm durch seine Datenbankhistorie lernfähig, denn es lernt mit der Zeit, welche Grenzwerte grün-, gelb- oder rote Ergebnisse erzeugen. Nach jeder Messung ist ADAE fähig, Messungen zu vergleichen, sie einzustufen und Grenzwerte sowie andere Zielwerte, immer wieder neu anzupassen bzw. zu verbessern. Irgendwann wird ADAE eine so große

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

Datenbank haben, dass eine durchschnittliche menschliche Meinung Bewertung eines Fahrmanövers automatisch nachgebildet werden kann, sodass das System in der Lage ist, Fahrmanöver am HiL-System zu simulieren, welches eine immense Zeit- und Kostenersparnis darstellt und zur schnelleren Verbesserung der Fahreigenschaften beiträgt.

Wenn das System eingelaufen ist wird selbstverständlich weiterhin stichprobenartig vom Fahrer selbst das Fahrmanöver nachgefahren und mit dem Ergebnis von ADAE verglichen. Falls Abweichungen auftreten sollten wird die ADAE-Datenbank entsprechend angepasst, sodass die Datenbank stets qualitativ hochwertige Daten beinhaltet und ständig verbessert wird.

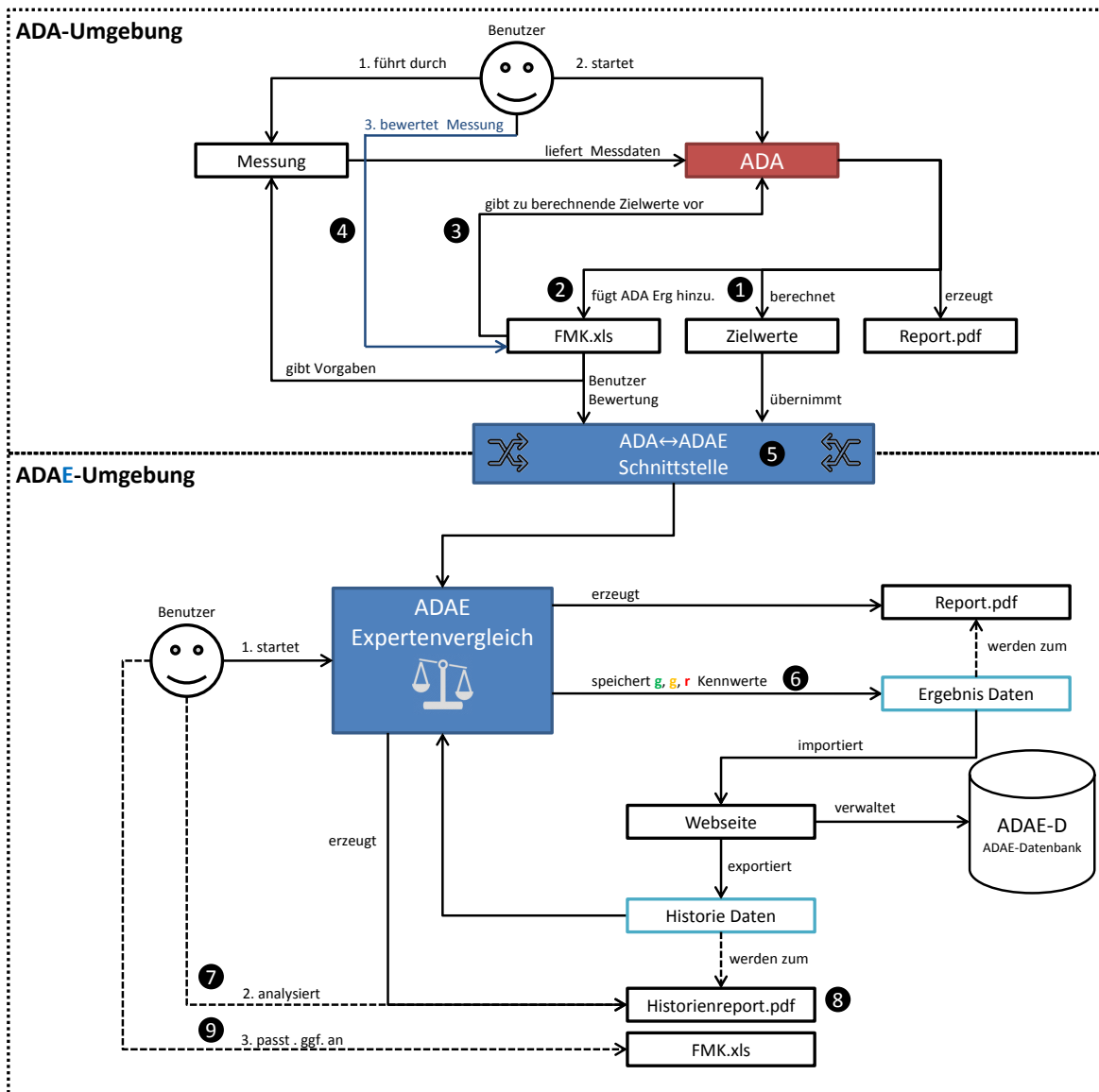


Abbildung 1: ADAL & ADAE Konzept

Bildquellen:

- Abbildung 1: Eigene Darstellung

Abhängigkeiten reduzieren mit Flow Design

Maximilian Schall*, Harald Melcher, Joachim Goll

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

„Abhängigkeiten sind eine der größten Geißeln der Softwareentwicklung.“[1] Doch was sind Abhängigkeiten und warum sind sie schlecht?

Eine Komponente ist immer dann abhängig von einer anderen Komponente, wenn sie diese zum Arbeiten benötigt. Das Problem dabei ist, dass durch eine übermäßige Nutzung von Abhängigkeiten der Code zunehmend schlechter wird. Es ist beispielsweise nur möglich, eine Komponente zu testen, wenn die von ihr verwendeten Komponenten auch verfügbar sind oder durch Attrappen ersetzt worden sind. Außerdem wird Code durch Abhängigkeiten schwerer verständlich, da Kenntnisse über die Funktionsweise der unabhängigen Komponenten nötig sind, um eine abhängige Komponente zu verstehen. Zusätzlich erhöhen viele Abhängigkeiten die Komplexität eines Systems drastisch, so dass es immer schwerer zu über-schauen und zu ändern ist.

Entwicklung von Flow Design

Diese Probleme waren unter anderem Anlass für Ralf Westphal, zusammen mit Stefan Lieser die Entwurfsmethode „Flow Design“ zu entwickeln.

Als weiterer Anreiz für Flow Design dien-te der Gedanke von Alan Kay, welcher 1966 die Grundidee zur Objektorientierung hatte. Dieser dachte sich, dass Objekte über unidirektionale Nachrichten kommunizieren sollten, wodurch sichergestellt würde, dass der Sender unabhängig vom Empfänger ist. Dieser Gedanke ist für Ralf Westphal in der heutigen objektorientierten Programmierung nur unzu-reichend umgesetzt.

Ein weiteres Problem in der aktuellen Softwareentwicklung sah Ralf Westphal in der schlechteren Lesbarkeit des produzierten Codes. Da für ihn die Lesbarkeit des Codes eines der wichtigsten Qualitätsmerkmals von Code ist, wollte er die Lesbarkeit des Codes mit Flow Design ebenfalls verbessern.

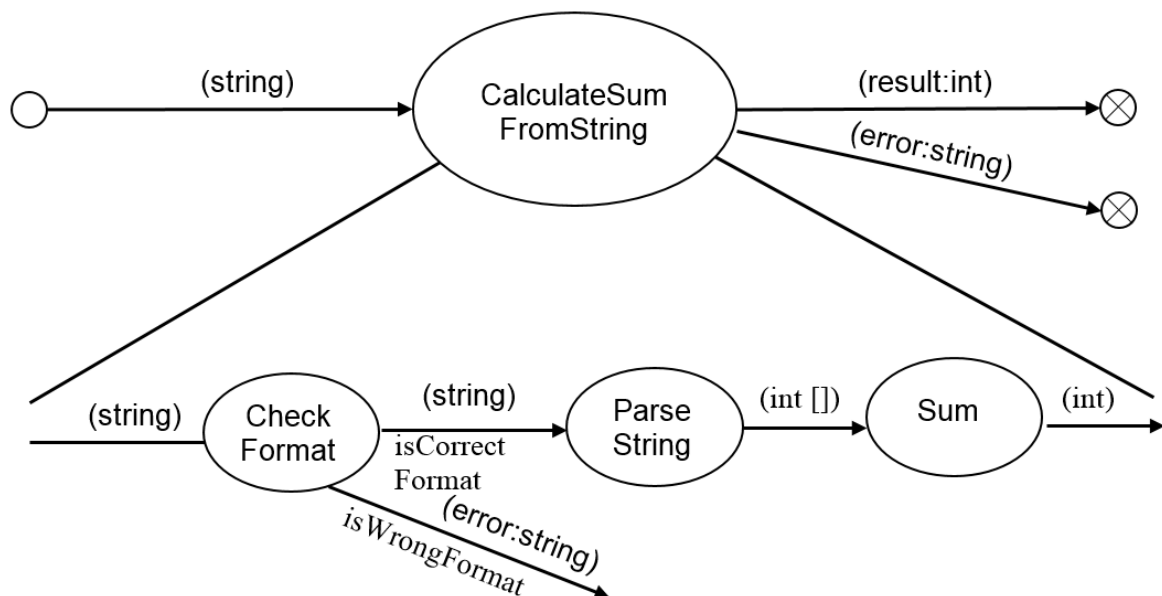


Abbildung 1: Flow Design-Diagramm

Bewertung von Flow Design

Flow Design ist eine Entwurfsmethode, bei der – anders als in der objektorientierten Programmierung – nicht nach Objekten ge-

sucht wird. Stattdessen wird – ähnlich wie bei der Suche nach Use Cases – nach Verhaltensweisen gesucht, die das System bieten soll. Ferner werden die in das System ein-

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

gehenden und ausgehenden Daten ermittelt. Falls eine Verhaltensweise zu komplex ist – wie im gezeigten Beispiel –, wird diese in kleinere, leichter umsetzbare Verhaltensweisen unterteilt. Im Beispiel sind beispielsweise `CalculateSumFromString` oder `CheckFormat` Verhaltensweisen, die im Kontext von Flow Design Funktionseinheiten genannt werden.

Durch die konsequente Verwendung von Flow Design entstehen viele Vorteile im entwickelten Code. Der Code hält automatisch viele anerkannte Prinzipien der Softwareentwicklung wie etwa das Prinzip „Single Level of Abstraction“ oder das „Single Responsibility Principle“ ein. Außerdem erhält der Entwickler durch die Arbeit mit Flow Design und dessen Verwendung ein besseres Verständnis für die automatisch eingehaltenen Prinzipien und begreift deren Sinnhaftigkeit.

Dazu tragen vor allem die zwei zu Grunde liegenden Prinzipien von Flow Design bei: Durch das „Integration Operation Separation Principle“ (IOSP) wird das „Was wird getan“ vom „Wie wird es getan“ getrennt. Dieses Prinzip teilt Funktionseinheiten in Operationen und Integrationen ein, wobei Operationen nur für die Logik zuständig sein sollen. Für das Verbinden der Operationen sollen hingegen sogenannte Integrationen benutzt werden, welche die Logik, die in Operationen gekapselt ist, miteinander verbinden. Dies ermöglicht ein einfacheres Lesen des Codes und somit ein schnelleres Verständnis für diesen. Durch die verkürzte Einarbeitungszeit in den Code ist es für Entwickler somit schneller möglich, Fehler im existierenden Programmcode zu finden und zu beheben.

Durch das „Principle of mutual oblivion“ (Prinzip der gegenseitigen Nichtbeachtung – PoMO) wird sichergestellt, dass einzelne Funktionseinheiten unabhängig voneinander sind.

Wenn Funktionseinheiten keine Abhängigkeiten untereinander aufweisen, ist es leichter, diese automatisch – beispielsweise mit Unit-Tests – zu testen, da keine Komponenten durch Mocks simuliert werden müssen. Zusätzlich wird es durch diese Reduzierung der Entwicklungsmöglichkeiten, den Code schneller zu verstehen.

Durch das vorherige Erstellen eines Flow Design-Diagramms wird zudem die spätere Implementierung einfacher, da dieses Diagramm mehr oder weniger nur noch in die jeweilige Programmiersprache übersetzt werden muss. Anders als bei vielen anderen Methoden, die mehrere Diagramme erfordern, kann dieses Diagramm als einzige Grundlage für die Implementierung dienen.

Abschließend bleibt die Frage offen, ob sich Flow Design gegen die bisherigen Methoden der objektorientierten Analyse und Modellierung durchsetzt.

Durch eher wenig öffentlich zugängliche und überwiegend deutsche Literatur zu Flow Design, die überdies nur in Bruchstücken und nicht in Form eines Lehrbuches zur Verfügung steht, wird diese Methode bisher überwiegend im deutschsprachigen Raum und auch dort nur vereinzelt benutzt. Inwieweit diese Methode auch Anerkennung in internationalen Softwareprojekten findet, bleibt abzuwarten.

Auch der radikale Kontrast zu den bisherigen Methoden macht es für Flow Design schwer, Fuß zu fassen. Allerdings gibt es Möglichkeiten, Flow Design im kleinen Maßstab einzuführen. So können beispielsweise einzelne Funktionen als Integration betrachtet werden und diese visuell mit einem Flow Design-Diagramm dargestellt werden. Das Diagramm kann später dann genauso implementiert werden.

[1] Ralf Westphal, One Man Think Tank, Abhängigkeiten bewusster wahrnehmen: <http://blog.ralfw.de/2010/09/abhangigkeiten-bewusster-wahrnehmen.html>, (Zugriff am 10.05.2016)

Bildquellen:

- Abbildung 1: Eigene Abbildung

Was ist Testen?

Markus Scheider*, Hans-Gerhard Groß, Joachim Goll

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Was ist Testen?

Testen ist eine der wichtigsten analytischen Qualitätssicherungsmaßnahmen im Entwicklungsprozess. Es wird an bereits realisierten Produkten durchgeführt. Diese sind sogenannte Testobjekt oder Testlinge, in der Regel das zu testende System oder das zu testende Zerlegungsprodukt. Jedes relevante Dokument und jedes Programm sollte getestet und abgenommen sein, ehe es abgegeben wird und eventuell in den Einsatz geht. **Dabei werden Tests mit der Absicht durchgeführt, Fehler zu finden.** Somit soll ein Test die Anwesenheit von Fehlern zeigen und nicht die Abwesenheit von Fehlern. Beim Testen wird der Unterschied zwischen dem tatsächlichen und dem geforderten Verhalten eines Systems bzw. Systemteils untersucht. Prinzipiell ist Testen also eine Bewertung, die gegen ein Maß vorgenommen werden muss. Das Maß wird hier Spezifikation genannt und muss selbstverständlich bekannt sein. Diese Spezifikationen können in unterschiedlicher Form vorkommen, wie ausgewiesene Requirements, Use Cases, User Stories, Testfällen oder in sonstiger Form. Um dieses Maß zu definieren, wird die Fähigkeit benötigt, für spezifizierte Routinen ihr Ergebnis vorauszusagen. Dies ist das sogenannte **Orakelproblem**. Man muss beurteilen können, ob die entsprechende Implementierung korrekt arbeitet oder ob sie ein Fehlverhalten aufweist. Beim Testen von Software werden Funktionen gegenüber Spezifikationen getestet. Zu beachten ist, dass es sich bei Softwaretests immer um ein **Stichprobenverfahren** handelt. Bei komplexen Systemen ist ein vollständiger Test, das heißt die Überprüfung aller möglichen Kombinationen von Daten in der Praxis nicht durchführbar.

Validierung und Verifikation

Im klassischen Software Engineering kennzeichnet „Verifizieren“ das Testen eines Zerlegungsproduktes oder des Systems gegen die aufgeschriebenen Anforderungen. „Validieren“ steht zusätzlich zur Verifikation für das Testen eines Systems in seiner Einsatzumgebung. Dabei wird geprüft, ob das System in der Praxis einen Nutzen bringt. Bei einer Verifikation eines Systems wird gegen die Anforderungen

geprüft, ob sich das System wie spezifiziert verhält. Bei einer Validierung wird zusätzlich zur erforderlichen Verifikation überprüft, ob das System eigentlich das ist, was der Nutzer wollte. Spezifikationsorientiert kann das Gesamtsystem validiert werden, bei einer agilen Entwicklung kann das gerade entwickelte Fragment verifiziert werden.

Verifikation

Bei der Verifikation eines Systems wird das fertige System gegen die mit dem Kunden abgestimmten Anforderungen an das System verglichen. Bei der Verifikation von Teilsystemen werden die von den Entwicklern selbst aufgestellten Requirements überprüft, es handelt sich bei der Verifikation von Teilsystemen um einen internen Qualitätsprozess. Zudem wird bei der Verifikation von Zerlegungsprodukten das Ergebnis einer Entwicklungsaktivität auf die Korrektheit aus reiner Entwicklersicht untersucht, bei beiden Fällen erfolgt das Verifizieren mit gestellten Testfällen und Daten.

Validierung

Die Validierung prüft das finale Produkt des Entwicklungsprozesses. Bei der Validierung ist der Kunde der Prüfer oder ist zumindest mit dabei, da nur der Kunde die „wahren Anforderungen“ kennt. Deswegen ist eine Validierung mehr wert als nur die Überprüfung der Anforderungen in der Verifikation. Die Validierung stellt einen externen Qualitätsprozess dar. Die Validierung erfolgt mit echten Anwendungsszenarien, die Live-Daten enthalten. Somit lassen sich fehlerhafte Anforderungen entdecken.

Testfälle

Unter einem Testfall versteht man einen kompletten Satz von Vorbedingungen für ein Testobjekt, die ein- und auszugebenden Daten während des Tests sowie alle zugehörigen Ergebnisdaten (Nachbedingungen). Es wird zwischen Testfällen unterschieden, die das spezifizierte Verhalten und das nicht-spezifizierte Verhalten testen. **Spezifiziertes Verhalten** schließt Normalwerte, Grenzwerte und Ausnahmebehandlung ein. **Nichtspezifiziertes Verhalten** schließt Fehlerwerte ein. Die zu den einzelnen Testfällen benötigten Testda-

*Diese Arbeit wurde durchgeführt bei der Firma IT-Designers, 73730 Esslingen

ten setzen sich zusammen aus vorbereiteten Testdaten zur Bereitstellung der Testumgebung und zur Übergabe an das Testobjekt, des Weiteren aus Daten, die während des Tests einzugeben und auszugeben sind und aus Ergebnisdaten, die sich zusammensetzen aus den angezeigten Daten und den gespeicherten Daten. Die Testumgebung beinhaltet die übrige Software, die das Testobjekt benötigt, mit Quellcode und Systemdokumentation, ggf. die (vorläufige) Bedieneranleitung, die erforderliche Hardware-Umgebung, die erforderliche Betriebssystem-Umgebung, die erforderlichen Testwerkzeuge und das für den Test erforderliche Testpersonal.

Testautomatisierung

Die Testautomatisierung ist ein wichtiger Teil des Testens. Jedoch unterliegt diese der falschen Annahme, dass sich eine Testautomatisierung sofort lohnt. Das Gegenteil ist der Fall. Das Anlegen von Testfällen für die Automatisierung erhöht zunächst den Aufwand und ist eine Investition für die Zukunft. Cem Kaner [2] rechnet mit einem 3 - 10 mal höheren Aufwand zur Erzeugung der Tests. Erst im späteren Verlauf kann sich die Testautomatisierung wegen der höheren Effizienz bei der Testdurchführung, der geringeren Kosten für die Wartung der Testfälle und der Wiederverwendung von Testfällen auszahlen. Die Testautomatisierung führt nicht automatisch dazu, dass mehr Fehler entdeckt werden. Erst wenn die bei der Testdurchführung gewonnene Zeit dazu genutzt wird, mehr Testfälle einzusetzen, kann sich die Testqualität erhöhen. Die Testerstellung und Testauswertung lassen sich dabei nur in Grenzen automatisieren.

Testauswertung

Bei der Auswertung eines Tests wird geprüft, ob die erwarteten Solldaten mit den Testergebnissen übereinstimmen. Außerdem muss beurteilt werden, ob die bearbeiteten Testfälle ausreichen. Beim Testfall selbst ist zu analysieren, ob Abweichungen des Testobjekts (Testlings) beim Soll-/Ist-Vergleich des Programms aus anderen Gründen als Fehler des Testobjektes herrühren. Ursachen hierfür sind beispielsweise Fehler des Testbereichs, der Testumgebung, eine mangelhafte Testspezifikation oder fehlerhafte Testfälle. Prinzipiell gehört die Fehlerbeseitigung nicht zu den Testaktivitäten, obwohl sie oft eng verzahnt mit dem Testen durchgeführt wird. Jedoch muss die Beseitigung von Fehlern überprüft werden.

Testende

Für das Ende des Testens werden in der Regel ein oder mehrere Kriterien vorgegeben. Ein Testende-Kriterium dient dazu, einen Test zeitlich zu begrenzen und einen objektiv messbaren Testumfang sicherzustellen. Ein

solches Kriterium ist oft die gefundene Anzahl von Fehlern pro Zeiteinheit. Nach Unterschreiten dieser vorgegebenen Anzahl ist das Testen nicht mehr wirtschaftlich. Diese Grenze richtet sich nach der Art der Software. Im ungünstigsten Fall wird das Testende durch den Abgabetermin oder den Verbrauch der Finanzmittel bestimmt, deswegen sollten die kritischsten Testfälle zuerst getestet werden.

Qualitäten

Testen hat auch starke Auswirkungen auf die Qualität der Software. Qualität schließt die funktionale Qualität ein, das heißt Funktionalität und Fehlerfreiheit, außerdem die strukturelle Qualität und die Qualität der Abläufe. Die Abhängigkeiten der verschiedenen Systemteile sollen minimiert werden, um das System wandelbar zu gestalten (Evolvability). Diese Qualität ist für den Entwickler sichtbar, für den Kunden ist sie erst in der Wartung sichtbar. Des Weiteren sollte der Umgang mit dem System in einfacher Weise erfolgen. Dies betrifft beispielsweise die Bedienbarkeit des Systems, seine Dokumentation aber auch die Testbarkeit. Testen allein ist jedoch kein Garant für Qualität, da die Testergebnisse davon abhängen, wie die Testfälle geschrieben werden und außerdem nur Stichproben sind. Zum anderen kann es beim Testen bereits zu spät sein. **Qualität muss in das System hinein konstruiert werden.**

Wichtige Tests

Es gibt Haupttests, die jedes System durchlaufen sollte. Dazu zählen Unit Tests, Integrationstest und Akzeptanztests. Je nach Situation sind weitere Tests erforderlich wie Performancetests, Stresstests, Tests zum Auffinden von Sicherheitslücken, Tests der Grafischen Oberfläche oder Regressionstests.

Unit Tests

Das Ziel von Unit Tests ist es, die verschiedenen Teile eines Programms zu isolieren und unabhängig voneinander auf Fehler in der Implementierung zu prüfen. **Wenn man Systemteile isoliert testen will, muss bereits geeignet programmiert werden.** Man kann eine Unit als die kleinste testbare Einheit einer Applikation bezeichnen. Unit Tests finden Probleme früh im Entwicklungszyklus, da Units die Grundlage des Testens sind. Unit Tests werden während des Entwicklungsprozesses üblicherweise vom Softwareentwickler geschrieben und durchgeführt, um sicherzustellen, dass der Code das Design trifft und sich wie erwartet verhält.

Was ist eine Unit?

Eine Unit ist ein Modul des Systems. **Eine Unit ist die kleinste Codeeinheit, die isoliert getestet werden kann.** Ein Modul muss also unabhängig von den anderen Modulen der Anwendung sein. Das ist eine wichtige Einschränkung.

kung! Eine Unit ist eine Übersetzungseinheit der verwendeten Programmiersprache. In der objektorientierten Programmierung sind Units üblicherweise Klassen, sie können aber auch individuelle Methoden sein. In der Regel werden die öffentlichen Schnittstellen einer Unit mit einem oder mehreren Unit Tests getestet. **Was ein Modul ist, ist nicht einheitlich definiert.** Objektorientierte Entwickler denken bei einem Modul gern an eine Klasse, die prozeduralen oder funktionalen Ansätze hingegen an eine einzige Funktion. Aber es hängt ganz von der Situation ab!. Das Team entscheidet, was sinnvollerweise als Einheit des Systems und seinen Test genommen werden soll.

Performancetest

Beim Lasttest oder Performancetest beobachtet man das Verhalten des Systems unter einer möglichen, sehr hohen Last. Man testet das System mit zunehmender Last bis zur Grenze der maximal zulässigen Last. Ziel ist es, Fehler zu finden, die im funktionalen Test nicht gefunden werden. Zudem ist das Verhalten nicht funktionaler Forderungen zu überprüfen. Ebenfalls können sich Fehler in den Nebenläufigkeiten (Parallelisierung) zeigen.

Stresstest

Beim Stresstest testet man punktuell außerhalb der gültigen Werte, das heißt unter abnormalen Bedingungen. Man kann Aussagen darüber gewinnen, ob ein System beim Überschreiten der Lastgrenze noch arbeitet. Es lässt sich sagen, wie das System sich zeitlich verhält, ob es abstürzt und ob die Daten konsistent bleiben. Nach Beseitigung der Überlastsituation zeigt sich auch, ob das System wieder in den normalen Zustand zurückkehrt.

Regressionstests

Ein Regressionstest ist ein erneuter Test eines bereits getesteten Programms nach dessen Modifikation. Das Ziel ist die Sicherstellung, dass durch die vorgenommen Änderung keine Fehler hinzukommen. Ebenfalls wird ermittelt, ob Fehler in unveränderten Teilen der Software freigelegt wurden.

Integrationstest

Integration bedeutet das Zusammenfügen von Modulen zu Teilsystemen bzw. von Teilsystemen zum fertigen System. Jede Integrationsstufe bis zum fertigen System muss daher getestet werden. Beim Integrationstest wird das Zusammenspiel vieler Module untersucht. Insbesondere Spezifikationsfehler kommen beim Integrationstest zum Vorschein. Die geläufigste Art der Integration ist die Bottom-up-Integration, da die Module als erste Zerlegungsprodukte beim Programmieren erstellt und getestet werden.

Akzeptanztests

Der Akzeptanztest eines Systems wird vom Auftraggeber durchgeführt. Ein Akzeptanztest stellt spezifikationsorientiert die letzte Phase eines Projektes dar, in der das vollständige System getestet wird. Dieser Test wird mit echten Anwendungsszenarien und Live-Daten durchgeführt. Somit lassen sich fehlerhafte Spezifikationen leicht entdecken. Diese Fehler entstehen durch unterschiedliches Interpretieren der Spezifikation durch Auftraggeber und Auftragnehmer. Bei einer agilen Vorgehensweise werden im Akzeptanztest kleine Systemfragmente getestet.

[1] J. Goll, Methoden des Software Engineering, Wiesbaden, Vieweg+Teubner Verlag, 2012

[2] Cem Kaner, Testing Computer Software, New York: Wiley, 1999

[3] Martin Fowler, Online Artikel Unit Test,

<http://martinfowler.com/bliki/UnitTest.html>, (Abgerufen am 10.05.2016)

[4] B. W. Boehm, Verifying and Validating Software Requirements and Design Specifications, IEEE Software, 1984

Auslegung, Implementierung und Inbetriebnahme eines Zustandsreglers sowie eines Fuzzy-Reglers für einen nichtlinearen Prozess mit zwei Stellgrößen zur Positionierung eines Balls in einem Ball-Wippe-System

Kerstin Schneider*, Hermann Kull, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Regelungstechnik ist ein Teilgebiet der Automatisierungstechnik. Heutzutage ist es in der Technik nahezu undenkbar darauf zu verzichten. Durch einen technischen Regelvorgang können physikalische, chemische oder andere Größen technischer Systeme gezielt beeinflusst werden. Dabei sollen die Regelgrößen auch durch Einwirkung von Störgrößen möglichst konstant gehalten werden. Mögliche Anwendungsgebiete der Regelungstechnik finden sich z.B. bei der Temperatur-, Positions-, Geschwindigkeits- oder Kraftregelung [1]. In einem Regelkreis wird die Regelgröße in der Regelstrecke gemessen und kontinuierlich mit der Führungsgröße verglichen. Das gewünschte Zeitverhalten der Regelgröße wird durch Bestimmung einer Stellgröße anhand der Regeldifferenz erreicht, die

auf die Regelgröße einwirkt und trotz vorhandener Störgrößen die Abweichung minimiert. Der „kritische Punkt“ bei dieser Bachelorarbeit ist, dass parallel zwei Messgrößen (Position des Balls und Winkel der Wippe) verarbeitet werden müssen und damit zwei Regelgrößen einen Mehrfach-Regelkreis bilden: Die gewünschte Position des Balls und der Wippenwinkel, der zu Null geregelt werden muss (Der Ball darf sich im stationären Zustand nicht mehr bewegen. Dies ist nur bei einem Winkel „Null“ der Fall). Es liegt somit kein einschleifiger Standard-Regelkreis vor, sondern eine komplexe, nichtlineare Mehrfach-Regelung. Eine solche kann prinzipiell mit einem Zustandsregler bzw. einem Fuzzy-Regler realisiert werden.

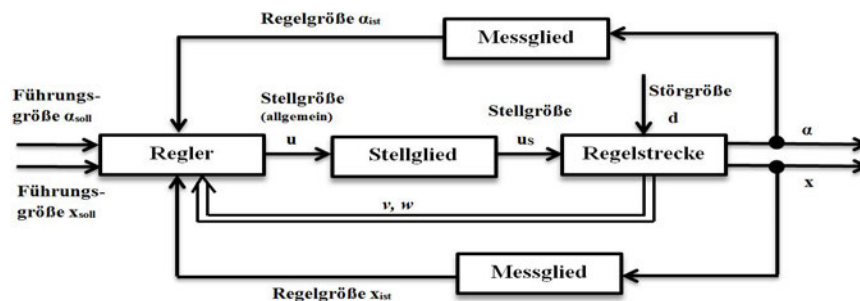


Abbildung 1: Blockdiagramm des Regelkreises

Systembeschreibung

Das Ball-Wippe-System, das im Rahmen der Bachelorarbeit verwendet wird, besteht im Wesentlichen aus einem Gleichstrommotor für die Wipp-Bewegung, einem Zahnriemen und einer Aluminium-Schiene, auf der ein Stahl- oder Squash-Ball positioniert werden soll, einem Drehgeber, der den aktuellen Winkel der Schiene misst, einer Kamera zur Erkennung der Ballposition und zwei Halogenlampen zur Beleuchtung. An der Seite des Systems sind Plexiglasplatten angebracht, um das System abzuschließen. Zwei Endschalter, die unterhalb der Wippe angebracht sind, geben an, ob diese

ihren maximalen Auslenkungswinkel erreicht hat. Folgende Abbildung 2 zeigt das eben beschriebene System.



Abbildung 2: Ball-Wippe-System

*Diese Arbeit wurde durchgeführt bei der Firma esip – Esslinger Innovationspartner, Esslingen

Zustandsregler

Eine Zustandsregelung ist im Allgemeinen ein Regelkreis, der eine Regelgröße, anhand einer Zustandsraumdarstellung, regelt. Dabei wird der Zustand der Regelstrecke durch Messung oder durch einen Beobachter der Regelstrecke zurückgeführt. Dieses Verfahren wird auch Regelung durch Zustandsrückführung genannt. Zustandsregler kommen oft bei nichtlinearen oder zeitinvarianten, sowie Mehrgrößensystemen, bei denen die Berechnung im Frequenzbereich nicht oder schwer anwendbar ist, zum Einsatz, ebenso bei schnellen Regelungen, die eine hohe Regelgüte fordern. Da das Konzept des Zustandsreglers des Ball-Wippe-System ein lineares Modell erfordert, wird es zunächst um einen bestimmten Arbeitspunkt linearisiert. Über eine Messeinrichtung und den Zustandsregler entsteht die Rückkopplung, die zusammen mit der Regelstrecke den Regelkreis bildet. Da Messungen von Zustandsgrößen aufwendig sind, werden Beobachter eingesetzt. Diese bestehen analog zur Regelstrecke aus Beobachter-Zustands-Differenzialgleichungen, einer Beobachter-Ausgangsgleichung und dem Beobachtungsvektor. Anschließend wird der Ausgang des Beobachters mit dem der Regelstrecke verglichen und über den Beobachtungsvektor auf eine Beobachter-Zustands-Differenzialgleichung zurückgeführt. Abbildung 3 zeigt ein solches System mit Beobachter.

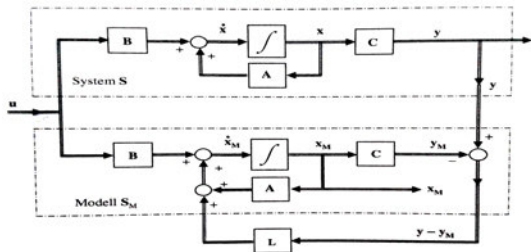


Abbildung 3: Blockdiagramm des Systems mit Beobachter

Fuzzy-Regler

Das Prinzip eines Fuzzy-Reglers besteht darin, scharfe physikalische Eingangssignale eines technischen Prozesses zu erfassen und mit Hilfe von linguistischen Begriffen über Zugehörigkeitsfunktionen und logischen Wenn-Dann-Operationen zu bewerten und den Übergang von linguistischen Variablen zu scharfen Stellgrößen zu bilden. Mit Hilfe von grafischen, meist dreieckförmigen, Fuzzy-Set-Modellen der unscharfen Mengen kommen dabei die Standardmethoden der „Fuzzifizierung“, „Inferenz mit Regelbasis“ und „Defuzzifizierung“ zur Anwendung. Die Grundidee eines Fuzzy-Reglers bezieht sich auf linguistische Variablen und Terme [2], durch die der Fuzzy-Regler optimal an einen nichtlinearen Prozess mit mehreren Ein- und Ausgangsgrößen modelliert wird, ohne dass das mathematische Modell des Prozesses vorliegt. Abbildung 4 zeigt linguistische Terme für die Ballposition x und den Wippenwinkel α des Systems.

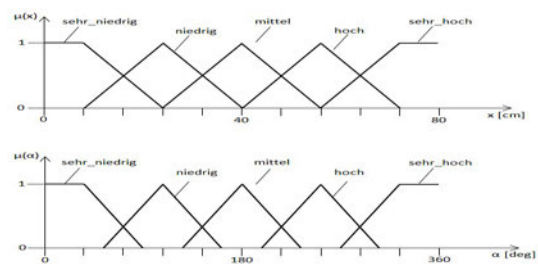


Abbildung 4: Linguistische Terme

Ziel

Für das Ball-Wippe-System soll vollständig in MATLAB/Simulink ein geeigneter Zustands- sowie Fuzzy-Regler entworfen und anhand von Simulationen und später an der realen Anlage getestet werden. Durch die Toolbox „Simulink Desktop Real-Time“ können die Regler auf den Prozess geschaltet werden (Embedded Design). Das in diesem Artikel beschriebene Ball-Wippe-System soll weiterführend zu Schulungszwecken für Masterstudenten und anderen Interessengruppen dienen.

- [1] Crashkurs Regelungstechnik, Jörg Kahlert, 2. Auflage, VDE Verlag GmbH
 [2] Fuzzy-Logik und Fuzzy-Control, Jörg Kahlert & Hubert Frank, Vieweg

Bildquellen:

- Abbildung 1,2,4: Eigene Darstellung
- Abbildung 3: Regelungstechnik für Ingenieure, Serge Zacher & Manfred Reuter, 14. Auflage, Springer Vieweg

Konzipierung und prototypische Realisierung eines generischen Testsystems zum automatisierten Testen von Motorcontrollern für Servomotoren nach dem Prinzip Hardware-in-the-Loop unter Verwendung von dSpace-Komponenten sowie MATLAB Simulink und C# für die Steuerung des Systems

Robin Schotka*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Entwicklungsbegleitendes Testen gewinnt für heutige und zukünftige Entwicklungen von Systemen zunehmend an Bedeutung. Denn durch immer größer werdende Projekte und eine zunehmende Komplexität, auch im Bereich Hardware, besteht ansonsten ein hohes Risiko, dass das resultierende Produkt qualitativ nicht den Ansprüchen gerecht werden wird. Für den funktionalen Test von Komponenten erscheint der Aufbau eines Testsystems nach dem Prinzip HiL (*Hardware in the Loop*) zielführend. Beim HiL-Test wird die zu testende Komponente, auch DUT (*Device under Test*) genannt, mit einem Simulator verbunden, der die Umgebung der Komponente vollständig simuliert. Bei jedem einzelnen Test können Messwerte aufgezeichnet und durch Vergleich mit dem zu erwartenden Ergebnis ausgewertet werden.

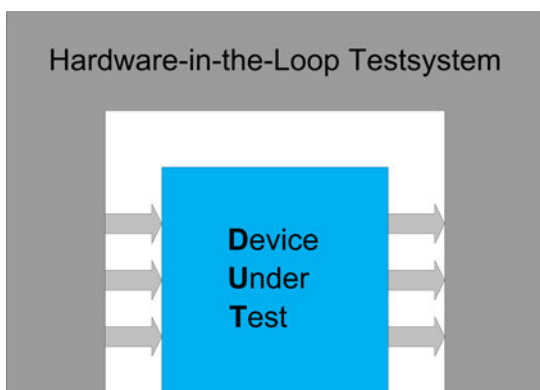


Abbildung 1: Aufbau eines HiL Testsystems

In Abbildung 1 ist der prinzipielle Aufbau eines solchen Systems zu sehen. Darin sind in Blau die zu testende Komponente und in Grau die Testumgebung dargestellt. Bei der Testumgebung kann es sich je nach Bedarf um eine oder mehrere Prozessor- sowie um FPGA-Karten handeln. Diese können beliebig kombiniert werden.

Die *API ASAM XIL*[1] ist eine standardisierte Schnittstelle für die Parametrierung und

die Kommunikation mit HiL-Komponenten. Mit dieser API ist es erstmals auf verhältnismäßig einfache Weise möglich, unterschiedliche Hardware-Komponenten, auch von verschiedenen Herstellern, in einem HiL-Prüfstand zu vereinen und diese über dieselbe Schnittstelle anzusteuern. Es ist sogar möglich, die Hardware von dSpace mit einem eigenen Programm oder auch mit der Software anderer Firmen wie National Instruments oder Vector Informatik anzusteuern. Die API berücksichtigt ebenso die Zeitversätze zwischen den einzelnen Komponenten, die einzelnen Signale werden direkt synchronisiert.

Im Rahmen der Bachelorarbeit wurde ein Konzept gesucht, das eine möglichst generische Gestaltung eines HiL-Tests ermöglicht, da durch einen generischen Ansatz viele verschiedene Testplattformen und Testkomponenten mit geringen Aufwand gepflegt werden können. Im Klartext bedeutet es, dass Testparameter, Simulationsdateien und auch die einzelnen Testfälle ohne einen Eingriff im Programmcode erstellt werden können.

Nach ausgiebiger Recherche wurde beschlossen, dass hierzu eine XML-Datei verwendet werden soll, über die ein kompletter Test mitsamt allen Testfällen, den zu messenden Signalen und den Pfaden für die Simulationsdateien konfiguriert und gespeichert werden kann. Für das Einlesen der XML-Datei und die Abarbeitung der Testfälle war eine Applikation in C# zu schreiben. Diese Applikation greift mit Hilfe der *API ASAM XIL* auf die Hardware-Komponenten des HiL-Prüfstandes zu. Zudem wird die zu testende Firmware automatisch auf den zu prüfenden Motorcontroller, die DUT (*Device under Test*), geladen. Der Pfad der Firmware, die zu verwendende XML-Datei und die Identifikationsnummern der Testfälle können dabei der Applikation direkt übergeben werden. Es wurde beschlossen, für die Auswertung der Messwerte Automation Desk von dSpace zu verwenden, da

*Diese Arbeit wurde durchgeführt bei der Firma Festo AG & Co. KG, Oberesslingen

die Auswertefunktionalitäten dieses Tools als hervorragend eingestuft werden. Zudem verfügt es über eine Kommunikationsschnittstelle, die die Übertragung von Messdaten aus einer C#-Applikation direkt ermöglicht. Diese Daten werden dann in Automation Desk verarbeitet und als HTML-Report exportiert. Zur Konfiguration kann in der XML-Datei festgelegt werden, welche Signale gemessen und wie diese im Report von Automation Desk dargestellt werden sollen. Zudem ist es möglich, ein Referenzsignal zu definieren. Dieses kann alternativ auch über eine Matlab-Simulation generiert werden. In diesem Fall wird das Gesamtsystem einschließlich der DUT in Matlab simuliert.

Für den ersten konkreten Anwendungsfall wurde ein Motorcontroller gewählt. Dieser kann in einen Leistungsteil und einen Steueranteil unterteilt werden. Für den HiL-Prüfstand soll der Leistungsteil vom Motorcontroller abgetrennt werden. Dieser Teil soll durch eine Simulation auf einer FPGA-Karte im HiL-Prüfstand ersetzt werden. Der Steueranteil wird dann ohne die Leistungselektronik mit dem Prüfstand verbunden. Die DUT, der zu testende Motorcontroller, bekommt über den mit dem HiL verbundenen Rechner seine Firmware aufgespielt. Zudem wird der Motorcontroller parametrisiert und der HiL-Test gestartet. Während des Tests werden Messwerte aufgezeichnet, die nach dem Test mithilfe von Automation Desk aufbereitet und ausgewertet werden.

Als Ergebnis der vorliegenden Arbeit ist es nun möglich, automatisierte HiL-Testsysteme mit dem generischen Ansatz zu erstellen, ohne

dafür den Quellcode des Testsystems verändern zu müssen. Dies gilt auch bei einer Änderung der Simulations-Hardware im Prüfstand. Es ist lediglich die erforderliche XML-Datei sowie ein MATLAB-Simulink-Projekt mit der gewünschten Simulation zum Download auf den Prüfstand zu erstellen.

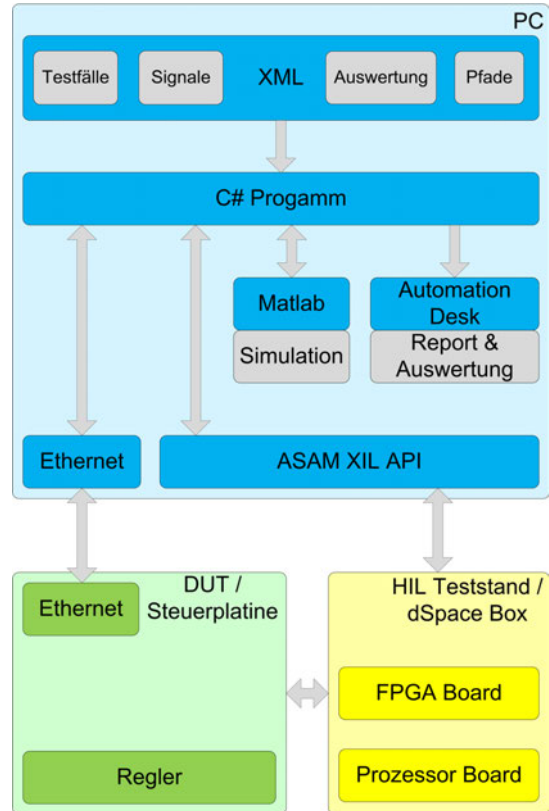


Abbildung 2: Konzeptübersicht

[1] https://vector.com/portal/medien/cmc/events/commercial_events/VectorCongress_2014/VeCo14_26Nov14_MARITIM_ECUtest_05_Rasche.pdf Stand:26.11.2014

Bildquellen:

- Abbildung 1,2: Eigene Darstellung

Entwicklung eines Feature-Trackings zur systematischen Bewertung der Usability eines Lifecycle-Management-Systems

David Seddaoui*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Softwarelösungen im Automobilbereich sind aufgrund des komplexen technischen Kontextes oft sehr schwer zu bedienen und erfordern eine lange Einarbeitungszeit. Die Untersuchung von Softwarelösungen auf Usability ist bisher in diesem Sektor noch nicht etabliert. Funktionalität kommt in der Regel vor Bedienbarkeit.

In dieser Arbeit wird als Beispiel für solch eine komplexe Softwarelösung ein Lifecycle Management System betrachtet, das aktuell in der Motorsteuergeräteentwicklung der Robert Bosch GmbH eingeführt wird. Ein Lifecycle Management System vereint alle Schritte eines Produktzyklus innerhalb einer Software. Dies ermöglicht eine Vernetzung der verschiedenen Produktzyklen und dadurch eine Steigerung der Effizienz und Übersicht.

Dies wiederum ist verknüpft mit dem Management Prozess Application Lifecycle Management, welcher genau diese Vernetzung der Segmente der Lebenszyklen eines Produktes beinhaltet. Dadurch entsteht eine Zusammenkunft von unterschiedlichen Funktionalitäten, verschiedenen Anwendungsfällen und dem sehr detaillierten und kundengetriebenen Prozess der Steuergeräteentwicklung. Dieser hoch komplexe Verbund bewirkt eine große Herausforderung, wenn daraus ein effektives, effizientes und zusätzlich noch ein intuitives und leicht zu bedienendes User Interface bereitgestellt werden soll.

Aufgrund dieser Schwierigkeit muss diese Tool Suite auf den Softwarequalitätsaspekt Usability überprüft werden, bevor sie ausgerollt wird. Diese Prüfungen wurden in der Vergangenheit auch schon durch Interviews und UX-Sessions durchgeführt [1]. Da aber die Mitarbeiter weltweit verteilt sind und nicht jede Person getestet werden kann, ist es Aufgabe dieser Bachelorarbeit ein Konzept auszuarbeiten, wie man Übersicht über die Usability des Systems mit Hilfe eines Webanalyse-Tools erhalten kann, um damit eine Verbesserung der Usability langfristig zu bewirken [2].

Während dieser Konzeptphase wurden vier Konzepte erarbeitet. Drei dieser Konzepte mussten aber aus technischen Gründen oder aus Gründen des Datenschutzes verworfen werden. Im Folgenden sind diese vier Konzepte kurz aufgeführt und die Pros und Contras beschrieben.

- System as a Service – Cloudbasierte Webanalysetools:
 - Systeme haben sich bereits im Bereich e-Commerce bewährt
 - Keine Datenhoheit gegeben
 - Server benötigt Internetverbindung
 - Hohe Kosten
- Logfile-Analyse:
 - Daten wären zentral abgelegt
 - Datenhoheit wäre gegeben
 - Art und Menge der Daten ist für diesen Anwendungsfall ungeeignet
- Serverbasiertes Webanalyse Tool:
 - Datenhoheit
 - Open-source
 - Nicht auf Anwendungsfall anpassbar
- Eigenimplementierung:
 - Datenhoheit
 - Perfekt auf den Anwendungsfall zugeschnitten
 - Einarbeitungs- und Implementierungsphase länger

Zuletzt blieb die Eigenimplementierung als Konzept übrig und wurde umgesetzt. Dabei wurde während der Bachelorarbeit ein Feature-Tracking System in prototypischer Form entwickelt und integriert. Hierzu wurde eine Schnittstelle in der Collaborative Lifecycle Management Toolsuite von IBM genutzt, mit der man JavaScript Code in die Toolsuite integrieren kann.

Durch diese Schnittstelle kann man die Interaktionen der User bis auf die Mausposition

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

verfolgen. Diese Daten werden dann über das Http-Protokoll von dem CLM Server zu einem entfernten Datenempfänger gesendet.

Auf Seiten des Datenempfängers und Verwalters wurde ein Server aufgesetzt, welcher Anfragen annimmt, auswertet und in einer Datenbank speichert. Zudem wurde ein genaues Augenmerk auf personenbezogene Daten und auf die Datenschutzrechte in Deutschland gerichtet, damit diese Bachelorarbeit auch produktiv eingesetzt werden darf [3].

Anschließend wurden diese Daten systematisch ausgewertet und ein geeigneter Workflow dieses Systems abgeleitet. Mit diesem Stand

kann die Usability der Toolsuite analysiert und verstanden werden und somit in Zukunft die User Experience der Benutzer gesteigert werden.

Diese Bachelorarbeit beinhaltet einige Herausforderungen in der Analyse der Gegebenheiten, der Erarbeitung eines schlanken und zielgerichteten Konzeptes und der Einhaltung von Restriktionen im Datenschutz, wie auch Restriktionen durch die it-Infrastruktur eines weltweit agierenden Unternehmens. All diese Herausforderungen wurden jedoch letztendlich erkannt, einbezogen und gemeistert.

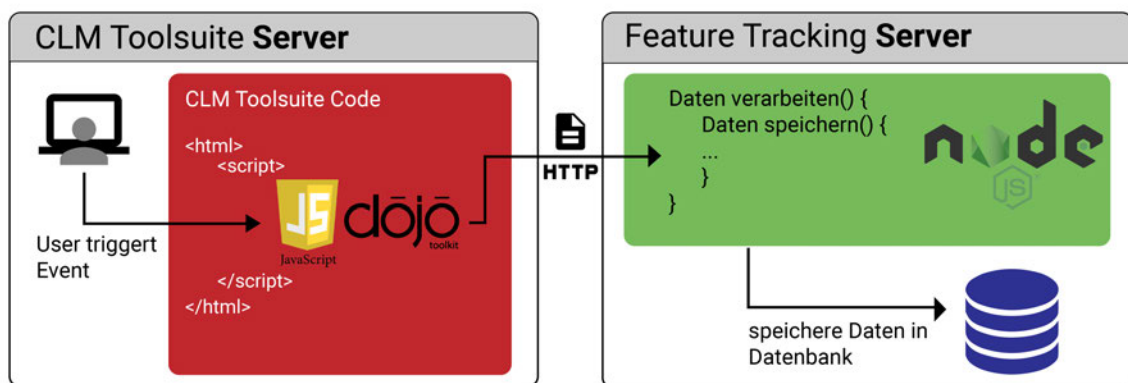


Abbildung 1: Gesamtsystem der Bachelorarbeit

-
- [1] User Experience Design, C. Moser, Springer Vieweg, pp. 13/186–190/224–230, Heidelberg, 2012
 [2] Web Analytics, Marco Hassler, Mitp, München, 2012
 [3] Bundesdatenschutzgesetz, betrieben vom Bundesbeauftragten für den Datenschutz und die Informationsfreiheit (BfDI), letzter Aktualisierung: 18.08.2012, <https://www.bfdi.bund.de>

Bildquellen:

- Abbildung 1: David Seddaoui – enthält offizielles Logo von JavaScript, Dojo Toolkit, Node.js und zwei Icons aus dem Google Material Design Katalog

Konzeption zur Leistungssteigerung von hochauflösenden Pixel-Scheinwerfern für ein Erprobungsfahrzeug einschließlich prototypischer Realisierung notwendiger Software-Komponenten in MATLAB Simulink

Sammy Traxler*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Statistiken belegen, dass die meisten tödlichen Verkehrsunfälle bei Dämmerung oder während der Nacht geschehen. Ein wesentlicher Faktor hierfür ist die ungenügende Ausleuchtung der Straße. Weitere Probleme stellen entgegenkommende Fahrzeuge dar, welche das Fernlicht zu spät oder gar nicht deaktivieren. Dadurch werden andere Verkehrsteilnehmer geblendet und können das Verkehrsgeschehen für kurze Zeit nicht mehr wahrnehmen. So kommt es oft zur sogenannten Blindsekunde, durch die betroffene Fahrer ihre Sicht vollständig verlieren können und dann nicht in der Lage sind, rechtzeitig auf Notsituationen zu reagieren.

Deshalb werden neuartige Lichtsysteme über einen Algorithmus angesteuert, der auf Basis von Kameradaten und weiteren Fahrzeugdaten der Situation angepasste Lichtverteilungen generiert. Befindet sich ein fahrendes Fahrzeug im Blendungsbereich, wird das Fernlicht mechanisch über eine Walze, die sich zwischen Lichtquelle und Linse befindet, an der entsprechenden Stelle abgedeckt. Allerdings stößt dieses System schnell an seine Grenzen, denn für den Fall, dass sich mehrere Verkehrsteilnehmer in unmittelbarer Nähe befinden, muss das Fernlicht komplett deaktiviert werden.

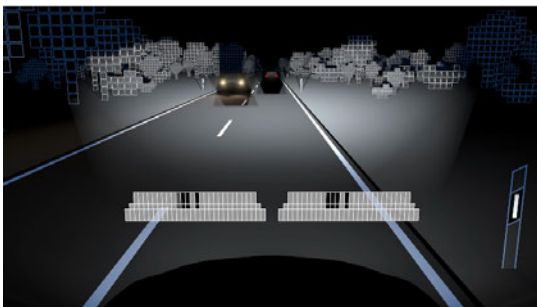


Abbildung 1: Aktuelle E-Klasse mit 84 LEDs pro Scheinwerfer

Aufgrund der neuesten Entwicklungen von Pixel-Scheinwerfern, die aus vielen einzelnen

LEDs bestehen, die einzeln elektronisch aktiviert, deaktiviert oder gedimmt werden können, gehören diese Probleme nun der Vergangenheit an. Fahrzeuge, die sich in Fahrtrichtung und in Gegenfahrtrichtung befinden, können frühzeitig und präzise ausgeblendet werden, ohne das Fernlicht komplett abzuschalten zu müssen und somit das Sichtfeld einzuschränken. Die aktuelle E-Klasse der Daimler AG verfügt über 84 LEDs pro Scheinwerfer.

Eine weitere dieser neuen Technologien, die sich noch in der Entwicklung befindet, trägt den Namen μ AFS. Dieser steht für micro-structured Adaptive Front-lighting System. Die Lichtquelle besteht aus einem kleinen Chip, auf dessen Fläche von 4mm x 4mm sich 1024 LEDs befinden. Das betreffende Verbund-Projekt wird vom Bundesministerium für Bildung und Forschung gefördert. Die beteiligten Kooperationspartner sind[1]:

- Osram Opto Semiconductors
- Osram GmbH
- Fraunhofer IZM
- Infineon Technologies AG
- Hella KGaA Hueck & Co
- Daimler AG

Drei dieser Chips befinden sich jeweils im linken und rechten Frontscheinwerfer eines Erprobungsträgers. Dadurch ergibt sich eine Gesamtanzahl von 6144 LEDs. Für die Berechnung der Lichtbilder wurde im Kofferraum des Fahrzeugs eine MicroAutoBox (MAB) installiert, auf der für die Ansteuerung der Scheinwerfer eine Software auf Basis eines MATLAB-Simulink-Modells läuft. Die MAB erhält über CAN alle relevanten Fahrzeug- und Kamerainformationen von den Steuergeräten des Fahrzeugs. Die errechneten und an die jeweilige Fahrsituation angepassten Lichtverteilungen werden dann ebenfalls über CAN an die Steuergeräte der Scheinwerfer gesendet.

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

Es hat sich jedoch herausgestellt, dass die verwendeten Komponenten, insbesondere das im Fahrzeug verwendete CAN, für eine effiziente Nutzung des Systems nicht über eine ausreichende Leistungsfähigkeit verfügen. Aus diesem Grund musste das Fahrzeug umgerüstet und die Software für die Ansteuerung der Scheinwerfer optimiert werden.

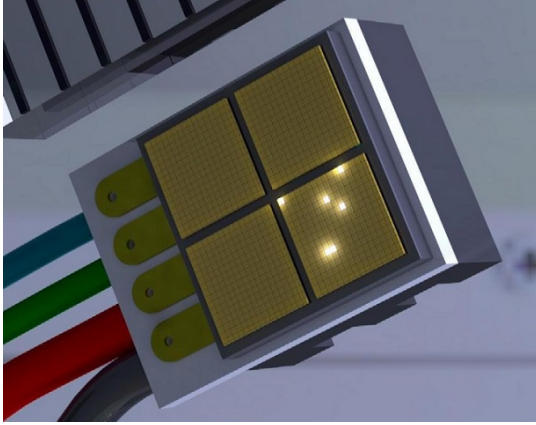


Abbildung 2: µAFS-Chip

Bedingt durch die große Anzahl an LEDs ergibt sich für die Ansteuerung der Lichtquel-

len ein erheblich größeres Datenvolumen als bei bisherigen Systemen. Das bisher verwendete Bussystem auf Basis CAN 2.0 hat sich für den Datentransfer dieser Größe als nicht ausreichend leistungsfähig erwiesen und musste daher durch CAN FD ersetzt werden. Ebenso war die Definition eines neuen Protokolls und dessen Integration in die Software für die Ansteuerung der Scheinwerfer erforderlich.

Dies allein ist jedoch nicht ausreichend. Damit für den Fahrer das schnelle und dynamische Wechseln der Lichtverteilungen als angenehm empfunden wird, müssen hohe Bildwiederholungsraten von 50–60Hz erreicht werden. Diese zusätzliche Anforderung machte eine Komprimierung der zu sendenden Daten unumgänglich. Die beiden in Frage kommenden Kompressionsverfahren waren die Lauf-längenkodierung und die Hoffman-Kodierung. Für das letztere Verfahren musste ein Algorithmus entwickelt werden, der in wenigen Millisekunden die Daten verlustfrei komprimiert.

Die Überarbeitung des Konzepts und die Optimierung der Ansteuerung werden einen effizienten Betrieb der neuartigen Scheinwerfer ermöglichen.

[1] http://www.photonikforschung.de/fileadmin/Verbundsteckbriefe/16_Integrierte%20Optik/mu-AFS_Projektsteckbrief-korr2012-04.pdf

Bildquellen:

- Abbildung 1: Daimler AG
- Abbildung 2: Osram OS

Machine-Learning-Verfahren zur Detektion von Mobilitätsmodi

Ricardo Zeich*, Andreas Rößler, Dirk Hesse

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Sommersemester 2016

Motivation

Viele Großstädte und Metropolregionen ächzen unter der Last des motorisierten Individualverkehrs. Nicht nur der immer dichter werdende Verkehr auf den Straßen, welcher lange Fahrzeiten und Lärm verursacht, sondern auch die Belastung durch z. B. Feinstaub machen ein Umdenken bei unserem Mobilitätsverhalten unausweichlich. Die Vorgehensweisen sind sehr verschieden. Ein sehr aktuelles Beispiel im Frühjahr 2016 ist die finanzielle Förderung des Kaufs von Elektrofahrzeugen für Endkunden vonseiten der Bundesregierung. Auch Car-sharing wird immer weiter ausgebaut. Zur Verringerung des Individualverkehrs, sowie zur Senkung der damit verbundenen Abgase wird auch auf die Ausweitung des öffentlichen Nahverkehrs oder das Angebot alternativer Transportmöglichkeiten gesetzt.

Um in Gegenden mit erschwerten Bedingungen, wie zum Beispiel dem Großraum Stuttgart, welcher durch die einzigartige Lage der Landeshauptstadt Stuttgart im „Talkessel“ nur begrenzte Möglichkeiten der Zu- und Abfahrt nach und von Stuttgart bietet, wird daher vermehrt auf eine durchdachte Mischung aus Individualverkehr verschiedenster Art und öffentlichem Personennahverkehr gesetzt. Die Mobilitätsbedürfnisse einzelner Menschen, als Teil einer großen Gesellschaft zu begreifen und sinnvoll in ein angemessenes Mobilitätsangebot zu gießen, stellt jedoch eine große Herausforderung dar.

Hier kommen Smartphones ins Spiel. Deren Verbreitung liegt in der Bundesrepublik mittlerweile bei 74% der Bevölkerung. Bei Menschen im Alter von 14 bis 49 Jahren beträgt der Smartphoneanteil gar 93%, bei 50- bis 64-Jährigen liegt dieser bei 83% [1]. Somit trägt ein Großteil der Bevölkerung, der täglich auf unterschiedlichste Weise in die Schule, Arbeit, Hochschule oder zu Freizeitaktivitäten unterwegs ist, ein Smartphone bei sich.

Da diese heutzutage auch in günstigeren Ausführungen sowohl angemessene Rechenleistung als auch weitreichende Sensorik mitbringen, tut sich hier ein ungeahnter Pool an wertvollen Mobilitätsdaten auf, der hervorragend geeignet scheint die Mobilitätsbedürfnisse einer modernen, nach Individualität stre-

benden aber gleichzeitig bewusster lebenden Gesellschaft zu evaluieren. Sensoren moderner Smartphones können von Luftdruck über Beschleunigung, Drehbewegungen, (Elektro-)magnetische Felder und Helligkeit bis hin zu GPS-Signalen eine schier unendliche Menge an Daten über die Umgebung liefern. Diese auszuwerten, um das Mobilitätsverhalten, konkreter: die Art der Mobilität herauszufinden, ist eine komplexe Aufgabe. Generell werden diese Mobilitätsdaten aufbereitet, indem man sogenannte „Features“ definiert und aus den Daten ableitet. Angewendet auf die Daten des Beschleunigungssensors können zum Beispiel Durchschnitt, Korrelation verschiedener Achsen zueinander, das Rauschen der verschiedenen Achsen, Stärke sowie Frequenz von Signalen als auch Minima und Maxima ein „Feature“ sein [2].

Um die verschiedenen Arten von Mobilität, wie zu Fuß gehen, Fahrrad, Auto, Bus und Bahn fahren möglichst präzise voneinander unterscheiden zu können, wird eine große Anzahl der genannten Features betrachtet. Dazu wird eine Aufzeichnung des Mobilitätsverhaltens über einen gewissen Zeitraum analysiert, was zu großen Datenmengen führt. Das System sollte eine hohe Präzision und Robustheit aufweisen, um in möglichst vielen Anwendungsbereichen korrekte Ergebnisse zu liefern. Ziel dieser Arbeit ist es, zu untersuchen, ob Machine-Learning-Methoden, welche diese Eigenschaften aufweisen, geeignet sind, um dieses Problem zu lösen. Methoden, die einmal eingelernt beständig dazulernen und sich an neue Gegebenheiten anpassen, bieten sich für diese Zwecke an.

Machine-Learning

Maschinelles Lernen ist mit dem Lernen, wie wir Menschen es tun, vergleichbar. Es strebt danach, dieses zu erreichen und dessen Vorteile zu nutzen. Anhand von Beispielen werden Zusammenhänge erlernt und so Erfahrungen angesammelt, auf welche bei zukünftigen Entscheidungen zurückgegriffen wird. Im Vergleich zur traditionellen Programmierung von Maschinen wird hier aber auf selbstständige Erkennung von Mustern, statt auf feste Vorschriften gesetzt [3]. So kann auch auf Basis

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

von völlig neuen Daten eine angemessene Entscheidung gefällt werden. Grundlage für die Entscheidungsfindung können unter anderem folgende Vorgehensweisen sein.

Entscheidungsbäume

Entscheidungsbäume stellen in Baumstruktur abgelegte Informationen dar. Von einem Wurzelknoten aus wird an jedem Knoten ein Feature betrachtet und eine Entscheidung zur Klassifikation gefällt. Von einem Knoten gelangt man so entweder zu einem weiteren Knoten, an welchem nochmals ein Feature betrachtet wird, oder man erreicht ein Blatt, was einer Entscheidung bzw. Einstufung entspricht. Einen solchen Baum lernt man ein, indem Beispieldaten mit korrektem „Label“ als Eingruppierung genutzt werden. Um viele verschiedene Merkmale zu betrachten, kann das „Random-Forest“-Verfahren, bestehend aus mehreren unabhängigen Bäumen, angewandt werden.

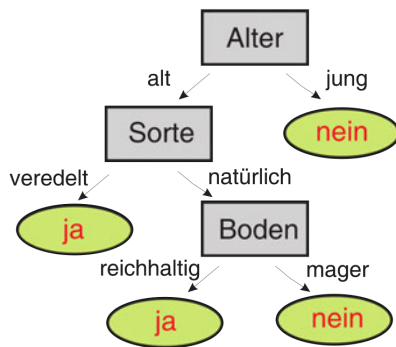


Abbildung 1: Ein einfacher binärer Entscheidungsbaum zur Entscheidung, ob ein Baum Früchte tragen wird.

k-Nearest-Neighbor-Algorithmus

Hier erfolgt die Beurteilung eines Datenpunktes anhand seiner k nächsten Nachbarnpunkte. Die Nachbarnpunkte beeinflussen insofern die Klassifikation, dass in der Regel die Klassifi-

kation der Mehrheit der k Nachbarnpunkte die Klassifikation des fraglichen Punktes bestimmt [4]. Dieses Verfahren wird auch „Nächste-Nachbarn-Klassifikation“ genannt.

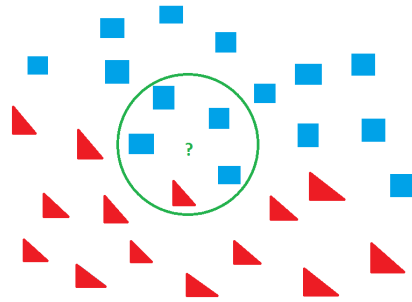


Abbildung 2: Der an der Stelle des Fragezeichens befindliche Punkt würde anhand seiner fünf nächsten Nachbarn als blaues Rechteck klassifiziert.

Neuronale Netze

Das menschliche Gehirn ist ungeheuer leistungsfähig. Es besteht aus unzähligen parallel arbeitenden Nervenzellen (Neuronen), welche lernfähig sind. So können wir als Menschen aus Beispielen lernen und diese dann auf ähnliche Situationen übertragen. Computer stehen dazu im Gegensatz, da sie aus Transistoren aufgebaut sind und vorher genau festgelegte Abläufe großteils seriell arbeiten. Eine technische Realisierung des biologischen Gehirns ist bereits seit Mitte des 20. Jahrhunderts Forschungsgegenstand. Ein solches künstliches neuronales Netz besteht aus nachgebauten Neuronen, welche die Ausgangssignale vieler weiterer als Eingangsvektor nutzen und diese Werte nichtlinear zu ihrem eigenen Ausgangswert verarbeiten. So wird stark vereinfacht ein künstliches Gehirn geschaffen, das sich, ähnlich dem menschlichen, einlernen lässt [5].

- [1] <https://www.bitkom.org/Presse/Presseinformation/Smartphones-sind-fuer-viele-Senioren-unverzichtbar.html> (Aufgerufen im Mai 2016)
- [2] Identifying active travel behaviors in challenging environments using GPS, accelerometers, and machine learning algorithms – Katherine Ellis, Suneeta Godbole, Simon Marshall, Gert Lanckriet, John Staudenmayer and Jacqueline Kerr.
- [3] Machine Learning in Healthcare Informatics – Sumeet Dua, U. Rajendra Acharya, Prerna Dua.
- [4] IKNN: Informative K-Nearest Neighbor Pattern Classification – Yang Song, Jian Huang, Ding Zhou, Hongyuan Zha, and C. Lee Giles.
- [5] Ein kleiner Überblick über Neuronale Netze – David Kriesel – erhältlich auf <http://www.dkriesel.com>

Bildquellen:

- Abbildung 1: <https://de.wikipedia.org/wiki/Entscheidungsbaum#/media/File:Entscheidungsbaum.svg>
- Abbildung 2: Eigenes Werk

