



Informationstechnik

Hochschule Esslingen
University of Applied Sciences

IT-Innovationen

Band 18
Januar 2017

Grußwort des Dekans

Liebe Leserinnen und Leser,

„Technik, die wir vor noch drei Jahren in weiter Ferne sahen, steht uns bereits heute zur Verfügung. Das autonome Fahren wird schneller kommen als wir damals dachten und manch einer von uns heute noch vermutet“, so der Entwicklungsleiter einer unserer Industriepartner.



Maßgeblich beigetragen zu dieser rasanten Entwicklung hat die Informationstechnik. Sie ist Treiber und hält in den klassischen Ingenieursdisziplinen rasant Einzug in das Curriculum. Allerorten spricht man von „Digitalisierung der Lehrinhalte“. Aber auch die Informationstechnik selbst stellt sich permanent den neuen Herausforderungen. Vernetzung, Maschinelles Lernen, IT-Sicherheit sind nicht nur für autonome Fahrzeuge von enormer Relevanz. Der Wunsch aus der Industrie nach zusätzlichen theoretischen Kenntnissen in der Kerninformatik wird immer häufiger an uns herangetragen. Der Start unseres Masters in Angewandter Informatik ist nur eine Antwort darauf. Auch in unseren Bachelorabschlüssen ziehen wir im Curriculum die Innovationen in der Informationstechnik permanent nach, um auch weiterhin dem exzellenten Ruf unserer Hochschule und deren Absolventen gerecht zu werden.

Die Ihnen vorliegende Ausgabe der IT-Innovationen zeigt wieder in eindrucksvoller Weise, aus welchem breitem Themenspektrum die Aufgabenstellungen der Abschlussarbeiten schöpfen und mit welcher Technologievielfalt die Industrie unsere Absolventen betraut. Überzeugen Sie sich selbst vom hohen Stand des Wissens, den Fertigkeiten und dem Können unserer IT-Absolventen.

Bleibende Eindrücke beim Lesen 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

Stavros Avramidis	Machbarkeitsstudie für Schnellprüfung der Katalysatordiagnose mittels Katalysatormodell	1
Alexander Barabanov	Research on Adaptive Aggregation Methods of Cooperative Awareness Messages in Car-to-X Communication Networks	3
Dennis Bauer	Sicherer Zugriff auf Intranet-Webseite aus einem externen Netzwerk durch Einsatz eines HTTP-Proxies	5
Jan Baumann	Entwicklung eines spezifischen Benchmarks zur Performance-Analyse neuer Rechnerstrukturen für Motorsport-Steuergeräte mittels aus MATLAB und Simulink generiertem Quellcode	8
Tobias Deißler	Visualisierung elektrischer Ströme mit Android	10
Jochen Eberle	Konzeption und Entwicklung eines Tools zur Automatisierung des Prozesses der Testdatenidentifizierung in globalen SAP Projekten	12
Rasty Faraj	Benefit of Quality	15
Oliver Feucht	Analyse und prototypische Realisierung einer Android App zu Verbindung des Connected Car mit anderen Domänen	17
Lennard Friedrich	Usability Engineering – Prototypische Rekonstruktion der Benutzeroberfläche einer Business Anwendung	19
Philip Furtwängler	Der Begriff der Modularisierung vor dem Hintergrund der Geschichte der Entwurfsprinzipien	21
Thomas Greule	Best Practices beim Übergang eines Unternehmens von einer traditionellen Entwicklungsmethode in eine agile Entwicklungsmethode	23
Ricardo Grilo	Realisierung einer Schnittstelle zwischen einem FPGA-basierten Sercos-Slave und der Arduino-Plattform einschließlich der zugehörigen API	25
Dennis Groß	Big Data Analytics auf Fahrzeugdaten	27
Christoph Gschrey	Die Konzepte „Dependency Look-Up“ und „Dependency Injection“	29
Timo Manuel Gutierrez	Entwicklung eines webbasierten Shopfloor Management Modul innerhalb der Planungs- und Steuerungssoftware 3Liter-PPS®	31
Felix Heil	Konzipierung und Realisierung eines Analysewerkzeugs für das Monitoring von Telegrammen der synchron-seriellen Schnittstelle SSI im Besonderen unter Aspekten der Funktionssicherheit für Positioniersysteme der Automatisierungstechnik	33
Patrice Heller	Konzeption, Realisierung und Qualifizierung eines Moduls zur Nachbildung des Gleichspannungszwischenkreises von Elektrofahrzeugen unter Verwendung der Curve Fitting Toolbox von Matlab	36
Oliver Hummel	Agiles Testen	38
Tobias Kiechle	Entwicklung einer Architektur für eine Definition von Bedienoberflächen für Touchdisplay und Implementierung einer Anwendung	40
Christoph Kreibich	Optimierung von Datenbeladungsstrecken und Datenmodellen durch Nutzung von In-Memory Technologien	42

Richard Krieger	Prototypische Umsetzung einer Augmented Reality Applikation zur Markierung von Objekten	44
Sven Lipiz Radalga	Ansätze zur Reduzierung von Installationspaketen auf der mobilen Plattform Android	47
Sebastian Lobsinger	Prototypischer Entwurf und Implementierung eines Vehicle-to-X Protokollstacks gemäß dem Standard WAVE einschließlich einer Beispielapplikation unter Verwendung des Nachrichtensets nach SAE J2735 in einer AUTOSAR-Umgebung	50
Kay Müller	Programmierung eines Rapid-Prototyping Universalboards und einer GUI für die Inbetriebnahme und den Betrieb unterschiedlicher DCDC-Wandler Topologien	52
Benjamin Munzinger	Prozessuale Einführung von Kanban (IT) in einem nach Scrum agierenden Team in Hinsicht auf Continuous Delivery	54
Konstantinos Papadopoulos	Entwicklung und Umsetzung eines Konzepts zum Data Management von referenz- und applikationsspezifischen Daten für iDome-Applikationen	57
Christian Rautenberg	Untersuchung und Anwendung von Algorithmen in Matlab zur Erkennung von Verkehrsschildern für das autonome Fahren	59
Alessandro Valentino Ribul Moro	„Favour Composition over Inheritance“	61
Chris Schäfer	Analyse und Lösung von OpenAI Environments mit Hilfe von Künstlichen Neuronalen Netzen und Reinforcement Learning	64
Ulf Schmelzer	Konzipierung und Realisierung eines Test-Access-Points für die PC-basierte Diagnose industrieller Kommunikationssysteme auf Basis eines FPGAs und Embedded-Linux	67
Benedikt Julian Schrade	Gamification zur Verbesserung der Prozess- und Datenqualität in Unternehmen	69
Christoph Stäbler	Informationsextraktion durch Methoden des maschinellen Lernens und Anwendung auf einen ETL-Prozess für Fahrzeugdaten	71
Waldemar Stenski	IT Prozess Stabilisierung & Optimierung für das MFTBC SAP System im Finanz & Controlling Umfeld	73
Georgios Terzoglou	Analyse und prototypische Implementierung von Schrittzähleralgorithmen mit einem MEMS Inertialsystem bestehend aus Beschleunigungs- und Drehratensensor	75
Marcel Trinks	Entwicklung eines verteilten Datenbanksystems mit Webfrontend für die Verwaltung und Analyse von Diagnosedaten im Automobilbereich	77
Konstantin Wacker	Konzipierung und Realisierung eines Plug-Ins als Schnittstelle zwischen einem Nachtfahringsimulator und dem Ansteuerungsalgorithmus eines adaptiven hochauflösenden Multipixel-Scheinwerfersystems für den Einsatz in zukünftigen Fahrzeuggenerationen	79
Patrick Wieland	Entwurf, Simulation und Implementierung von Hard- und Software eines Flugreglers für das Versuchs-Segelflugzeug Discus 2C des DLR	81
Hatice Yildirim	Imitation of Life – Die Autonomie moderner UI-Konzepte vom Skeuomorphismus zum Flat Design	83
Marcel Zeller	Automatisierte Sicherheitsanalyse von Datenverkehr mit Intrusion Detection in proprietären Protokollen	86

Machbarkeitsstudie für Schnellprüfung der Katalysatordiagnose mittels Katalysatormodell

Stavros Avramidis*, Kai Warendorf, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Parallel zur stetigen Verschärfung der Emissionsgesetzgebung, erhöhen sich auch die Anforderungen der On-Board-Diagnose (OBD) kontinuierlich weiter. Zur Einhaltung der Emissionsgrenzwerte reichen motorische Maßnahmen allein nicht mehr aus. Neben der Reduzierung von Kraftstoffverbrauch und Partikelemissionen durch Direkteinspritzung und Abgasrückführung, ist beim Ottomotor eine katalytische Abgasnachbehandlung erforderlich. Dabei hat sich der Drei-Wege-Katalysator mit Lambdaeule als leistungsfähigstes Konzept in Hinblick auf Schadstoffreduzierung für Ottomotoren erwiesen. Um die gesetzlich geforderten Emissions-OBD-Schwellen einzuhalten und das Bauteil auf Schäden überprüfen zu können, wird mittels OBD der Zustand des Katalysators im Betrieb überwacht.

Aktuelle Serienfahrzeuge sind mit Diagnoseverfahren ausgerüstet, welche direkt oder indirekt auf der Analyse der Sauerstoffspeicherfähigkeit eines Drei-Wege-Katalysators basieren. Dieses Konzept bringt aufgrund der Temperaturabhängigkeit der Sauerstoffspeicherfähigkeit des Katalysators einen großen Nachteil mit sich. Je nach Fahrzeug kann so die Diagnose nur in einem Temperaturfenster von 500 bis ca. 800°C Katalysatortemperatur ablaufen, was durch neue abgastemperaturärere Motorkonzepte bereits problematisch ist. Darüber hinaus ist die Aussage über die Konvertierungsleistung durch die Sauerstoffspeicherfähigkeit kritisch zu betrachten, da die Sauerstoffspeicherfähigkeit unter bestimmten Umständen nicht die volle Konvertierungsleistung des Katalysators abbildet. Aufgrund dessen hat die Daimler AG ein neuartiges modellbasiertes Verfahren zur Katalysatordiagnose entwickelt. Hierbei wird das bisherige Maß für die Güte des Drei-Wege-Katalysators durch einen SC-Wert abgelöst, um den Alterungszustand temperaturunabhängig und kontinuierlich zu bewerten. Die Storage Capacity „SC“ gibt an, wieviel konvertierungsfähiges Material noch im Katalysator vorhanden ist. Dabei

wird die Konvertierungsleistung des Katalysators vollständig abgebildet. Der Einsatz dieser Diagnose bringt eine Kraftstoffersparnis mit sich, da keine aktive Lambdaanforderung mehr benötigt wird. Simultan reduzieren sich dabei auch die Emissionen.

Im Rahmen dieser Arbeit soll die modellbasierte Katalysatordiagnose auf eine von der Daimler AG neuen Motorsteuerung „MRG 1“ angepasst werden. Hierzu ist eine Voruntersuchung der alten Motorsteuerung „MED 17.7“ notwendig. Die Implementierung folgt dabei in MATLAB. Das Katalysatormodell besteht aus den drei Teilmodellen, Rohemissionsmodell, Drei-Wege-Kat-Modell und Lambda-Sprungsonden-Modell. Dabei beschreibt es die wichtigsten auf der Katalysatoroberfläche ablaufenden Reaktionen abhängig von SC. Aufgrund der Tatsache, dass SC nicht direkt gemessen werden kann, wird das Diagnosemodell um einen Sigma-Punkt-Kalman-Filter erweitert, welcher als Beobachter die modellierte Nachkatsondenspannung mit der vom Motorsteuergerät gemessenen Nachkatsondenspannung vergleicht und den SC-Wert schätzt [1]. Diese Schätzung dient, als Ausgang des Katalysatormodells, zur Diagnose des Alterungszustands eines Drei-Wege-Katalysators (siehe Abbildung 1 und 2).

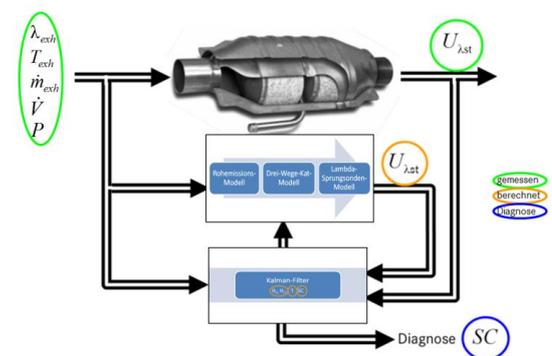


Abbildung 1: Gesamtmodell mit Beobachter für Katalysatordiagnose

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

Der Einsatz einer neuen Motorsteuerung, einer neuartigen Beschichtungsvariante des Katalysators sowie einer neuen Lambdasonde bedarf einer einmaligen Neukalibrierung der Modellparameter. Hierzu werden Messungen auf einem Motor- und Synthesegasprüfstand durchgeführt, durch diese verschiedene Gaskonzentrationen sowie SONDENSPIEGEL ermittelt werden können. Mit den ermittelten Messergebnissen und den dabei gewonnenen Erkenntnissen kann so das Modell in MATLAB neu kalibriert werden.

Angesichts zunehmender Hybridisierung verkürzt sich die Motorlaufzeit der Verbrennungsmotoren deutlich, sodass je nach Fahrprofil verschiedene Diagnosefunktionen im Motorsteuergerät, die den Verbrennungsmotor in eingeschaltetem Zustand benötigen, deutlich weniger Zeit haben abzulaufen. Dies kann zur Folge haben, dass die vom Gesetzgeber geforderte Mindestablaufhäufigkeit der Diagnosen (IUMPR) nicht mehr bzw. immer schwerer erreicht wird. Im Fall der Katalysatordi-

agnose ist das Ergebnis der IUMPR erst gültig, wenn auch die Diagnosen der Lambdasonden gut geprüft wurden. Für die Prüfung werden über eine aktive Lambdaverstellung definierte Lambdasprünge gebildet, welche zwecks Diagnostizierung sowohl für die Lambdasondiagnosen als auch für die aktive Katalysatordiagnose Verwendung finden. Für den Fall der modellbasierten Katalysatordiagnose, welche passiv zum Einsatz kommt, ist zu prüfen, ob und in welchem Rahmen die ursprüngliche Lambdaverstellung noch benötigt wird. Die Möglichkeit, die aktive Lambdaverstellung zu vermeiden oder zu reduzieren, zieht eine Anpassung der Diagnosefunktionen nach sich. In diesem Zusammenhang soll zuletzt untersucht werden, inwieweit die modellbasierte Katalysatordiagnose als Schnellprüfung eingesetzt werden kann und welche Lambdasondiagnosen angepasst werden müssen, um eine Schnellprüfung des Katalysators sowie der Sondendiagnosen weitestmöglich gewährleisten zu können (Quick-Pass-Strategie).

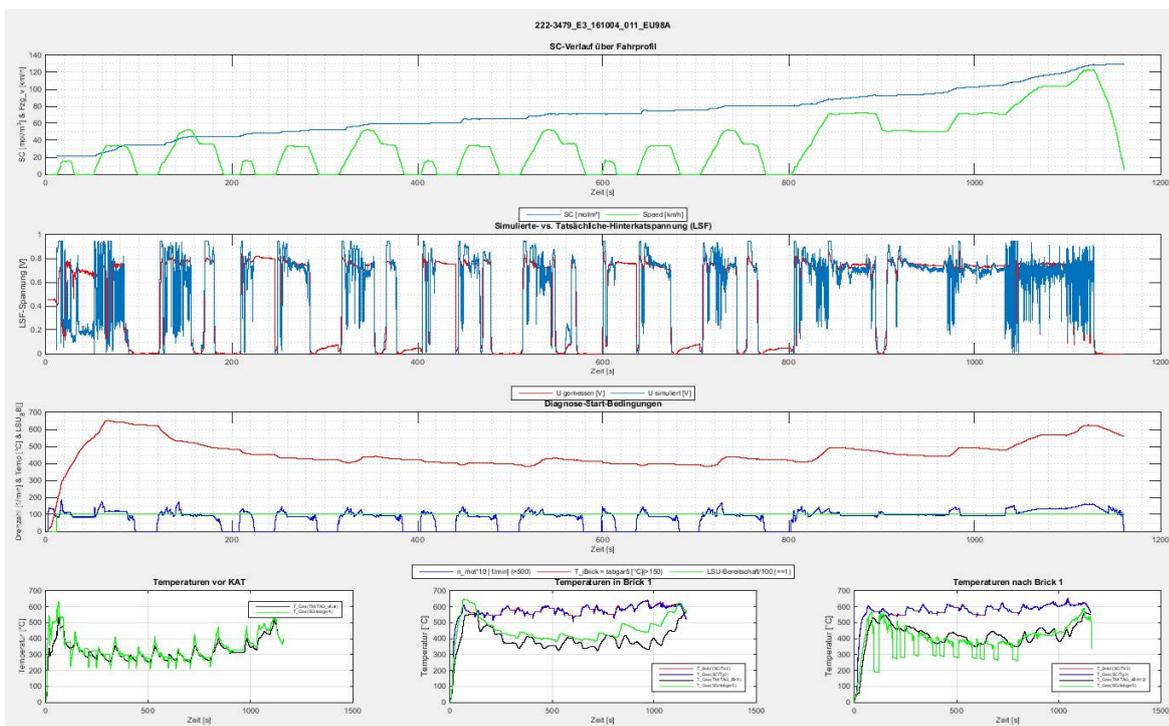


Abbildung 2: Ausschnitt Matlab-Plot

[1] J. Konrad, N. Ricker, I. Malisi: TWC System – Diagnosekonzept. Sindelfingen: Daimler AG, April 2013

Bildquellen:

- Abbildung 1: Daimler AG
- Abbildung 2: Eigene Darstellung in Matlab

Research on Adaptive Aggregation Methods of Cooperative Awareness Messages in Car-to-X Communication Networks

Alexander Barabanov*, Reinhard Keller, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Direct communication between vehicles and the surrounding infrastructure not only provides a broad number of advantages, e.g., safety, comfort and infotainment of the traffic participants, but also brings a number of challenges, among which is high data traffic in frequently changing Vehicular Ad-Hoc Networks (VANET), based on the ETSI ITS-G5 standard [1] – the dominant VANET standard in Europe. Cooperative Awareness Messages (CAM) [2] are defined as containers for general information about the vehicle, such as position, velocity and size, as well as specific information, such as type (family car, truck, bicycle) or role (ambulance, police). All traffic participants transmit up to 10 CAMs per second, depending on the traffic dynamic.

In the previously proposed framework [3], infrastructure units (Roadside units, RSU), located alongside the road, receive CAMs in their operational range and forward, i.e., multi-hop the received data over several RSUs to a processing application on a control RSU (C-RSU). The application then evaluates the information from all RSUs and warns the traffic participants upon potential dangers in real time.

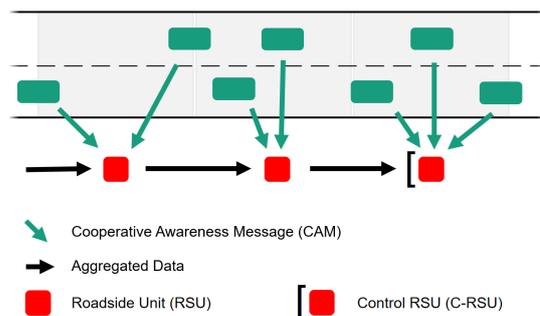


Figure 1: Multi-hop of the Vehicular Data

In order to provide a better control over the dataflow and reduce the risk of congestion on the shared wireless channel, a data aggregation scheme has been introduced [4]. The scheme uses a tree-based data structure to group the traffic participants by predefined criteria, allowing switching between aggrega-

tion levels, thus adjusting the accuracy and the size of the transmitted data.

The next challenge in increasing the efficiency of the data aggregation scheme is to automate the grouping criteria parameters optimization, introducing adaptive aggregation metrics and processing the received data according to those metrics.

Simply speaking, the challenge can be broken down to mimicry of human conversational behavior depending on the conditions: if the speaker is asked to describe a road situation in full details, the description might look like an essay. The less time the speaker has available, the more details are going to be left out. Once the speaker is only given a second to respond, or, in terms of the topic, the wireless channel gets to the point of the peak load, the description might come down to a few words, such as “all clear” or “there is a reckless driver”.

This thesis presents a research on the ways to reproduce a similar filtering and aggregation mechanisms with an ability to autonomously adapt to the changing environmental settings.

Data aggregation describes the process of combining various data records, e.g., of vehicles in a VANET. The proposed aggregation process can be generally divided into three phases: decision phase, where the aggregation scheme identifies and groups data records to fuse; fusion phase, where each group is merged into a single data record; dissemination phase, where the merged data is forwarded towards a data sink.

During the decision phase, data records are identified for fusion and grouped by certain criteria. The grouping strategies, proposed in SOTIS and CASCADE, are to split the road into segments. The strategy of TrafficView uses a cost function to identify the two items with the least fusion cost. The function takes the distance between the vehicles and the number of vehicles represented by a data record into account.

Fusing data by one certain metric, e.g., segment of the road or distance between traffic

*Diese Arbeit wurde durchgeführt bei der Firma Fraunhofer Institute for Embedded Systems and Communication Technologies ESK, Munich

participants, might lead to information loss issues: during an overtake two or more vehicles might be close to each other, yet have a highly different dynamic. On the other hand, several vehicles might have similar velocity and acceleration, yet be uniformly distributed along the road, making a velocity-based aggregation less efficient in terms of the information loss.

Extreme values, e.g., an unusually fast or slow vehicle, are the key in safety threat recognition. Therefore, an efficient set of metrics has to fuse as many similar data records as possible, leaving the anomalies as single data records.

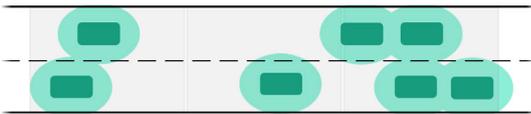


Figure 2: Decreased Tolerance Level in a Distance Metric



Figure 3: Increased Tolerance Level in a Distance Metric

As a solution, a set of adjustable metrics has been proposed. The set mainly includes position-based metrics (distance, lane, heading) and dynamic-based metrics (velocity, acceleration). Depending on the aggregation level required, the metrics group data records by the corresponding data fields with a given tolerance, i.e., parameters. For example, figures 2 and 3 show a decreased and an increased tolerance level in a distance metric, forming 3 and 2 groups accordingly. Likewise, the lane metric groups vehicles, located in the same lane on the lowest tolerance level, or within a certain number of adjacent lanes on higher tolerance levels. As the names suggest, the heading, velocity and acceleration metrics

group vehicles depending on their heading angle, velocity and acceleration values accordingly.

Each metric has an assigned weight that can vary depending on the traffic situation. Altogether, the metrics and their weights provide the decision component with a polynomial grouping function, where each metric is represented by one or several terms.

The proposed autonomous parameter adjustment mechanism (APA) is another step into further increasing the flexibility of the framework. Granted, the framework is provided with a number of data aggregation metrics, the APA evaluates the current road and network conditions and alters the set of metrics' parameters and weights to get the best fitting values.

Since the current road conditions can barely be predicted nor described formally, the evaluation component here is based on a machine learning scheme. The scheme is analyzing a snapshot of the traffic, searching for a set of aggregation metrics' parameters that will fit the current network congestion level, i.e., maximize the packet delivery ratio and minimize the delay, and for a set of metrics' weights that will produce the least error caused by data fusion. Apart from that, the evaluation also considers feedback from the next RSU, as well as from the next C-RSU. While the next RSU provides the evaluation component with local information, e.g., current congestion and aggregation levels, the C-RSU provides it with application, i.e., global scope data, thus aligning the state of the whole topology.

As the result, the proposed data aggregation metrics combined with the autonomous parameter adjustment mechanism provide a flexible way to process and evaluate the given data in order to reduce the information loss due to data fusion. Instead of using a predefined set of intervals, the metrics at hand fuse data records depending on their overall pattern, providing the decision making component with an adaptive fusion cost function.

-
- [1] ETSI EN 302 663 V1.2.1 – Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band, 2013.
 [2] ETSI EN 302 637-2 V1.3.2 – Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, 2014.
 [3] Jiru, J.; Bremer, L.; Graffi, K.: Data Aggregation in VANETs – A Generalized Framework for Channel Load Adaptive Schemes. The 39th IEEE Conference on Local Computer Networks (LCN), Edmonton, Canada, 2014.
 [4] Jiru, J.; Mammu, A.; Roscher, K.: Adaptive decision algorithms for data aggregation in VANETs with defined channel load limits. Submitted for 2015 IEEE Intelligent Vehicles Symposium (IV2015), Seoul, Korea, 2015.

Bildquellen:

- Abbildung 1-3: Eigene Darstellung

Sicherer Zugriff auf Intranet-Webseite aus einem externen Netzwerk durch Einsatz eines HTTP-Proxies

Dennis Bauer*, Manfred Dausmann, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

In jeder Firma gibt es Daten, die den Angestellten zur Verfügung gestellt werden, um ihre Aufgaben erledigen zu können. Dies ist auf unterschiedliche Art und Weisen möglich. Neben dem Bereitstellen der Daten auf einem zentralen Laufwerk, auf das von jedem Computer der Firma zugegriffen werden kann, ist es auch möglich, diese auf einer Webseite anzubieten. Die Webserver, auf denen diese Seiten liegen, sind meist nur innerhalb des Firmennetzes zu erreichen. So kann sichergestellt werden, dass nur Personen, die in der Firma arbeiten, auch Zugriff darauf haben.

Problemstellung

Allerdings gibt es auch Situationen, in denen ein Mitarbeiter auf Firmenreise, bei einem Kunden oder aus einem anderen Grund nicht in der Firma selbst ist. Dennoch muss er etwas recherchieren oder Daten auf einem Unternehmensserver eintragen. Diese sind aber nicht außerhalb des Netzwerkes verfügbar. In Abbildung 1 wird dieses Problem beispielhaft dargestellt. Der Laptop des Mitarbeiters ist hier nur symbolisch und kann selbstverständlich auch ein Desktop PC sein. Um trotz der Firewall auf die gewünschte Webseite zu gelangen, gibt es mehrere Möglichkeiten.

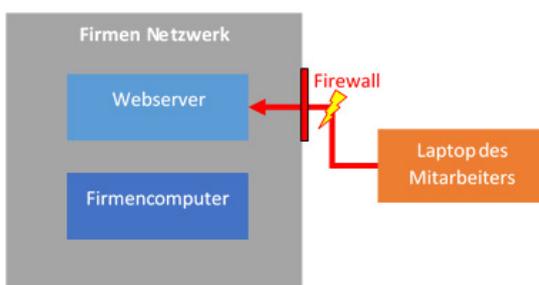


Abbildung 1: Direkte Verbindung zum Webserver

Zum einen kann eine Verbindung, über die Software TeamViewer, zu einem eingeschalteten Computer innerhalb der Firma hergestellt werden. Per Remote Control (Fernsteuerung) kann anschließend ein Browser geöffnet, die Webseite aufgerufen und die entsprechende Aufgabe erledigt werden. Dazu ist jedoch

ein dauerhaft eingeschalteter Firmencomputer notwendig. Zudem muss für diese Lösung jeder Mitarbeiter einen eigenen, eingeschalteten PC besitzen, auf den er jederzeit zugreifen kann. Die übertragenen Daten sind bei dieser Art des Zugriffs größer als die reine Webseite, da der Bildschirminhalt als Bild an den Laptop des Mitarbeiters geschickt wird. Auf Grund dieser Tatsache leidet zudem die Qualität der Webseite, da diese nicht auf dem Endgerät angezeigt, sondern nur ein Bild davon dargestellt wird. Beim Lesen von Text kann dies ebenfalls störend sein, da die Qualität des Schriftbildes deutlich schlechter ist. Zudem kommt es zu Verzögerungen beim Scrollen, da der neue Bildschirminhalt zunächst erneut übertragen werden muss.

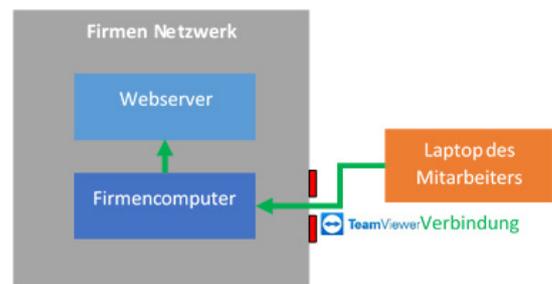


Abbildung 2: Verbindung zu Webserver über TeamViewer

In dieser Bachelorarbeit bei der TeamViewer GmbH wurde versucht, für diesen Anwendungsfall eine bessere Lösung zu finden und gleichzeitig die Bedienung zu vereinfachen. Die Idee war, die Webseite direkt auf dem Laptop des Mitarbeiters darzustellen und nicht auf einem Firmencomputer. Dieser sollte nur ein Gateway zum Erreichen des Webserver sein, um diese anschließend im Internet Explorer darzustellen. Der Vorteil des Internet Explorer ist es, dass er zum einen auf jedem Windows Computer vorinstalliert ist und keine separate Software heruntergeladen werden muss. Zum anderen aber auch, weil er das Einbinden in ein eigenes Fenster erlaubt, was für diesen Anwendungsfall vorgesehen ist. Für die Verbindung selbst soll TeamViewer einen Tunnel zwischen dem Laptop und dem Computer inner-

*Diese Arbeit wurde durchgeführt bei der Firma TeamViewer GmbH, Stuttgart-Vaihingen

halb der Firma bereitstellen und so die Firewall überwinden. Von diesem Computer aus kann der Webserver ohne Probleme erreicht werden (siehe Abbildung 2). Die TeamViewer Verbindung garantiert außerdem die Verschlüsselung jeglichen Datenverkehrs zwischen den beiden Computern.

Lösungsansätze

Zur Umsetzung der Idee wurden mehrere Möglichkeiten untersucht. Die erste Variante war der Einsatz eines SOCKS-Proxies [1], welcher für die Weiterleitung der Daten vom Webserver zu dem entfernten Computer und umgekehrt verantwortlich wäre. Allerdings gab es dabei auch einige Nachteile:

1. Die DNS Namensauflösung findet bei diesem Proxy auf Client Seite statt. Das bedeutet auf dem Laptop des Mitarbeiters außerhalb des Netzwerkes. Da die internen DNS Server von dort ebenfalls nicht erreicht werden können, kann auch der Name des internen Webserver nicht aufgelöst werden und es kann keine Verbindung hergestellt werden.
2. Der SOCKS-Proxy wird von Microsoft im Internet Explorer nicht mehr aktiv unterstützt.

Eine Alternative ist der HTTP-Proxy. Dieser hat dieselben Funktionen wie der SOCKS-Proxy, DNS Namensauflösungen können allerdings auf Serverseite, also auf dem Firmencomputer, der als Gateway fungiert, durchgeführt werden. Auf diese Weise werden die DNS Server der Firma verwendet und die URL wird korrekt aufgelöst. Des Weiteren wird der HTTP-Proxy im Internet Explorer noch aktiv von Microsoft unterstützt.

Lösungsalternative

Eine VPN Verbindung, genauer definiert von Paul Ferguson [2], zu dem Firmencomputer ist die dritte Möglichkeit um den Webserver zu erreichen. Auch bei dieser Art der Verbindung erfolgt die DNS Auflösung auf Serverseite. Somit ist auch hier eine Verbindung mit internen DNS Namen möglich. Allerdings hat VPN auch den Nachteil, dass sämtlicher Datenverkehr eines Computers über diese Verbindung gesendet und empfangen wird, da diese systemweit aufgebaut wird. Dies bedeutet zum einen, dass mehr Daten über die Verbindung und das Firmennetzwerk gesendet bzw. empfangen werden, zum anderen aber auch, dass Daten übertragen werden, die nicht zwingend über diese Verbindung gesendet werden müssten. Zudem kann es auch sein, dass die Firewall in der Firma bestimmte Seiten sperrt, die dann auch auf dem Computer des Mitarbeiters außerhalb des Netzwerkes nicht mehr erreicht werden können.

Realisierung

Eine Proxy Verbindung bietet den Vorteil, dass dieser nicht zwingend global definiert ist und sich für eine einzelne Browsersitzung einstellen lässt. Somit hat der Nutzer die Möglichkeit in einer anderen Browsersitzung weiterhin alle Webseiten direkt zu erreichen. Ein weiterer Vorteil ist ebenfalls, dass das Firmennetzwerk selbst besser geschützt ist, da Programme, die auf dem Computer aktiv sind, keinen Zugriff darauf haben. Aus den oben genannten Gründen wurde für einen HTTP-Proxy entschieden. Da bei dieser Lösung keine Verbindung zu einem Computer hergestellt und dort ein Browser geöffnet werden muss, um die Webseite darzustellen, können über den HTTP-Proxy mehrere Mitarbeiter gleichzeitig den Computer als Gateway nutzen und unterschiedliche Webseiten anfordern.

Neben der Einschränkung der Verbindung nur innerhalb einer bestimmten Browsersitzung, wurde zusätzlich ein eingebetteter Internet Explorer eingesetzt. Dies hat den Vorteil, dass zum einen installierte Browser Erweiterungen keine ungewollten Daten über das Firmennetzwerk senden können und zum anderen die aufrufbaren Webseiten bereits auf Client Seite eingeschränkt werden können, da sich die Eingabe der URL deaktivieren lässt. Eine Einschränkung der erreichbaren Webseiten ist auch auf Serverseite möglich, jedoch können Webseiten auch Teile einer anderen Seite einbinden. In diesem Fall ist es schwierig zu entscheiden, ob diese Adresse erlaubt ist oder nicht. Durch das Deaktivieren der freien URL Eingabe wird das Firmennetzwerk stärker vor einem möglichen Missbrauch der Verbindung geschützt, da so nur ausgewählte und erlaubte Seiten erreicht werden können und eine Verbindung nur mittels des eingebetteten Internet Explorers möglich ist.

Durch die Einbettung des Browsers in ein eigenes Fenster wird nur die Webseite selbst angezeigt, nicht jedoch die Navigationsleiste oder andere Toolbars. Aus diesem Grund ist es schwierig, in diesem Browserfenster zur vorherigen Seite oder auch zur Startseite zu navigieren, da dies nur über das Kontextmenü per Rechtsklick erreicht werden kann. Daher wurde für die Lösung in der Bachelorarbeit eine eigene Navigationsleiste implementiert. Zum Erzeugen der Navigationsleiste wurde WTL (Windows Template Library) eingesetzt und drei Buttons hinzugefügt. Diese bieten die Funktionalitäten „zurück“, „vorwärts“ und „Home“. Letzterer navigiert den Benutzer auf eine Übersichtsseite der möglichen Webseiten.

Die Definition, welche Webseiten für den Mitarbeiter über den HTTP-Proxy verfügbar

sind, können über eine JSON Datei auf dem Firmencomputer vorgenommen werden. Dies kann je nach Anwendungsfall der Mitarbeiter selbst oder auch ein Administrator vornehmen. Beim Verbindungsaufbau wird diese Konfiguration verschlüsselt an den Laptop übertragen. Dieser erzeugt aus der Liste der Webseiten eine HTML Seite, die dann in dem eingebetteten Webbrowser als Startseite angezeigt wird. Diese Seite ist auf Abbildung 3 zu sehen. Darüber kann der Benutzer anschließend bequem auf die entsprechenden Seiten navigieren und diese Übersicht jederzeit über den Home Button erneut aufrufen. Von den über die Links erreichbaren Internetseiten kann beliebig weiter navigiert werden.



Abbildung 3: Screenshot der Startseite

Fazit

Die aktuelle Version der Software ermöglicht den schnellen und einfachen Zugang zu internen Webseiten einer Firma. Dabei wird auf dem Firmencomputer festgelegt, welche Seiten verfügbar sind. Von der Startseite aus kann beliebig auf diese Seiten navigiert werden. Die in einer Toolbar ergänzten Buttons ermöglichen ein einfaches Zurückspringen auf die vorherige Seite, sowie auf die Startseite. Durch die Wahl des Proxies können andere Netzwerkaktivitäten ohne Einschränkungen parallel ablaufen. Zudem können mehrere Mitarbeiter zeitgleich über einen Firmencomputer interne Webseiten aufrufen. Dieses Projekt bietet nur die Grundfunktionalitäten und kann nach Abschluss der Arbeit noch um weitere Funktionalitäten ergänzt werden.

[1] Leech u.a., SOCKS Protocol Version 5, 1996, <https://tools.ietf.org/html/rfc1928>, Zugriff am 21. November 2016

[2] Ferguson, Huston What is a VPN?, 1998, <http://www.potaroo.net/papers/vpn.pdf>, Zugriff am 21. November 2016

Bildquellen:

- Abbildung 1-3: Eigene Darstellung

Entwicklung eines spezifischen Benchmarks zur Performance-Analyse neuer Rechnerstrukturen für Motorsport-Steuergeräte mittels aus MATLAB und Simulink generiertem Quellcode

Jan Baumann*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Zur Beurteilung der Leistung eines Systems werden oft Benchmarks eingesetzt. Im Allgemeinen versteht man unter Benchmark einen genormten Test, der bestimmte Lasten erzeugt, die für das Testsystem typisch sind. Diese Lasten sollen Abläufe nachbilden, die in der Regel häufig auf dem zu testenden System ausgeführt werden. Dabei ist das Ziel eines Benchmarks, eine reproduzierbare Kennzahl zu liefern, anhand derer sich ein Vergleich der getesteten Systeme durchführen lässt [1]. Dabei gilt es zu beachten, dass verschiedene Benchmarks zueinander nicht vergleichbar sind. Jeder Benchmark setzt den Fokus auf andere Abläufe, beispielsweise Basisrechnungen oder Textvergleiche. Deshalb muss darauf geachtet werden, dass der Benchmark entsprechend der Verwendung des zu testenden Systems gewählt wird.

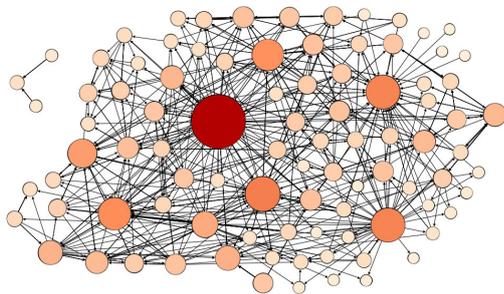


Abbildung 1: Hauptfunktionen (Knoten) des Motorsport-Steuergeräts und deren Verbindungen (Kanten) untereinander

Motivation

Da die stark verbreiteten Benchmarks, wie beispielsweise Dhrystone und Linpack, sich vor allem auf Workstation- oder Serversysteme fokussieren, liefern sie für ein Motorsport-Steuergerät (MS) keine optimalen Erkenntnisse [2][3]. Die Funktionsstruktur eines MS unterscheidet sich erheblich von der eines Serversystems. So sind zum einen die Regelalgorithmen sehr stark miteinander verstrickt, was eine Parallelisierung sehr erschwert. Eine typische

Vernetzung ist in Abbildung 1 dargestellt, wobei die Knoten einer Funktion und die Kanten den ausgetauschten Daten entsprechen. Dabei gilt, dass je größer der Knoten, desto mehr Daten von der Funktion ausgetauscht werden. Zum anderen sind die Funktionen von unterschiedlichen Mustern geprägt, die so in anderen Systemen kaum ins Gewicht fallen. Wegen der erwähnten Unterschiede ist also ein spezieller Benchmark für MS nötig, um unter Berücksichtigung der gegebenen Umstände relevante Ergebnisse zu erhalten.

Mit einem geeigneten Benchmark für MS lassen sich neue Rechnerstrukturen im Voraus testen. So kann vor einem Architekturwechsel oder Chip-Upgrade bereits auf einem Evaluationsboard die entsprechende Hardware getestet und mit der aktuellen verglichen werden. Dies ermöglicht es zu beurteilen, welche Hardware sich am besten für das MS eignet. Zusätzlich kann durch einen skalierbaren Benchmark auch eine Größenänderung der Software evaluiert werden. Dies kann erreicht werden, indem der Benchmark um einen entsprechenden Faktor vergrößert und dann die gewonnene Kennzahl mit der vorherigen verglichen wird. Bricht die Leistung im neuen Test stärker ein, als der Faktor um den der Benchmark vergrößert wurde, ist davon auszugehen, dass die Systemleistung leiden wird.

Analyse

Wie bereits erwähnt wurden die Funktionen des MS und die ausgetauschten Daten analysiert und in einem Graphen dargestellt. Ein weiterer Analyseschritt war der Vergleich diverser gängiger Benchmarks, wie zum Beispiel Dhrystone, Linpack, usw. Diese wurden auf dem MS und zum Vergleich auf einem Raspberry Pi 3 und Odroid C2 ausgeführt. Dies soll einen Grundvergleich der Prozessor-Architektur des MS liefern. Die wichtigste Grundlage für die Implementierung lieferte die Analyse des Funktionsrahmens. Ziel dabei war es, wiederkehrende Muster in der Software des MS zu finden. Diese Muster wurden an Hand

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

ihrer Abtastzeit und Häufigkeit gewichtet. Beispielsweise wird ein Muster „A“ häufig gefunden, 10% der Funktionen beinhalten dieses Muster. Je nach der Funktion, die das Muster enthält, variiert jedoch die Abtastzeit des Musters (Abb. 2).

Implementierung

Ziel der Arbeit war es, den Benchmark als Simulink-Modell zu implementieren und daraus Code generieren zu lassen. Dies entspricht auch dem Vorgehen bei der eigentlichen Software. Da jedoch nicht das gesamte Simulink-Modell manuell gezeichnet werden soll, wurde ein Matlab-Skript geschrieben, welches automatisch alle gewünschten Blöcke hinzufügt. Zusätzlich definiert das Skript die entsprechenden Parameter und fügt die Verbindungslinien zwischen den Blöcken hinzu. Dies erlaubt somit eine Skalierbarkeit des Benchmarks. Wird im Matlab-Skript die gewünschte Größe entsprechend angepasst, werden mehr Blöcke hinzugefügt. Für das Beispiel bedeutet dies, dass das Muster „A“ 10% des gesamten Benchmarks ausmachen soll. Dabei können die Parameter und die Abtastzeit entsprechend angepasst werden.

Testen

Der Code des Benchmarks lässt sich nach der Generierung sowohl auf dem MS, als auch auf anderen Plattformen mit C-Compiler ausführen. So können zur Bewertung neuer Hardware entsprechende Evaluationsboards getestet werden. Dadurch lassen sich mehrere Varianten vergleichen und die beste Alternative finden. Außerdem kann durch Skalieren des Benchmarks simuliert werden, wie sich eine erweiterte Software auf der aktuellen Hardware ausführen lässt.

Zusammenfassung

Abschließend wird in Abbildung 2 nochmals der Ablauf der Entwicklung des Benchmarks dargestellt. Basierend auf den analysierten Funktionen des MS wurden Simulink-Modelle nachgebildet, welche dessen Aufbau widerspiegeln. Durch ein Matlab-Skript wird der Benchmark in der gewünschten Größe als Simulink Modell erstellt und schließlich der Code generiert. Der generierte Code lässt sich schließlich auf verschiedenen Systemen testen.

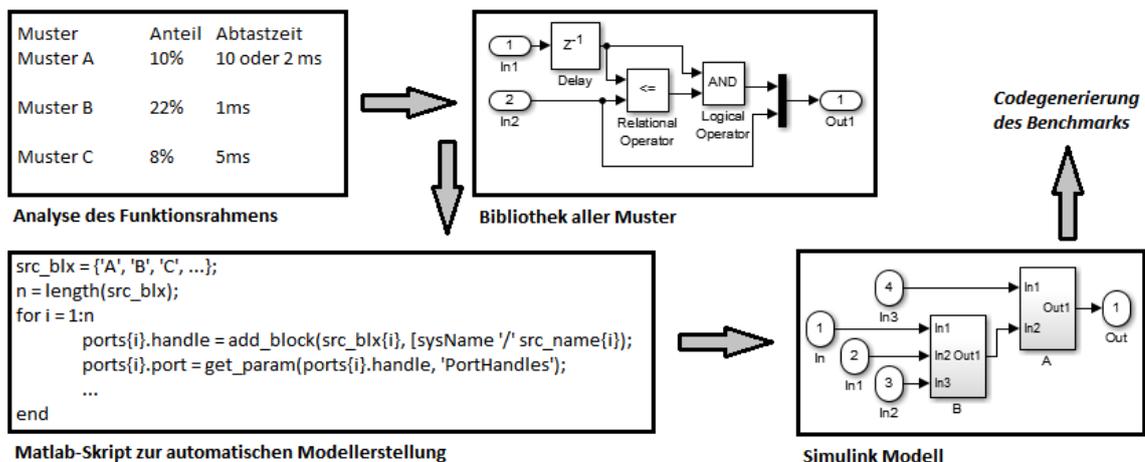


Abbildung 2: Schematisches Vorgehen bei der Benchmark Entwicklung

- [1] ITWissen (Hrsg.): Definition Benchmark. www.itwissen.info (15.11.2016)
 [2] Longbottom, R.: Dhrystone Benchmark. www.roylongbottom.org.uk (15.11.2016)
 [3] Prometheus GmbH (Hrsg.): Linpack Benchmark www.top500.org/project/linpack (15.11.2016)

Bildquellen:

- Abbildung 1-2: Eigene Abbildung

Visualisierung elektrischer Ströme mit Android

Tobias Deißler*, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Erweiterte Realität in der Bildung

Die erweiterte Realität, oft Augmented Reality (AR) genannt, wurde bislang in verschiedenem Umfang definiert. Grundsätzlich zeichnet sie sich durch das möglichst nahtlose Einbringen zusätzlicher Information in die verschiedenen Sinne unserer Wahrnehmung aus. Des Weiteren wird der Begriff teils durch Voraussetzung einer Darstellung in Echtzeit oder einer Mensch-Maschine-Interaktion einzugrenzen versucht [1]. Durch die hohe Performance der Berechnungs- und Ausgabegeräte können heute ganze dreidimensionale Objekte perspektivisch korrekt unserer Wahrnehmung hinzugefügt oder real vorhandene Objekte im Virtuellen in Form und Farbe verändert werden. Das Loslösen komplexer Anwendungen von stationären Rechnern und damit einhergehender Verbreitung auf mobilen Geräten wie Tablets und Smartphones trägt zur Ausbreitung von AR bei. Der aufkommende Einsatz von Datenbrillen steigert deren Benutzbarkeit nochmals erheblich.



Abbildung 1: App „Curiscope Virtuali-Tee“

Neben ihrer Verbreitung in sämtlichen Branchen hat die AR auch vor Bildungseinrichtungen keinen Halt gemacht, um sowohl in spezialisierten Mitarbeiterschulungen in der Industrie, als auch im öffentlichen Bildungswesen, die Ausbildung effizienter zu gestalten. So können sinnvoll eingesetzt, komplexe Sachverhalte auf interaktive Weise besser vermittelt werden, als mit statischem Videomaterial, Bildern und Texten. Häufig bestehende Probleme mit der räumlichen Vorstellung können so

angegangen und die Neugier an abstrakten Inhalten spielerisch gesteigert werden.

AR-Plattform Vuforia

Bei Vuforia handelt es sich um eine Plattform zur Entwicklung von AR-Anwendungen für Android, iOS oder die Universal Windows Platform. Zu ihren Leistungsmerkmalen zählt es, den Entwickler zu befähigen, in seiner Software Muster-, Text- und Objekterkennung einzusetzen. Das Tracking von Objekten bzw. der Position des Benutzers erfolgt dabei stets merkmalsbasiert (engl. „feature“).

Dies kann sich hier einerseits auf zweidimensionale, eine ID codierende, Muster oder vom Benutzer selbst festgelegte Grafiken stützen. Andererseits können auch räumliche Objekte nach vorherigem 3D-Scan zur Positionsbestimmung eingesetzt werden [2]. Da Features anhand kontrastreicher Punkte des zu verfolgenden Objekts erstellt werden, muss der Entwickler auf deren zahlreiches Vorhandensein achten. So eignet sich ein Bild mit vielen punktuellen, weit verbreiteten, Helligkeitsunterschieden sehr gut, eine einfarbige, nur aus Rundungen bestehende Form hingegen gar nicht (s. Abb. 2).

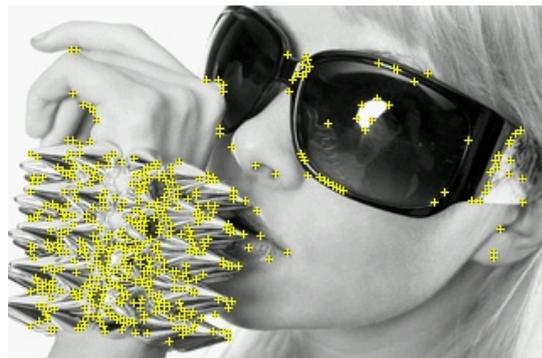


Abbildung 2: Erkannte Features einer Foto-Marke

Die Unterstützung der Game-Engine Unity 3D macht diese auch im Bereich der AR-Anwendungsentwicklung zu einem mächtigen Werkzeug. Konkret bedeutet dies, dass in Selbige ein spezielles Asset-Paket importiert und auf wiederverwendbare Objekte, sogenannte

*Diese Arbeit wurde durchgeführt an der Fakultät Informationstechnik

„Prefabs“ zurückgegriffen werden kann. Diese stellen beispielsweise eine Kamera zur Objekterkennung oder die zur Nachverfolgung verwendeten Marken dar. Der Zugriff auf das API erfolgt schließlich über die diesen bereits zugeordneten Skripte oder in vom Entwickler neu geschaffenen Programmcode.

Das Projekt

Einer der Gründe für das frühe Ausscheiden von Studenten aus ihrem Studium an der HS Esslingen sind unter anderem Probleme mit dem Verständnis der Grundlagen der Elektrotechnik. Dieser Ursache versucht die im Rahmen dieser Arbeit entwickelte Android-Anwendung entgegenzuwirken, in dem sie sich der Simulation von Strömen in elektrischen Netzwerken annimmt und dabei auf die genannten Technologien zurückgreift. Der Nutzer platziert quadratische Marken in einem Raster. Jede Marke stellt für sich ein Bauteil bzw. eine Verknüpfung von Leitungen dar und bildet im Zusammenspiel mit den umliegenden Elementen ein elektrisches Netzwerk, angelehnt an die aus Vorlesungen bekannten Ersatzschaltbilder (s. Abb. 3).

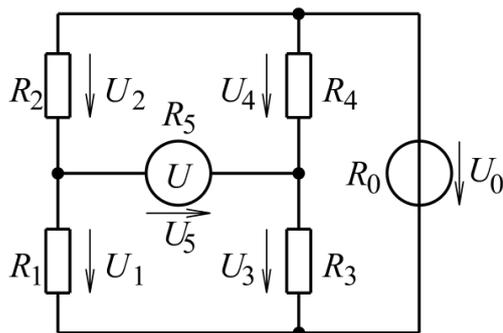


Abbildung 3: Ersatzschaltbild

Im Hintergrund leistet die Anwendung nun unter anderem das Zusammenstellen der für die Netzwerkberechnung nach der Zweigstromanalyse benötigten Elemente wie Netzwerk-knoten, Zweige und Maschen. Schließlich werden diese zur Bestimmung der veränderlichen physikalischen Werte verwendet, wie dem momentanen Stromfluss durch das Bauteil oder der über ihm abfallenden Spannung. Durch Betrachten des Aufbaus mittels der Kamera des Mobilgeräts erscheinen die Bauelemente dem Nutzer plastisch durch 3D-Modelle augmentiert und von den berechneten Strömen durchflossen. Dieser verdeutlicht sich daraufhin visuell, durch Experimentieren an der Schaltung und eingehender Betrachtung der Verzweigungen der Ströme, die Grundgesetze der Elektrotechnik und eignet sich ein intuitives Wissen über diese an. Sowohl seine Änderungen an der Markenordnung als auch Veränderungen der, in der Realität fixen, Bauteilparameter sollen zu weiteren Erkenntnissen führen. Neben der grafischen Darstellung der Ströme können darüber hinaus numerische Werte eingeblendet werden, die Informationen über weitere Größen wie die elektrische Spannung oder den ohmschen Widerstand geben.

[1] Dörner, Broll, Grimm, Jung: Virtual und Augmented Reality, S. 245, Springer Vieweg, 2013

[2] <https://www.vuforia.com/Features> (Zugriff: 20.11.2016)

Bildquellen:

- Abbildung 1: http://www.electronicproducts.com/Mobile/Apps/This_augmented_reality_app_allows_you_to_see_inside_the_human_body.aspx
- Abbildung 2: <https://library.vuforia.com/articles/Solution/Natural-Features-and-Ratings>
- Abbildung 3: <https://de.wikipedia.org/wiki/Ersatzschaltbild>

Konzeption und Entwicklung eines Tools zur Automatisierung des Prozesses der Testdatenidentifizierung in globalen SAP Projekten

Jochen Eberle*, Thomas Rodach, Dirk Hesse

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Die Anzahl parallel laufender Software-Großprojekte in Unternehmen erhöht sich ständig bei einer gleichzeitigen Verkürzung der Release-Zyklen. Dies führt zu einer Erhöhung des Testaufwands und sorgt für steigenden Zeitdruck. Dadurch kommt der Effizienz des Softwaretestens eine zunehmende Bedeutung zu. Ziel ist es, die Abläufe innerhalb des Testprozesses weitestgehend zu automatisieren. Der Fokus liegt dabei i.d.R. auf der Testdurchführung, dem Kern-Prozessschritt eines jeden Testprozesses. Mit Hilfe sogenannter Capture-and-Replay-Werkzeugen wird dabei eine einmalige, manuelle Ausführung eines Testfalls aufgezeichnet und kann so bei Bedarf erneut automatisch ausgeführt werden [1]. Auch im Zusammenhang mit Testumgebungen und Testdaten gibt es Möglichkeiten einer Automatisierung.

1. Testdaten, welche zur Vorbereitung der Testumgebung benötigt werden
2. Ein- und Ausgabedaten während des Tests
3. Ergebnisdaten, welche durch den Testfall erzeugt und auf dem Ausgabegerät angezeigt oder im System gespeichert werden [2]

Die Daten für die Vorbereitung der Testumgebung werden meist mit Hilfe von Automatisierungswerkzeugen, i.d.R. frei am Markt erhältlicher Standardsoftware, generiert. Daten werden dazu von Produktivsystemen selektiert und in die Testumgebung kopiert. Dieser Prozess wird als Testdatengenerierung bezeichnet. In der Phase „Testanalyse und Testentwurf“ werden Testfälle entworfen und in diesem Zuge die Eingabedaten für die Testdurchführung definiert und vorbereitet. D.h. in den einzelnen Schritten des Testfalls muss genau angegeben werden, was der Tester in das System einzugeben hat. Die Testschritte enthalten dazu variable Platzhalter. Im Rahmen des als Testdatenidentifizierung bezeichneten Prozesses werden die Testdaten ermittelt und die Variablen ersetzt.

Die übliche Vorgehensweise ist eine manuelle Identifizierung und Bereitstellung der Testdaten, Standardsoftware ist nicht verfügbar. Die Gründe hierfür liegen in den sich stark unterscheidenden, individuellen Prozesslandschaften der Unternehmen und den sich daraus ergebenden Anpassungen an den SAP Systemen. Sowohl die Testprozesse, als auch die Testfälle mit den zugehörigen Testdaten variieren stark. Soll dieser Prozess automatisiert werden, muss die Softwarelösung genau an die individuellen, unternehmensinternen Testfälle angepasst werden, welche ihrerseits wiederum genau das Unternehmen, die Prozesse, und letztlich das SAP System widerspiegeln.

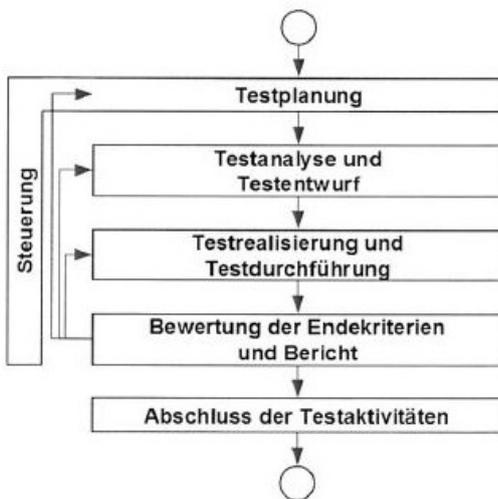


Abbildung 1: Der generische Testprozess

Die Rolle von Testdaten

Testdaten kommen meist im Zusammenhang mit Testfällen auf System- und Akzeptanztest Ebene, also Tests aus Kundensicht, zum Einsatz. Unterschieden werden Testdaten grundsätzlich in folgende drei Kategorien:

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

Der Prozess der manuellen Testdatenidentifizierung

Das manuelle Vorgehen, wie es derzeit bei der Robert Bosch GmbH durchgeführt wird, ist zeit- und kostenintensiv. Die Identifizierung kann auf zwei Varianten erfolgen:

- Das Central Test Team (kurz CTT), ein speziell für Testaufgaben zuständiges Team, führt den Prozess aus und berechnet Kosten pro Testfall
- Ein Prozessexperte, häufig auch der Ersteller des Testfalls, stellt die Testdaten selbst bereit

Unabhängig von der ausführenden Datei läuft der Vorgang immer auf die gleiche Art und Weise ab, wie in Abbildung 2 beispielhaft dargestellt. Die Person erhält eine Datei, die alle Testfälle mit den zugehörigen benötigten Testdaten enthält. Nachdem sie sich einen Überblick über Anforderungen verschafft hat, muss sie die Daten manuell auf dem SAP Testsystem mit den entsprechenden Transaktionen ermitteln. Dieser Prozess dauert bei Ausführung durch einen Prozessexperten für ein Prozessgebiet mit durchschnittlich 12 Testfällen 40 bis 60 Minuten.

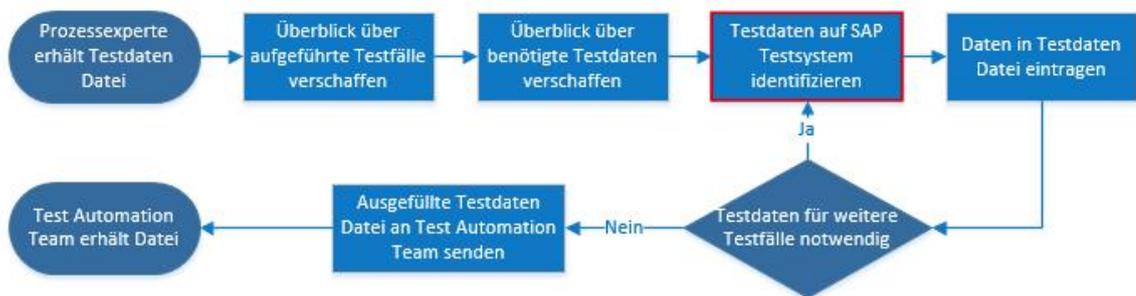


Abbildung 2: Prozess der Bereitstellung von Testdaten für einen Auto Test durch einen Prozessexperten

Ein wichtiger Punkt ist die Qualität der Testdaten. Zwar sind die Daten genau definiert, aber die Art der benötigten Daten kann von Testfall zu Testfall variieren. Beispielsweise wird häufig eine Materialnummer benötigt, doch um herauszufinden ob es sich um eine 10- oder 13-stellige Materialnummer handelt, ist es wichtig, den Testfall und den Prozess inhaltlich zu verstehen. Die qualitativ hochwertigeren Testdaten liefert in aller Regel der Prozessexperte, aus Zeitgründen wird der Prozess aber trotzdem häufig an Mitarbeiter des CTT abgegeben. Dadurch sind inhaltliche Fehler oder Ungenauigkeiten keine Seltenheit. Beispielsweise kann es vorkommen, dass die Testdaten inhaltlich nicht zusammenpassen, z.B. passt ein Lagerort nicht zu einer Materialnummer. Oder bestimmte Daten, meist Bewegungsdaten wie z.B. eine Produktionsauftragsnummer, sind nicht mehr aktuell. In beiden Fällen würde der Testfall auf Fehler laufen aufgrund der mangelnden Qualität der Testdaten. Ein Tool, das diesen Prozess automatisieren soll, muss daher alle Regeln, die bei einer manuellen Ausführung zu beachten sind, berücksichtigen und abbilden.

Anforderungen an ein Tool zur Automatisierung der Testdatenidentifizierung

Die Basis des Tools sollen vom SAP Testsystem heruntergeladene Tabellen bilden. Eine direkte Schnittstelle zum SAP System ist wenig sinnvoll, da für jede Testphase ein neues Testsystem aufgesetzt wird und die Schnittstelle

bei jedem Mal entsprechend angepasst werden müsste. Hinzu kämen erhebliche Berechtigungsprobleme, da direkt auf SAP Stammdaten zugegriffen wird.

Nach dem Download der notwendigen SAP Tabellen werden diese in das Tool importiert. Innerhalb des Tools soll die Möglichkeit bestehen, mit einer 13-stelligen Materialnummer als Suchkriterium alle für ein Prozessgebiet notwendigen, zur Materialnummer gehörigen Daten zu ermitteln.

Produce to Stock	
Warehouse Management & Material Flow	
Import SAP Tables	
Create Test Data	
Credentials	
Material Number	H479.845.240-6RT
Test Data	
Material Number (13 digit)	H479.845.240-6RT
Material Number (10 digit)	H479.845.240
Plant	961W
Warehouse Number	96W
Storage Type	802
Storage Bin	UD
Storage Location	HIEA
Vendor	97900904
Purchase Order	47066172
MM Scheduling Agreement(PO)	55064389

Abbildung 3: Ausschnitt aus einem VBA-Pilotprogramm

Abbildung 3 zeigt wie eine mögliche Realisierung aussehen könnte mit Testdaten zu einer Materialnummer für den Prozessbereich „Warehouse Management & Material Flow“. Eine Umsetzung mit VBA bietet sich an, da sowohl den Input als auch den Output des Tools Excel Dateien darstellen. Die Herausforderung bei der Erstellung des Tools liegt letztlich in der Abbildung aller Regeln, welche berücksichtigt werden müssen um qualitativ hochwertige Testdaten zu erhalten.

Ausblick

Die probeweise Arbeit mit dem Pilotprogramm hat gezeigt, dass das Tool eine sehr große Zeitersparnis bringt. Bei einer manuellen Durchführung steigen Zeit- und Kostenaufwand bei steigendem Testaufwand linear. Der Einsatz des Tools dagegen bringt zwar bei der erstmaligen Nutzung für eine Testphase einen relativ hohen Zeitaufwand von ca. 30 Mi-

nuten mit sich, da alle Tabellen heruntergeladen werden müssen. Aber zum einen ist dieser Zeitaufwand immer noch geringer als die manuelle Identifizierung, zum anderen ist die Zeitersparnis ab der zweiten Nutzung enorm, wenn die Datenbasis bereit ist und lediglich eine Materialnummer eingegeben werden muss. Die Ausführung des Programms dauert dann nur wenige Sekunden. Des Weiteren ist eine hohe Qualität der Testdaten sichergestellt, da die Regeln für die Identifizierung fest hinterlegt sind und so die bereits erwähnten Risiken minimiert werden.

Nach dem erfolgreichen Test für das Prozessgebiet „Warehouse Management & Material Flow“, ist geplant auch die anderen beiden Gebiete der Prozesskette „Produce to Stock“, „Production & Material Planning“ und „Production Preparation & Execution“ in das Tool zu integrieren. So erhofft man sich, die Vorteile die sich durch die Nutzung des Tools ergeben, großflächiger auszunutzen.

[1] Jackson, Recardo. 2009. Testmanagement: Professionelles Testen. Informatik Spektrum. 2009, Bd. 32.

[2] Goll, Joachim. 2012. Methoden des Software Engineering. Funktions-, daten-, objekt-, und aspektorientiert entwickeln. Wiesbaden : Springer Vieweg, 2012.

Bildquellen:

- Abbildung 1: G. Muth Partners GmbH. 2015. [Seminar-Unterlagen]. Certified Tester Foundation Level. Wiesbaden, Deutschland : 2015
- Abbildung 2-3: Eigene Erstellung

Benefit of Quality

Rasty Faraj*, Gabriele Gühring, Dirk Hesse

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Ein Kunde hat genaue Vorstellungen und Erwartungen, die ein Produkt, z.B. eine Kaffeemaschine, erfüllen muss, um seinen Bedürfnissen gerecht zu werden. Sie muss gut verarbeitet sein, optisch ins Auge fallen und zeitgleich preiswert sein. Klar ist, dass alle Kaffeemaschinen den Fokus auf einem Ziel haben: Am Ende soll der Kunde eine warme Tasse Kaffee genießen. Diese Anforderung erfüllt eine 30.000€ Maschine, jedoch aber auch eine Maschine für 30€.

Hat ein Hersteller ein Qualitätsmanagementsystem in seinem Betrieb eingeführt und aktiv gelebt, so weiß er bereits bei der Entwicklung genau, welche Anforderungen der Kunde an das Produkt hat. Er wird sich bemühen von vorn herein für eine hochwertige Optik und langlebige Materialien zu sorgen.

Angewandte Prozesse und Methoden werden durch engagierten Mitarbeiter so umgesetzt und optimiert, dass der Hersteller dem Kunden durch eine hochwertige Kaffeemaschine ein einzigartiges Kaffeeerlebnis, zu einem angemessenen Preis anbieten kann.

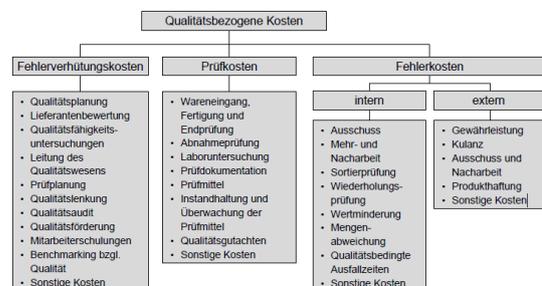


Abbildung 1: Qualitätsbezogene Kosten

Durch einen kontinuierlichen Verbesserungsprozess und daraus abgeleiteten Maßnahmen kann ein Hersteller seine Produkte qualitativ höherwertiger gestalten und durch reduzierte Fehlerkosten, einen wirtschaftlichen Vorteil in einem stabilen Umfeld für das Unternehmen schaffen, ohne gegen die eigenen Unternehmenswerte zu verstoßen. Diese Faktoren und unzählig weitere beeinflussen die Qualitätswahrnehmung eines Kunden und verleiten ihn in ihrer Gesamtheit mit höherer Wahrscheinlichkeit dazu, diese Kaffeemaschine zu kaufen.

Mit einem ähnlichen Fall müsste sich damals die WMF AG auseinandergesetzt haben. Das zertifizierte und erfolgreiche Qualitätsmanagement der WMF AG entwickelte das Unternehmen zum unbestrittenen Weltmarktführer im Verkauf professioneller Kaffeemaschinen.

Gut sind heute viele, billig auch. Wer im Wettbewerb bestehen will, muss sich aber mit seinem Produkt von der Konkurrenz abheben. Um auf dem internationalen Markt wirtschaftlich erfolgreich zu sein, ist es nicht mehr ausreichend „nur“ eine hochwertige Produktqualität anzubieten. [1] Die Produktqualitätsumgebung wie unter anderem Termin- und Liefertreue, Kundenbetreuung, Zuverlässigkeit, und neuerdings auch die Umweltfreundlichkeit spielen immer weiter zunehmend eine vitale Rolle für den Gesamterfolg des Unternehmens.

„Qualität ist kein Zufall, sie ist immer das Ergebnis angestrengten Denkens“[2]

Nach einem Benchmarking der VDMA (Verband Deutscher Maschinen- und Anlagenbau) machen interne Fehlerkosten bei \emptyset 1,3%, externe Fehlerkosten \emptyset 1,4%, Prüf- und Sortierkosten \emptyset 1,6% und Fehlerkosten \emptyset 0,9% des Umsatzes aus.

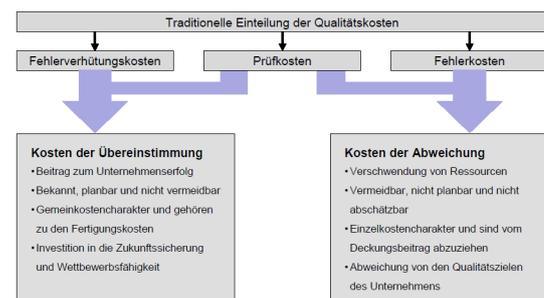


Abbildung 2: Einteilung von Qualitätskosten

Daraus folgt, dass Unternehmen durchschnittlich 2,7% von ihrem Umsatz für Fehler ausgeben oder anders interpretiert, Unternehmen entgehen im Durchschnitt 2,7% ihres Umsatzes vom Gewinn. Bei einem mittelständischen Unternehmen mit 100 Mio. € Umsatz entsprechen dies einem entgangenen Ge-

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

winn in Höhe von 2,7 Mio. Euro jährlich. In einer anderen Dimension spielt die Robert Bosch GmbH. Die hatten nämlich im Jahr 2015 einen Umsatz von 70,7 Milliarden €. Effektives und effizientes Qualitätsmanagement ist hier gefragt denn je.

Je mehr Fehler man proaktiv vermeiden kann, desto geringere Fehlerkosten fallen an und umso mehr bleibt dem Unternehmen als Gewinn. Auch die Unternehmensrisiken reduzieren sich durch Qualität. Weitere positive Aspekte wie gesteigertes Kundenvertrauen, Image-steigerung durch Zuverlässigkeit, zufriedene Mitarbeiter und nicht zu vergessen, die Nachhaltigkeit kommt hinzu und steigert die Unternehmensleistung.

Wie erreicht man also ein solches Qualitätsmanagementsystem?

Mindset: „Qualität ist jedermanns Aufgabe“ [3]

Glückliche und engagierte Mitarbeiter bilden die Säulen eines erfolgreichen Unternehmens. Deren Einbeziehung und aktive Förderung in einer transparenten Unternehmensumwelt quittieren sich durch eine kontinuierliche Verbesserung der Unternehmens-

leistung, geprägt von einem hohen Maß an Qualitätsbestreben und ständig neuen Innovationen. Um diese Verbesserungen mess- und nachvollziehbar haben zu können, werden KPIs (Key Performance Indicator) und KRIs (Key Result Indicator) benötigt. In der wissenschaftlichen Arbeit „Benefit of Quality“ werden die Vorteile von Qualität durch die Verbesserung von KPIs anhand eines Qualitätsverbesserungsprozesses erörtert.



Abbildung 3: Qualitätsverbesserungsprozess

Eine Wertstromanalyse soll dabei unterstützen, Transparenz zu schaffen und nachhaltig nicht wertschöpfende Aktivitäten zu eliminieren. Auch auf weniger messbare Indikatoren wie Mitarbeiterzufriedenheit wird eingegangen und kritisch die Auswirkung in Hinblick auf die Gesamtleistung des Unternehmens hinterfragt.

„Qualität ist eine Investition in die Zukunft, die sich bereits nach kurzer Zeit amortisiert“ [4]

[1] Sonyi, Richard ; Zinser, Hans-Peter: Auf dem Weg zum Zertifikat : Qualitätsmanagement-Systeme in kleinen und mittleren Unternehmen. Berlin Heidelberg New York: Springer-Verlag, 2013. –ISBN 978-3-663-05662-1. S. 3

[2] Zitat von John Ruskin (1819–1900)

[3] Zitat von Armand Vai Feigenbaum (1922–2014) Zitat von Oskar-Werner Nastl, Lead-Auditor im Bereich Customer Service (Robert Bosch GmbH)

Bildquellen:

- Abbildung 1: Brüggemann, Holger ; Bremer, Peik: Grundlagen Qualitätsmanagement : Von den Werkzeugen über Methoden zum TQM. Berlin Heidelberg New York: Springer-Verlag, 2015. –ISBN 978-3-658-09221-4. S. 203
- Abbildung 2: Brüggemann, Holger ; Bremer, Peik: Grundlagen Qualitätsmanagement : Von den Werkzeugen über Methoden zum TQM. Berlin Heidelberg New York: Springer-Verlag, 2015. –ISBN 978-3-658-09221-4. S. 210
- Abbildung 3: Vgl. Werner Lobinger, Hannes Brunner, TÜV-SÜD Gruppe: TÜV-SÜD Schulungsunterlagen – QM-Tools: 5.7-B-QM Ausgewählte QM-Tools in Produktion und Dienstleistungen. Rev.1: Juli 2016, 111115/5.7-B-QM

Analyse und prototypische Realisierung einer Android App zu Verbindung des Connected Car mit anderen Domänen

Oliver Feucht*, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Bis vor einigen Jahren, legte der Kunde beim Kauf eines Automobils den Fokus auf Design und Motorenleistung. Im letzten Jahrzehnt hat die Informationstechnik durch vermehrte Sensorik und das Entertainmentsystem immer mehr Einzug in das Automobil erhalten. Es entwickelt sich nun stetig zum Softwareprodukt. [1]

Zum jetzigen Stand der Entwicklung wird der nächste Schritt der Automobilindustrie das Connected Car sein. Unter dem Begriff Connected Car versteht man die Vernetzung des Automobils mit dem Internet und die Möglichkeit, eine Vielzahl von Services an dieses anzubinden. Durch dieses System kann das Fahrzeug beispielsweise direkt mit dem Autohersteller kommunizieren, bei einem Unfall den Notruf wählen oder auf die online gespeicherte Mediensammlung des Fahrers zugreifen.

Ein weiteres aktuelles Trendthema ist das Smart Home. Smart Home bezeichnet die Vernetzung von Geräten im Haushalt wie beispielsweise Licht, Thermostat, Spülmaschine, Kühlschrank oder auch Unterhaltungselektronik. Diese Geräte können Daten speichern und besitzen eine eigene Steuerungslogik. Smart Home Geräte sind mit dem Internet verbunden und können daher vom Benutzer per App gesteuert und kontrolliert werden. [2]

Auch dem Bereich des autonomen Fahrens gehört die Zukunft, sowohl im privaten PKW, als auch im Transportwesen. Schon jetzt beweisen Testfahrten von autonomen Fahrzeugen, dass die Technik bald alltagstauglich werden kann [3].

Bei all diesen neuen technischen Einsatzgebieten stellt sich die Frage, wie diese zusammen spielen. Welche Möglichkeiten ergeben sich durch die Verbindung des Connected Cars mit anderen Bereichen wie beispielsweise Smart Home oder Autonomem Fahren ?

Im Rahmen der Bachelorarbeit wird die Verbindung von mehreren Domänen mit dem Connected Car untersucht. Betrachtet werden in diesem Zusammenhang hauptsächlich folgende Bereiche

- Home Connect fähige Küchengeräte
- Bosch Smart Home
- autonome Transportroboter

Die Ziele dieser Arbeit sollen gewinnbringende Szenarien sein, die für den zukünftigen Käufer einen Nutzen darstellen, wie beispielsweise die automatische Steuerung der Heizanlage via Internet vom Automobil aus. Im Zuge dessen werden mehrere theoretische Szenarien erarbeitet und einige davon in Form eines Prototyps umgesetzt. Der Prototyp wird als Android App realisiert, die per mySpin Schnittstelle mit der Head Unit des Automobils verbunden ist. Auch die Anbindung der unterschiedlichen Bereiche an die App soll genau betrachtet werden.

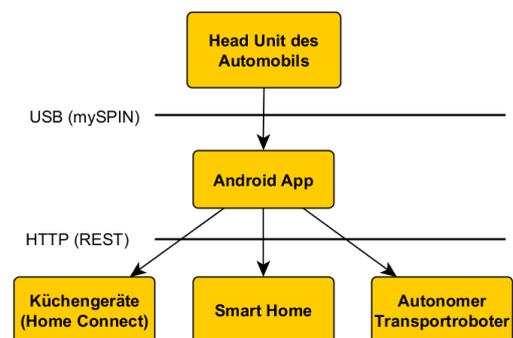


Abbildung 1: Architektur des gesamten Systems

Abbildung 1 zeigt die gesamte Architektur des Systems. Die mySPIN Technologie ermöglicht die Integration des Smartphones in die Head Unit des Autos. Kompatible Apps werden direkt in das In-Vehicle Information (IVI) System eingebunden und im Display des Autos dargestellt. Der Fahrer kann solche Apps mit dem Touchscreen des Automobils bedienen [4]. Die physikalische Verbindung erfolgt per USB Kabel. Das mySPIN Framework gibt freigegebenen Apps Zugriff auf Fahrzeugdaten,

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

wie beispielsweise Reifendruck oder Tankfüllstand. Die Android App enthält die komplette Steuerlogik und übernimmt die Netzwerkkommunikation mit den anderen Bereichen.

Home Connect (HC) ist eine standardisierte Technologie zur Integration von Haushaltsgeräten ins Internet of Things (IoT) Umfeld. Die eigenen Haushaltsgeräte können so von überall aus über das Internet gesteuert und überwacht werden. Es ist beispielsweise möglich auf dem Heimweg ein Spülvorgang seiner Spülmaschine zu starten. Beim Verlassen des Hauses kann der Besitzer auf ein nicht ausgeschaltetes Kochfeld hingewiesen werden.

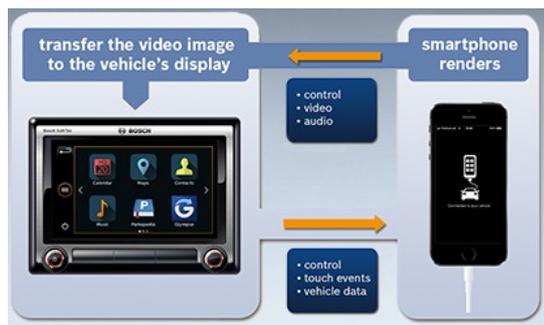


Abbildung 2: Smartphoneintegration mit mySPIN

Der Smart Home Bereich von Bosch beinhaltet Thermostate, Steckdosen, Rauchmelder, Tür- und Fenstersensoren. Die Geräte kommunizieren über ein Funk Protokoll mit dem Smart Home Controller. Der Controller ist die zentrale Steuerungseinheit die auch mit dem Heimnetz verbunden ist. Zur firmeninternen Entwicklung besitzt er eine REST Schnittstelle, mit der die einzelnen Smart Home Geräte gesteuert werden können.

Als autonomer Transportroboter wird der von der Firma Bosch entwickelte Autobod ver-

wendet. Er kann Material eigenständig an einem Lagerort abholen und es zu einem Zielort transportieren [5]. Beim ersten Einsatz in einer neuen Umgebung, wird diese gescannt und zu einer Karte verarbeitet. Danach kann sich der Autobod selbstständig in der bekannten Umgebung fortbewegen und Hindernisse umgehen. Es können Wegpunkte, Strecken und Aufträge definiert werden. Über eine REST Schnittstelle kann man dem Transportroboter die Ausführung eines Auftrags mitteilen.

Die im Rahmen der Bachelorarbeit entwickelte App setzt beispielsweise folgendes Szenario um: Der Benutzer begibt sich auf den Heimweg, steigt in sein Automobil und verbindet sein Smartphone mit diesem. Die mySPIN fähige Smartphone App berechnet als erstes die Fahrtdauer nach Hause. Diese Berechnung erfolgt aus der vom Smartphone gelieferten GPS-Position und der Position des Zielorts. Für eine genaue Berechnung kann auch ein externer Service wie beispielsweise Google Maps Distance Matrix verwendet werden. Alternativ kann auch direkt aus den ausgelesenen Fahrzeugdaten die Fahrtdauer ermittelt werden. Aufgrund des Wissens der verbleibenden Fahrtdauer, können dann Benutzerspezifische Events ausgelöst werden. So kann ab einer verbleibenden Fahrtdauer von mindestens 30 Minuten zu Hause die Heizung angeschaltet werden. Sobald der Benutzer mit seinem Automobil den Heimatort erreicht, wird das Licht angeschaltet.

Ein alternatives Szenario: Ein Zulieferer befindet sich auf dem Weg zum Warenlager. Er hat mehrere Paletten mit Waren geladen. Bei seiner Ankunft wird der Transportroboter automatisch gerufen um die Paletten abzuladen.

-
- [1] V. Johanning und R. Mildner, Car IT kompakt, Das Auto der Zukunft – Vernetzt und autonom fahren. Springer Vieweg, 2015, S. 1-2.
 [2] Wikipedia, Smart Home — Wikipedia, Die freie Enzyklopädie, [Online; Stand 1. November 2016], 2016. Adresse: <http://bit.ly/2gxYJ0J>
 [3] Heise Online, Uber: Autonomer Lastwagen liefert 50.000 Dosen Bier, [Online; Stand 1. November 2016], 2016. Adresse: <http://bit.ly/2gy0f2S>
 [4] Bosch SoftTec GmbH, mySPIN, [Online; Stand 2. November 2016], 2016. Adresse: <http://bit.ly/10AnfY1>
 [5] PresseBox, Bosch steigert mit industrie 4.0 seine wettbewerbsfähigkeit, [Online; Stand 23. November 2016], 15. Juni 1998. Adresse: <http://bit.ly/2gNRC3h>

Bildquellen:

- Abbildung 1: Selbst erstellt
- Abbildung 2: <http://bit.ly/2g72r0h>

Usability Engineering – Prototypische Rekonstruktion der Benutzeroberfläche einer Business Anwendung

Lennard Friedrich*, Dirk Hesse, Thomas Rodach

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Im Rahmen der vorliegenden Bachelorarbeit wird die Restrukturierung des Bedienkonzeptes einer Business Anwendung aufgezeigt und der Ablauf ausführlich beschrieben. Dabei werden mehrere Phasen durchlaufen um eine Oberfläche zu gestalten, die voll und ganz mit den Anforderungen der Benutzer übereinstimmt. Ziel ist es die Usability der Software zu verbessern und den Benutzern die tägliche Arbeit mit dem System zu erleichtern.

Nutzerforschung

Bei einer erfolgreichen Software stimmt die Funktionalität mit den Anforderungen der Benutzer überein. Aus diesem Grund ist es sehr wichtig den Kontakt mit den Anwendern zu suchen, um die Anforderungen exakt zu spezifizieren. Diesen Schritt in der Softwareentwicklung nennt man Nutzerforschung. Es gibt mehrere Methoden der Nutzerforschung, welche sich je nach Art und Umfang der benötigten Informationen in ihrer Durchführung unterscheiden. Eine dieser Methoden ist das Contextual Inquiry. Hier handelt es sich um eine Zwischenform eines klassischen Interviews und einer Beobachtung. Das Contextual Inquiry findet direkt am Arbeitsplatz des Benutzers statt, wo ein Usability-Experte dem Benutzer bei der täglichen Arbeit zusieht und dessen Arbeitsweise genau beobachtet [1].

Datenauswertung

Reine Notizen zu den Beobachtungen und Anforderungen der Benutzer bringt den Entwickler fürs Erste nicht weiter. Deshalb müssen die erlangten Informationen ausgewertet und aufbereitet werden. Dabei spielt es wiederum eine Rolle, um was für eine Art von Informationen es sich handelt.

Eine sehr beliebte Methode ist das Erstellen von Personas. Diese spiegeln die Anforderungen einer Nutzergruppe wider und helfen den Entwicklern sich in die Benutzer hineinversetzen zu können. Dadurch wird das Verständnis für die Anforderungen erhöht und die Qualität der Software verbessert nachhaltig verbessert.

Eine weitere Methode um die Anforderungen darzustellen ist das Erstellen von Szenarien. Diese beschreiben den Arbeitsablauf der Benutzer in einem System und bilden dadurch einen Teil der Anforderungen ab. Durch ein Szenario können teilweise sehr komplexe Funktionalitäten übersichtlich dargestellt werden und somit leichter verstanden werden. In der Abbildung 1 ist ein Szenario zur Abrechnungserstellung dargestellt. Daraus entnehmen kann man die Anforderung, dass ein Benutzer der Software während der Erstellung einer Abrechnung Zugriff auf die Stammdaten haben möchte.

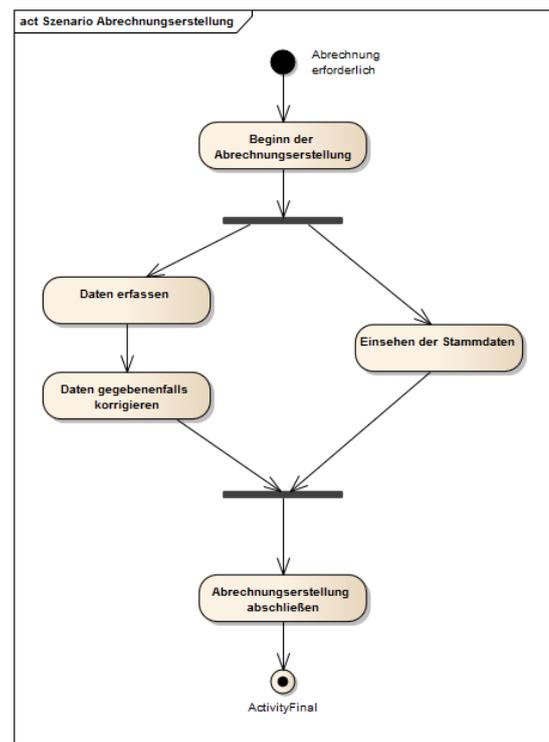


Abbildung 1: Szenario Abrechnungserstellung

Benutzerschnittstelle

Basierend auf den Anforderungen geht es im nächsten Schritt darum, die eigentliche Gestaltung der Benutzerschnittstelle vorzunehmen. Dabei sollte darauf geachtet werden,

*Diese Arbeit wurde durchgeführt bei der Firma iT Engineering GmbH, Pliezhausen

dass die Benutzeroberfläche verständlich und übersichtlich aufgebaut wird. Besonderes Augenmerk sollte auf Nutzerfeedback, Konsistenz sowie einer sinnvollen Auswahl an Interaktionselementen gelegt werden. Zu Beginn ist es empfehlenswert die Ideen auf einem einfachen Papier zu skizzieren. Diese Methode nennt man Paper-Prototyping. Ein solcher Papierprototyp kann einfach angepasst werden

und es wird nicht viel Zeit für den Entwurf benötigt [2]. Nach und nach werden die Papierprototypen zur Evaluation mit den Benutzern verwendet und darauf basierend weiterentwickelt. Ein bereits vollentwickelter Prototyp würde an dieser Stelle eine gewisse Endgültigkeit vermitteln und somit zu weniger Diskussionen anregen. In Abbildung 2 ist ein Papierprototyp beispielhaft dargestellt.

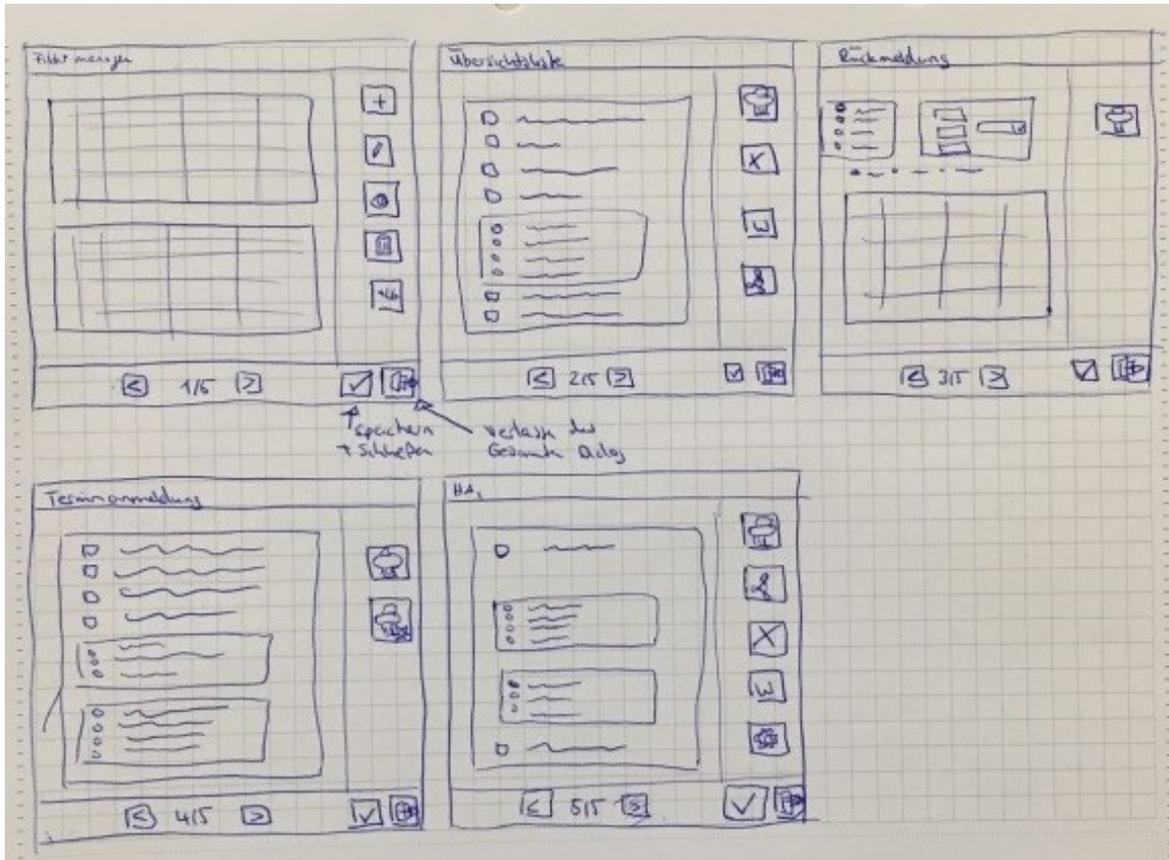


Abbildung 2: Papierprototyp

Ist in der Entwicklung eine gewisse Reife vorhanden, kann ein professioneller Prototyp in der Art eines Klick-Dummys erstellt werden, um letztendlich den Abschluss der Konzeptphase zu bilden.

Zukunft

Auf Basis der Anforderungen und Prototypen, welche während der Bachelorarbeit ent-

standen sind, haben die Entwickler einen eindeutigen Fahrplan zur Umsetzung des Projektes. Es ist gewährleistet, dass die Entwicklung den Anforderungen der Anwender entspricht und das Projekt somit erfolgreich ablaufen wird.

- [1] Moser, Christian (2012): User Experience Design. Mit erlebniszentrierter Softwareentwicklung zu Produkten, die begeistern. Berlin Heidelberg: Springer Berlin Heidelberg (X.media.press)
 [2] Weber, Wibke (Hg.) (2008): Kompendium Informationsdesign. Berlin u.a.: Springer (X.media.press)

Bildquellen:

- Abbildung 1-2: Eigene Darstellung

Der Begriff der Modularisierung vor dem Hintergrund der Geschichte der Entwurfsprinzipien

Philip Furtwängler*, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Bereits beim Bau der ersten Softwaresysteme in den 60er Jahren wurde erkannt, dass es sinnvoll ist, den Aufbau der Software modular, also aus Teilsystemen, aufzubauen.

Modularisieren bedeutet mehr, als nur zu separieren. Ein System wird in benachbarte Teile, sogenannte „Module“, aufgeteilt, so dass alle diese Module über entsprechende „schmale“ Schnittstellen miteinander interagieren können. Die Implementierung eines Moduls ist von außen nicht sichtbar. Ein Modul kennt nur die Aufrufschnittstellen eines anderen Moduls. Die innere Struktur eines Moduls ist gekapselt und bleibt nach dem Geheimnisprinzip verborgen. Damit beeinflussen Module einander nur wenig. Alle Informationen sollen über Schnittstellen laufen.

Dies wird als Benutzungsabstraktion bezeichnet. Im folgenden Abschnitt werden historisch wichtige Prinzipien für die heutige Bedeutung der Modularität aufgeführt und diskutiert.

Historische Entwicklung

Das erste Entwurfsprinzip zur Verringerung von Abhängigkeiten zwischen Teilsystemen war das Prinzip **Loose Coupling and Strong Cohesion** von Larry Constantine aus dem Jahr 1968 [1]. Dieses besagt, dass Teilsysteme untereinander schwach gekoppelt, jedoch in sich selbst stark zusammenhängend sein sollen. Das bringt die Vorteile einer einfacheren Änderbarkeit und Austauschbarkeit der Teilsysteme, einer höheren Stabilität des Systems sowie einer unabhängigen Entwickelbarkeit der Teilsysteme und ein getrenntes Testen der Teilsysteme.

Das Prinzip des **Information Hiding** von Parnas aus dem Jahr 1971 (veröffentlicht durch Parnas im August 1971 an der Carnegie Mellon University in Pittsburgh, Pennsylvania) machte den Begriff des Moduls populär [2]. Beim Information Hiding wird Code, bei dem sich Designentscheidungen vermutlich noch ändern werden, zur Erzeugung von Stabilität in Modulen gekapselt und ein Zugriff nur über Schnittstellen erlaubt. Die Implementierung ei-

nes Moduls bleibt also vor dem Zugriff von außen verborgen.

Das Prinzip **Separation of Concerns** von Dijkstra aus dem Jahr 1974 [3] ist ein generelles Prinzip, das zum Ziel hat, eine Ordnung in verschiedene Belange (Interessen, engl. concerns) zu bringen. Bezogen auf die Programmierung bedeutet es, dass ein Programm so in verschiedene Teile separiert wird, dass ein jedes Teil einen anderen Belang hat. Diese Teile sollen sich in ihrer Funktionalität so wenig wie möglich überschneiden. Hierbei ist zu beachten, dass im Gegensatz zum SRP keine physische Aufteilung in Module verlangt wird.

Das **Single Responsibility Principle** von Robert C. Martin aus dem Jahr 2002 [4] besagt, dass ein Modul nur eine einzige Verantwortlichkeit hat und dann stark zusammenhängend ist, wenn es nur eine einzige Kraft gibt, die auf das Modul wirkt und damit zur Änderung des Moduls führen kann. Dementsprechend hat ein Modul nur eine einzige Verantwortlichkeit bzw. Funktionalität als Grund für die Änderung des Moduls. Anzumerken ist, dass die Änderbarkeit eines Moduls als Grund für die Modularisierung bereits 1971 von Parnas gefordert wurde. Es dauerte aber bis zum Single Responsibility Principle von Robert C. Martin im Jahre 2002, bis die Fachwelt die Änderungswahrscheinlichkeit als Modularisierungsgrund akzeptierte. Nach Aussage von Robert C. Martin wurde dieses Prinzip aus der „strong cohesion“ (möglichst hohe logische Bindungsstärke innerhalb eines Moduls) von Loose Coupling and Strong Cohesion abgeleitet.

Definition des Moduls

Ein Modul ist eine abgekapselte Einheit, also ein isoliertes Teilsystem des Entwurfs eines Gesamtsystems, das über Zugriffsschnittstellen verfügt, über die es mit anderen Modulen interagieren kann. Eine Modularisierung eines Systems bedeutet also die Strukturierung dieses Systems in schwach wechselwirkende Systemteile, die Module. Diese können verwendet werden, ohne eine Kenntnis über die innere Struktur eines benutzten Moduls zu haben.

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

Um die einzelnen Teile eines Systems zu finden, muss abstrahiert werden. Bei einer Abstraktion wird das Wesentliche herausgeschält. Dies erfolgt in zwei verschiedenen Schritten. Im ersten Schritt werden Systemteile identifiziert, um das System in Teile zu separieren. Der zweite Schritt beschäftigt sich damit, die schmalen Schnittstellen der Module zu finden, welche die Services der Module anbieten und dabei die Implementierung verbergen (Benutzungsabstraktion).

Vorteile von Modulen

Modularisieren erbringt für die Entwicklung von Softwaresystemen einige Vorteile:

- Eine Einteilung in Module führt zu einer besseren Verständlichkeit des Systems.
- Ein Modul kann verwendet werden, ohne dessen Aufbau zu kennen.
- Durch die schmale Schnittstelle wird sichergestellt, dass das Innere eines Moduls beliebig verändert werden kann, ohne dass dies Auswirkungen auf die Kommunikation zwischen den Modulen hat.
- Ein Modul kann unabhängig von anderen Modulen implementiert werden.
- Durch das Single Responsibility-Prinzip haben Module nur eine einzige Verantwortlichkeit und damit auch nur einen einzigen Grund für Änderungen.
- Module sind kontextunabhängig und sind daher komplett unabhängig von ihrer Umgebung entwickelbar.
- Die Korrektheit eines Moduls ist nachprüfbar, ohne das Gesamtsystem oder andere Module zu kennen.
- Durch die modulare Struktur kann deutlich besser getestet werden.

Konstruktion von Modulen

Module dürfen nach dem Single Responsibility-Prinzip nur eine einzige Verantwortlichkeit haben. Zudem müssen Module nach dem Prinzip der

Kapselung umgesetzt werden. Auch darf die Implementierung eines Moduls nicht für Implementierungsentscheidungen anderer Module herangezogen werden. Der Begriff der Kapselung umfasst sowohl die Realisierung von Information Hiding, als auch die Realisierung einer schmalen Schnittstelle als Abstraktion. Beim Information Hiding wird die Implementierung innerhalb der Kapsel vor der Umgebung verborgen. Bei der Benutzungsabstraktion wird die Kapsel durch eine schmale Schnittstelle nach außen ansprechbar.

Die folgende Abbildung symbolisiert einen korrekten Zugriff auf die Services eines Moduls über die Aufrufschnittstellen der nach außen angebotenen schmalen Schnittstelle:

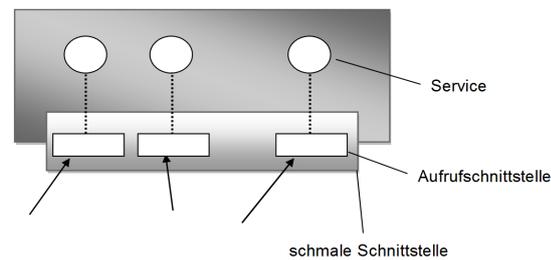


Abbildung 1: Gekapselter Zugriff

Hierbei wird der direkte Zugriff auf die interne Datenstruktur dieses Moduls unterbunden. Es ist stattdessen nur ein **Zugriff über definierte Aufrufschnittstellen** möglich.

Fazit

Module müssen in sich stark zusammenhängend sein und damit eine starke Kohäsion ausweisen. Eine starke Kohäsion erzwingt, dass jedes Modul nur einen einzigen Grund für Änderungen hat (Single Responsibility Principle von Robert C. Martin). Zwischen den verschiedenen Modulen muss eine schwache Kopplung bestehen, damit bei einer Änderung kein Domino-Effekt auftritt. Module dürfen nur über eine schmale Schnittstelle genutzt werden (Benutzungsabstraktion). Die Designentscheidung und Implementierung eines Moduls muss verborgen sein (Information Hiding).

-
- [1] Barnett, T. O., Barnett, L. L. und Constantine, L. L. 1968. Segmentation and Design Strategies for Modular Programming.
- [2] Parnas, David Lorge. 1971. On the criteria to be used in decomposing systems into modules. [Zitat vom: 16. August 2016.] <http://repository.cmu.edu/cgi/viewcontent.cgi?article=2979&context=compsci>
- [3] Dijkstra, E. 1974. On the role of scientific thought. [Zitat vom: 16. August 2016.] <https://www.cs.utexas.edu/users/EWD/transcriptions/EWD04xx/EWD447.html>
- [4] Martin, Robert C. 2002. Agile Software Development: Principles, Patterns, and Practices.

Bildquellen:

- Abbildung 1: Eigener Entwurf

Best Practices beim Übergang eines Unternehmens von einer traditionellen Entwicklungsmethode in eine agile Entwicklungsmethode

Thomas Greule*, Manfred Dausmann, Hans-Gerhard Groß

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Jeglicher Anfänger in der Informatik beginnt seine ersten Schritte wohl mit dem Slogan „Hello-World!“. Dieser wird meist mit der möglichst einfachen Syntax ausgegeben, wodurch ein erstes, winziges Programm entsteht. Natürlich wurde hierfür kein wirklicher Entwicklungsprozess erstellt, dennoch folgte die Entwicklung des Programms einem grundlegenden System: Ein Ziel wurde gesetzt, umgesetzt und getestet. Auch dies ist bereits ein Prozess, diesen jedoch bereits eine Methode zu nennen wäre übertrieben. Die Grundzüge sind allerdings bereits vorhanden, ein geradliniges System an welchem sich die Entwicklung orientiert.

Anschließend wird ein Anfänger weiterhin kleine Projekte durchführen, welche sich mit einer einfachen Projektstruktur realisieren lassen: Wenige Anforderungen von wenigen Parteien, überschaubarer Aufwand, wenige bis keine Abhängigkeiten zu anderen Projekten. Sobald die Projekte jedoch größer werden, wird auch die zugrundeliegende Entwicklungsmethode aufwendiger. Wobei die ersten Projekte noch mit einer grobschlächtigen Vorstellung einer Methode auskamen, wird nun etwas Größeres, Mächtigeres notwendig. Des Weiteren sollte diese Methode auch nicht aus den Gedanken des Entwicklers entspringen, um Verwirrung bei Einsteigern in das Projekt auf ein Minimum zu begrenzen, sondern nach bereits bekannten und gebräuchlichen Entwicklungsmethoden konzipiert werden.

Traditionelle Entwicklungsmethoden

Das Wasserfallmodell oder auch das V-Modell sind Vertreter der traditionellen Entwicklungsmethoden, welche die einzelnen Teile der Entwicklung in Phasen unterteilen [1].

Hier wird jeweils immer nur eine Phase des Projektes zum selben Zeitpunkt durchgeführt. Dies bedeutet, dass sämtliche Planungen für das Projekt durchgeführt werden, bevor die erste Codezeile für dieses geschrieben wird.

Natürlich ist es wünschenswert, sämtlichen anfallenden Aufwand in den Anfangsphasen des Projektes festzustellen. Diese Vorgehensweise hat jedoch einen entscheidenden Nachteil: Sollte sich nur eine entscheidende Voraussetzung für die zugrundeliegende Planung in der Mitte des Projekts ändern, muss die gesamte Planung wiederholt werden.

Für den Anfänger, Einzelgänger oder Studenten mag dies kein Problem sein, dieser muss meist keine oder kaum Rücksicht auf andere Parteien nehmen. Unternehmen jedoch besitzen diesen Luxus nicht, denn ihre Projekte sind meist bedeutend größer, über Jahre oder gar Jahrzehnte, angelegt. Dies schränkt sie in der Flexibilität ein, auf Änderungen der Anforderungen angemessen zu reagieren, was wiederum schnell zu unvorhergesehene Kosten und längeren Entwicklungszeiten führen kann, welche im ursprünglich geplanten Vorgehen nicht berücksichtigt wurden.

Agile Entwicklungsmethoden

Um dieses Problem der traditionellen Entwicklungsmethoden zu adressieren, entstanden um die Jahrtausendwende die ersten agilen Entwicklungsmethoden, wie zum Beispiel Scrum, Extreme Programming oder Kanban. Mit diesen sollten die Probleme, welche durch die stetig größer werdenden Fortschritte in der IT Branche entstanden sind, handhabbar gemacht werden [2]. Hierbei werden die einzelnen Entwicklungsschritte stetig neu durchschritten. Scrum als Beispiel bedient sich dem Deming Cycle [3], zu sehen ist dieser in Abbildung 1.

Die einstmals statischen Systeme für einzelne Parteien in den Anfängen der Softwaretechnik, seien es Banksysteme, die teilweise über Jahrzehnte in Verwendung waren, oder auch die Softwaresysteme für einen Flug zum Mond, sind nunmehr Einzelfälle. Diese Ent-

*Diese Arbeit wurde durchgeführt bei der Firma Parametric Technology GmbH, Sindelfingen

wicklungsvorgänge hatten fest gesteckte Ziele, welche unumstößlich waren, das Verwalten von Vermögen oder die Nutzung von Naturgesetzen, um eine Mission zum Mond zu schicken. Und solange die Voraussetzungen für ein Projekt unumstößlich waren, reichten die traditionellen Entwicklungsmethoden vollständig aus. Heutzutage jedoch basiert ein Großteil der Projekte auf der globalisierten, sich schnell wandelnden Welt, nicht mehr auf den Naturgesetzen wie es für die Software der Apollo 11 der Fall war.

Die Umstellung auf eine Agile Softwareentwicklung mag für den kleinen Entwickler nicht allzu schwer sein, seine Projektlaufzeiten sind kurz und meist nicht zusammenhängend. Somit könnte er einfach ein Projekt mit einer traditionellen Methode abschließen und das nächste mit einer agilen beginnen. Bei einem Unternehmen ist dies jedoch nicht einfach möglich. Meist ist bereits ein großer Prozessapparat in der Entwicklung aktiv, mit welchem auch laufend Projekte entwickelt werden. Und da dies meist verzahnte Projekte sind, können hier Änderungen der Projektstruktur nur zögerlich und mit reichlich Vorbereitung durchgeführt werden. Dabei ist es zweitrangig, ob dies Projekte im eigenen Unternehmen oder in Zusammenarbeit mit gänzlich anderen Unternehmen sind. Es muss für jedes Unternehmen einzeln analysiert werden, in wie weit alte, traditionelle Prozessstrukturen übernommen werden können und welche neue, agile Aspekte in den aktualisierten Prozess einfließen sollen.

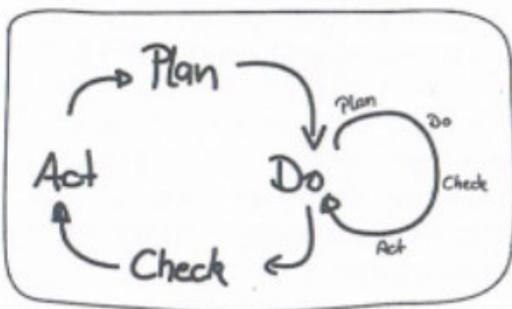


Abbildung 1: Deming Cycle bei Scrum

Dieser Umstieg auf eine agile Entwicklungsmethode ist bereits in einem reinen Softwareunternehmen umständlich. Jedoch ist es nicht

mehr nur die reine Softwareentwicklung, die sich mit dem Problem der immer schneller werdenden Entwicklung der Technik und des globalen Marktes unterwerfen muss. Auch alt-hergebrachte Wirtschaftszweige werden zunehmend mit der Softwareentwicklung verschmolzen, seien es die Fahrzeugindustrie oder die Hersteller von Haushaltsgeräten.

Praktisch jedes Gerät, das entwickelt wird, muss sich mittlerweile den Tatsachen, welche in der Informationstechnik herrschen, unterwerfen. Die Entwicklungszyklen werden auch für diese stetig schneller, da immer neue, bessere Software für diese bereitsteht. Jedoch kann nicht alles wie bei der Entwicklung einer reinen Softwareanwendung von statten gehen. Es müssen physikalische Einflüsse berücksichtigt werden, seien es die Produktionsdauer eines Prototyps oder die Lieferzeit einzelner Bauteile.

Die Probleme und die Vorteile des Umstieges, von einer traditionellen auf eine agile Entwicklungsmethode, sollen im Rahmen dieser Bachelorthesis analysiert werden. Die Erfahrungen verschiedener Unternehmen, welche diesen Wandel bereits durchgeführt haben, werden mithilfe von generalisierten Fragen analysiert. Hierzu werden die Mitarbeiter der Unternehmen zu den Verhältnissen vor und nach dem Umstieg in verschiedenen Bereichen befragt. Dies umfasst unter anderem objektive Daten, wie den benötigten Zeitaufwand für einzelne Entwicklungsvorgänge, die Auswirkung von Fehlplanungen oder sich ändernden Voraussetzungen auf den Entwicklungsprozess.

Auch subjektive Eindrücke fließen in die Bewertung mit ein, so zum Beispiel generelle Zufriedenheit der Mitarbeiter mit dem Prozess, ihre empfundene Arbeitslast und ob es negative Einflüsse während der Umstellung gab. Ebenso werden Probleme, die während des Transfers aufgetreten sind, berücksichtigt. Die Antworten auf diese Fragen werden in generelle Ratschläge für Unternehmen, die ebenfalls diesen Schritt wagen wollen, gewandelt. Diese Ratschläge wiederum sollen als universelle Best Practices zusammengefasst und zur besseren Bewältigung zukünftiger Übergänge anderer Unternehmen bereitgestellt werden.

[1] J. Goll, Methoden des Software Engineering, Esslingen: Springer Vieweg, 2012.

[2] K. Beck, Extreme Programming. Die revolutionäre Methode für Softwareentwicklung in kleinen Teams, München: Addison Wesley, 2000.

[3] B. Gloger, Scrum, München: Carl Hanser Verlag, 2013.

Bildquellen:

- Abbildung 1: B. Gloger, Scrum

Realisierung einer Schnittstelle zwischen einem FPGA-basierten Sercos-Slave und der Arduino-Plattform einschließlich der zugehörigen API

Ricardo Grilo*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Programmierbare integrierte Schaltkreise bieten immer mehr Funktionen an. Daher sind auch Entwicklungsumgebungen umfangreicher und komplexer geworden. Diese benötigen aber lange Einarbeitungsphasen, welche hinderlich sein können. Insbesondere bei der Erstellung sogenannter *proof of concepts*, bei denen eine schnelle Demonstration der Machbarkeit im Vordergrund steht, sind solche Entwicklungsumgebungen von Nachteil. Auch die Lehre profitiert von kurzen Einarbeitungsphasen, da so mehr Zeit auf die Vermittlung weiteren Wissens verwendet werden kann. Deshalb müssen mächtige, aber damit meist auch komplexe Entwicklungsumgebungen nicht immer von Vorteil sein. Zumal diese auch meist mit hohen Lizenzkosten verbunden sind.

Arduino

Arduino ist eine Plattform, die entwickelt wurde, um Interessierten mit geringem technischem Wissen eine kostengünstige Einführung in die Welt der Mikrocontroller zu bieten. Sie besteht aus einer mittlerweile recht großen Anzahl an Boards, die je nach verbautem Mikrocontroller unterschiedliche Leistungsanforderungen erfüllen. Zur Programmierung werden sie von einer einfachen und leicht verständlichen Entwicklungsumgebung begleitet. Diese unterstützt sowohl die Programmiersprachen C++ als auch Processing. Boards und Entwicklungsumgebung sind dabei quelloffen und können ohne Lizenzkosten genutzt werden. So entstehen bei der Nutzung lediglich die geringen Kosten für die Boards. In den letzten Jahren gewann die Arduino-Plattform an großer Beliebtheit und es entwickelte sich ein großes Portfolio an Zubehör wie Aufsteckplatinen, Projekte zur Unterstützung weiterer Programmiersprachen und Bibliotheken.

Sercos

Sercos III ist ein auf Ethernet basierendes Kommunikationssystem für I/O-, Motion

Control- und Safety-Applikationen. Es bietet Echtzeitfähigkeit mit kurzen Zykluszeiten und hoher Synchronität. Sercos basiert auf dem Ethernet-Standard IEEE 802.3 und erlaubt auch eine optionale Nutzung von IP-basierten Protokollen parallel zu dem Echtzeit-Betrieb [1]. Im Echtzeitkanal wird dabei unterschieden zwischen zyklisch übertragenen Echtzeitdaten, den *Connections*, und azyklisch übertragenen Parametern im sogenannten *Service Channel*.

Zielsetzung

Das Ziel dieser Bachelorarbeit ist es, die einfache Welt der Arduino mit der Funktionsvielfalt von Sercos zu verbinden. Dafür wurde im Rahmen der Bachelorarbeit die Implementierung der Schnittstelle zwischen Arduino und Sercos-Slave realisiert. Die Schnittstelle ist dabei das *EasySlave Shield*, siehe Abbildung 1. Im Rahmen der Arbeit wurden die *EasySlave Shield*-Schnittstelle sowie die API und eine beispielhafte Applikation implementiert.

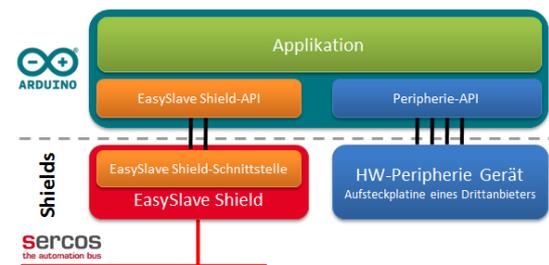


Abbildung 1: Übersicht der Schnittstellen

An die Schnittstelle wurden dabei folgende Anforderungen gesetzt:

- Beidseitige Übertragung der Prozessdaten
- Registrierung von Parametern sowie Parametrierung
- Einfache Programmierschnittstelle innerhalb der Arduino-Umgebung

Diese Anforderungen sollen eine schnelle Eingliederung von Arduino-Boards in ein Sercos-

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

Netz ermöglichen.

Wie in Abbildung 2 zu erkennen, ist das *EasySlave Shield* als Aufsteckplatine für das Arduino-Board realisiert. Diese kommuniziert mit dem Arduino-Board und Sercos via Stiftkontakte beziehungsweise RJ45-Buchsen.

Das Shield enthält einen FPGA, in dem Logik-Schaltungen sowie ein Mikrocontroller realisiert sind. Durch diese vorhandenen Ressourcen können zudem weitere Funktionen auf dem Shield realisiert werden. So können zum Beispiel Pins durch logische Schaltungen im FPGA verbunden und damit umgeleitet werden, weswegen das Shield über eine zweite Reihe an Buchsenleisten verfügt. Auch der Mikrocontroller des FPGA kann weitere Funktionen übernehmen.

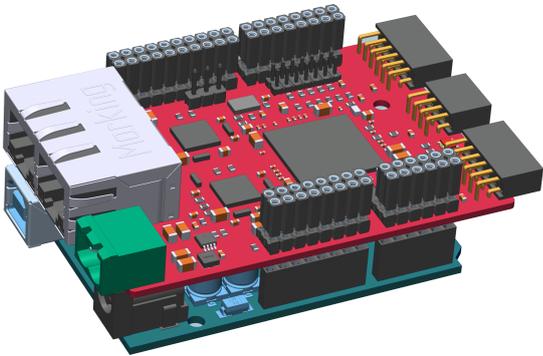


Abbildung 2: EasySlave Shield aufgesteckt auf Arduino-Board

Im Rahmen dieser Abschlussarbeit wurden Kommunikationsmöglichkeiten zwischen Arduino und Aufsteckplatine analysiert. Die größten Einschränkungen stellen dabei die Mikrocontroller der Arduino-Boards dar. Diese bieten die höchsten Übertragungsgeschwindigkeiten an, wenn sie als SPI-Master eingesetzt werden. Aus diesem Grund wurde das Shield als SPI-Slave konzipiert. Dies entspricht nicht der eigentlichen Hierarchie der Datenübertragung, da der Systemtakt von Sercos übertragen wird. Es ermöglicht aber eine höhere Übertragungsrates und eine einfachere Nutzung der Schnittstelle auf der Seite des Arduino-Boards, da dieser nicht gezwungen wird, synchron zum Sercos-Takt zu bleiben.

Verfügbare FPGA-IP-Cores für SPI besitzen leider nicht genug Pufferspeicher, um die Dauer der Ausführung einer nicht unterbrechbaren Funktion des Sercos-Slaves zu überbrücken.

Infolge dessen könnte es zur Übertragung fehlerhaften Nachrichten über das SPI kommen.

Aus diesem Grund musste ein eigener IP-Core mit SPI-Slave entwickelt werden. Dieser sieht einen zyklischen Austausch von Prozessdaten vor. Als Ergänzung kann der IP-Core eine zusätzliche Leitung nutzen, um das Eintreffen neuer Daten an das Arduino-Board zu signalisieren. Diese Leitung kann das Arduino-Board als Trigger für eine Synchronisierung des Prozesses an den Zyklustakt von Sercos nutzen.

Um eine korrekte Übertragung von Nachrichten zu gewährleisten, darf der IP-Core nur entweder das SPI oder den FPGA-internen Zugriff bedienen. Daher ist ein wechselseitiger Ausschluss ähnlich einer Mutex realisiert worden. Nach der Verifikation des IP-Cores durch Simulation wurde eine Bibliothek für die einfache Nutzung des Cores erstellt und in die vorhandene Entwicklung eines Sercos-Slaves integriert.

Für die Arduino-Plattform wurde eine einfache Programmierschnittstelle realisiert, die über wenige zentrale Funktionen einfach angesteuert werden kann. Dabei wird eine Initialisierungsphase verwendet, in der die Datenbreite der übertragenen Prozessdaten und die möglichen Parameter festgelegt werden. Anschließend wechselt die Verbindung in den Übertragungsbetrieb, in dem dann die Prozessdaten und die Parameter ausgetauscht werden.

Die korrekte Funktion der Entwicklung wurde beispielhaft mit einer kleinen Anwendung getestet und demonstriert. Dazu werden zwei Einheiten, bestehend aus Arduino M0 Pro und *EasySlave Shield*, genutzt. In der Anwendung ist an dem einen Arduino-Board ein Gestenmodul und an dem Anderen eine LED-Matrix angeschlossen. Die Daten des Gestenmoduls werden von einem Arduino-Board ausgelesen und über Sercos an eine SPS mit Sercos-Master übertragen. Diese verarbeitet die Daten und gibt Steuerungsdaten über Sercos an das andere Arduino-Board weiter, welches diese auf der LED-Matrix ausgibt. Folglich ist es mit dieser Arbeit gelungen eine Möglichkeit zu erschaffen, anhand der schnell und einfach Sercos-Slave-Geräte auf Arduino-Basis realisiert werden können.

[1] SERCOS International e.V. 2009. General Overview and Architecture V1.1.2.1.3 [PDF] Suessen : s.n., 2009

Bildquellen:

- Abbildung 1: Eigene Abbildung – enthält offizielles Logo von Arduino und Sercos
- Abbildung 2: Steinbeis Embedded Systems Technologies GmbH

Big Data Analytics auf Fahrzeugdaten

Dennis Groß*, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Im Jahr 2018 sollen 70% der verkauften Fahrzeuge in West-Europa vernetzt sein [1]. Durch diese massive Vernetzung kommunizieren die Fahrzeuge untereinander und mit externen Infrastrukturen. Dies führt zu gewaltigen Datenmengen. Die gesammelten Daten enthalten Informationen über den aktuellen Zustand des Fahrzeuges, das Fahrverhalten des Fahrers, der Fahrzeugumgebung und der Insassen.

Die dadurch gewonnenen Daten steigern den Wert eines jeden verkauften und vernetzten Fahrzeuges um ca. 95 € (siehe Abbildung 1) durch ihren Weiterverkauf. Zieht man zusätzlich in Betracht, das dieses Fahrzeug kontinuierlich Daten erzeugt und somit für die Fahrzeughersteller zu einer ebenfalls kontinuierlichen Geldquelle wird, werden aus diesen 95 € über die Fahrzeuglebenszeit hinweg mehrere hundert Euro hinzukommen. Hauptabnehmer dieser Daten sind hauptsächlich Versicherungen. Dadurch können sie ihre Preise individuell an jeden Kunden anpassen und wissen, wie hoch das Risiko eines jeden Kunden bzgl. eines Autounfalls oder ähnlichen ist.

Use Case	Value Per Vehicle Per Year
Pay As You Drive Insurance	€ 2.46*
Pay How You Drive Insurance	€ 79.00
CRM - know the customer - vehicle sales	€ 7.50
CRM - service - MILEAGE	€ 1.60
Prognosis	€ 1.25
Advertising - PULL - proximity deals	€ 0.03
Weather	€ 0.02
Traffic - Live FCD	€ 0.12
Road Maintenance	€ 0.01
Insurance - previous driving record	€ 8.75*
Warranty trends (quality improvement)	€ 5.00
Product design	€ 1.25
Advertising - trip analysis	€ 0.00
Trip analysis - Public	€ 0.01
Advertising - PUSH Radio ads	€ 0.004
TOTAL	€ 95**

Abbildung 1: Fahrzeugdaten und ihr Wert

Bei der Übermittlung der Fahrzeugdaten an den Hersteller, spielen vor allem die sogenannten Diagnostic Trouble Codes (DTC) eine wichtige Rolle. Ein DTC ist Fahrzeugfehlercode, der in dem fahrzeuginternen Fehlerspeicher abgelegt wird, sobald es in einem Fahrzeugsystem zu Problemen kommt. Somit kann ein DTC auftreten, wenn zum Beispiel die

Einspritzdüse nicht mehr funktioniert. In diesem Fall leuchtet die Motorkontrollleuchte im Fahrzeug auf und signalisiert, dass man eine Werkstatt aufsuchen sollte. Die Werkstatt hat anschließend die Möglichkeit, den Fehlerspeicher auszulesen und das Problem zu beheben. Bei vernetzten Fahrzeugen werden die DTCs zusätzlich über eine Telematikeinheit an ein Backend-System gesendet.

Da die Bosch GmbH neben einer eigenen Werkstatte namens Bosch Car Services, eine Connected Vehicle Flotte besitzt, bestand in dieser Bachelorarbeit die Möglichkeit, auf beide Datenquellen zuzugreifen.

Das eigentliche Ziel der Bachelorarbeit war es, herauszufinden, welche DTCs gemeinsam auftreten bzw. voneinander abhängig sind. Hierbei waren vor allem die Zusammenhänge interessant, die nicht in der technischen Beschreibung eines DTCs beschrieben wurden und somit an dieser Stelle noch kein offensichtlicher Zusammenhang entdeckt worden ist.

Für die Datenanalyse wurden vor allem die Daten der Bosch Car Service Werkstätten analysiert. Hierbei handelt es sich um mehrere Terabyte an Daten, die während den Diagnosesessions (unter anderem Fehlerspeicher auslesen) in den Bosch Car Services gesammelt wurden. Anschließend wird noch kontrolliert, ob die gewonnenen Erkenntnisse auf die Daten der Connected Vehicle Flotte übertragbar sind. Sollte dies der Fall sein, könnte man das dazugewonnene Know-How verwenden, um ein Assistenzsystem zu entwickeln, das den Fahrer rechtzeitig vor möglichen Schäden warnen könnte. Bei bisherigen Analysen wurden schon interessante Muster entdeckt, die nicht aus der Dokumentation von DTC erkennbar waren. Daneben fand man auch einige Muster, die aus logischer Sicht ersichtlich waren. Das war jedoch ebenfalls in Ordnung, da man somit erkennen konnte, dass die Verwendung der Assoziationsanalyse auf die Daten der richtige Ansatz war.

Als Beispielszenario für einen späteren Use Case könnte man sich vorstellen, dass man durch die Datenanalyse entdeckt hat, dass die DTCs P0300 und P0400 mit hoher Wahrscheinlichkeit mit dem noch gravierenderen DTC P0500 auftreten. Somit könnte man den Fah-

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

rer rechtzeitig daraufhin weisen, dass es zu einem noch kostspieligeren Problem kommen könnte, wenn dieser beim Auftreten der DTCs P0300 und P0400 nicht eine Werkstatt aufsucht.

Für die Datenanalyse wurden vor allem die Big Data Technologien Apache Spark und Apache Hive verwendet, da die Datenmengen nicht auf einem einzelnen Computer in einer akzeptablen Zeit verarbeitet werden konnten. Unter Apache Spark versteht man ein Cluster, der mehrere Computer miteinander bündelt und dadurch eine höhere Rechenleistung bietet. In Abbildung 2 sieht man den Aufbau eines Apache Spark Clusters.

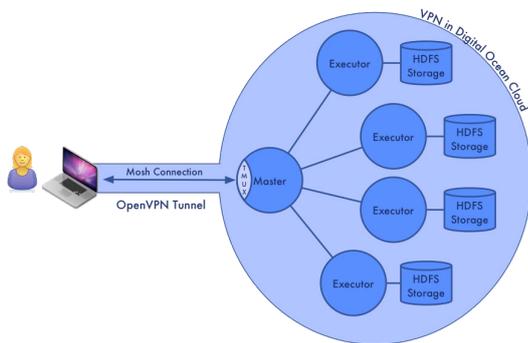


Abbildung 2: Apache Spark Cluster

Der Apache Spark Cluster besteht aus einem Master-Knoten, der die Aufgabe des Benutzers entgegennimmt und diese in mehrere Teilaufgaben an seine Executor (Slaves) verteilt. Am Ende der Teilaufgabenbearbeitung werden alle Ergebnisse zusammengefasst und ein Gesamtergebnis an den Benutzer geliefert. Dabei handelt es sich um das MapReduce-Prinzip, das von Google am Anfang des Jahrtausends entwickelt wurde.

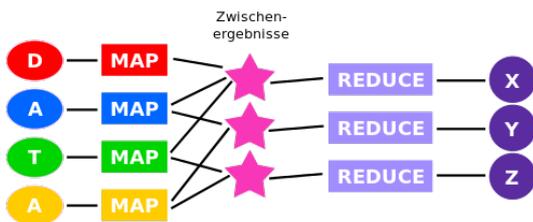


Abbildung 3: MapReduce Prinzip

In Abbildung 3 wird MapReduce demonstriert. Zuerst werden Teilaufgaben auf mehrere Clusterknoten verteilt, diese berechnen ihre Aufgaben und zum Schluss werden diese Ergebnisse zusammengeführt. Apache Hive ist wiederum ein Data-Warehouse für sehr große Datenmengen im Terrabyte-Bereich und größer. Hive wurde für die Analyse und Speicherung großer Datenmengen optimiert und ist nicht so sehr für das schnelle Hin- und Herschieben von Daten geeignet. In diesem Fall sollte man auf die bekannten Technologien, wie zum Beispiel dem Microsoft SQL Server zurückgreifen. Da es sich bei dem Apache Spark Cluster um eine relativ neue Technologie handelt und es in der aktuellen Literatur noch keine vergleichbaren Arbeiten gibt, die das Verhalten von Assoziationsanalyse-Algorithmen auf Apache Spark Clustern vergleicht, werden neben der eigentlichen Datenanalyse auch verschiedene Algorithmen dieser Art verglichen.

Hierbei werden, die im Laufe der Bachelorarbeit ermittelten und implementierten parallelen Assoziationsanalysealgorithmen für den Apache Spark Cluster namens MapReduce-Apriori Algorithmus und der parallele Frequent Pattern Algorithmus (PFP) betrachtet. Durch die bisherigen Analysen kam heraus, dass der Assoziationsanalysealgorithmus PFP um ein vielfaches performanter als die parallele Apriori Implementierung ist. Dies liegt vor allem daran, dass der PFP eine eigene Datenstruktur für die Abspeicherung der zu verarbeitenden häufigen Muster verwendet und somit nicht jedes Mal alle Transaktionen aufs Neue nach häufigen Mustern durchsuchen muss. Trotzdem lieferte der parallele Apriori Algorithmus eine Performancesteigerung gegenüber seiner sequentiellen Implementierung, die für riesige Datenmengen nicht geeignet ist und zu lange Laufzeiten für einen Datenanalysten aufweist.

[1] SBD, Automotive Big Data: What's It Really Worth? 2014.

Bildquellen:

- Abbildung 1: SBD, Automotive Big Data: What's It Really Worth? 2014.
- Abbildung 2: <http://www.infolace.com/images/2015-02-27-vpn-layout.png>
- Abbildung 3: <https://upload.wikimedia.org/wikipedia/commons/thumb/6/6c/MapReduce2.svg/500px-MapReduce2.svg.png>

Die Konzepte „Dependency Look-Up“ und „Dependency Injection“

Christoph Gschrey*, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Bei der Erstellung von objektorientierten Systemen können Abhängigkeiten zwischen den einzelnen Komponenten des Systems und zwischen dem System und seiner Umgebung entstehen. Diese wirken sich negativ auf die Evolvierbarkeit (Wandelbarkeit) des Programms aus. Ralf Westphal sagt zu Abhängigkeiten:

“Abhängigkeiten sind eine der größten Geißeln der Softwareentwicklung. Wenn es ein allgemein anerkanntes Prinzip gibt, dann ist es, Abhängigkeiten zu minimieren. Jede Hilfe ist da willkommen“ – Ralf Westphal [1]

Eine Abhängigkeit, die bei der Erstellung neuer Objekte durch andere Objekte entsteht, ist eine sogenannte «Create»-Abhängigkeit. Dabei erzeugt ein Objekt einer Klasse das benötigte Objekt einer anderen Klasse selbst. Dazu muss das erzeugende Objekt jedoch bereits zur Kompilierzeit die Klasse des zu erzeugenden Objekts kennen. Dadurch ist das erzeugende Objekt aber bereits von dem zu erzeugenden Objekt abhängig. [3]

Die folgende Abbildung stellt diesen Vorgang in UML-Notation dar:



Abbildung 1: Abhängigkeit durch das Erzeugen eines Objekts

Die Konzepte “Dependency Look-Up“ und “Dependency Injection“ ermöglichen es, «Create»-Abhängigkeiten abzuschwächen. Dabei verfolgen beide Konzepte die Strategie, die Kontrolle über die Erzeugung der Objekte, die von einer nutzenden Klasse benötigt werden, an eine eigens dafür vorgesehene Instanz zu delegieren.

Dependency Look-up

Bei Dependency Look-up sucht das nutzende Objekt das benötigte Objekt, mit dem es eine Verknüpfung herstellen will, anhand einer für das Objekt spezifischen Schlüssel-

information. Bei dieser Information kann es sich zum Beispiel um den Namen des Objekts oder um ein Interface seiner Klasse handeln. Damit das nutzende Objekt das benötigte Objekt verwenden kann (z. B. Zugriff auf dessen Dienste bekommt), benötigt es des Weiteren das Wissen darüber, wie das benötigte Objekt verwendet werden kann. Dies wird durch die Einführung einer Abstraktion der Klasse des benötigten Objekts als Schnittstelle oder abstrakte Klasse im Programm gewährleistet. Die Klasse des benötigten Objekts muss den Vertrag dieser Schnittstelle oder abstrakten Klasse erfüllen. Auch die nutzende Klasse muss diesen Vertrag kennen und einhalten, um das benötigte Objekt verwenden zu können.

Der Vorteil ist, dass das suchende Objekt nicht mehr die konkrete Klasse des benötigten Objekts kennen muss. Durch die Entkopplung der beiden Objekte voneinander steigt die Flexibilität bei der Herstellung von Verknüpfungen, beispielsweise bei Objekten, die von Frameworks generiert und geliefert werden, und weitere Verknüpfungen mit anderen Objekten benötigen.

Bei der Umsetzung des “Dependency Look-up“ wird häufig eine sogenannte Registratur verwendet. Hierbei wird eine zentrale Klasse eingeführt, die mithilfe einer Hashmap oder einem Dictionary Schlüssel-Wert-Paare speichern kann. Diese Klasse wird so implementiert, dass sie global verwendet werden kann und von ihr nicht mehrere Instanzen gebildet werden können, damit nicht verschiedene Objekte in verschiedenen Instanzen unter demselben Schlüssel registriert werden können. Dies würde sonst zu Redundanzen und Widersprüchen führen. Hierfür bietet sich die Umsetzung als zentrale Klasse, die nur statische Member hat, oder eine Implementierung nach dem Singleton-Pattern an, da von einer Singleton-Klasse nur eine einzige Instanz erzeugt und verwendet werden kann.

Eine Registratur kann benötigte Objekte unter Verwendung einer Schlüsselinformation registrieren. Benötigt ein Objekt einer nutzenden Klasse ein bestimmtes Objekt, übermittelt es

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

die Schlüsselinformation an die Registratur. Anschließend wird anhand der Schlüsselinformation (meist der Name des Objekts oder das Interface des benötigten Objekts), unter der das Objekt registriert wurde, nach dem betreffenden Objekt gesucht. Wird das Objekt gefunden, liefert die Registratur eine Referenz auf das gesuchte Objekt an das suchende Objekt zurück.

Eine Registratur kann auch weiterentwickelt werden, um komfortablere Dienste anbieten zu können, die dem Entwickler oder den suchenden Objekten weitere Vorteile verschaffen. Die Konfiguration der Registratur kann beispielsweise an externe Konfigurationsdateien ausgelagert werden, sodass die Konfiguration nicht im Quellcode vorgenommen werden muss.

Dependency Injection

Das Konzept "Dependency Injection" wurde erstmals von Martin Fowler definiert [2]. "Dependency Injection" dient als Konzept genauso wie "Dependency Look-Up" dazu, die Kopplungen zwischen Objekten abzuschwächen. Diese engen Kopplungen entstehen dadurch, dass die Objekte oft zu viele Informationen über andere Objekte haben müssen, um mit ihnen eine Verknüpfung einzugehen. "Dependency Injection" löst dieses Problem, indem es die Erstellung und Verwaltung von Objekten und deren Verknüpfungen an eine extra dafür vorgesehene Instanz delegiert. Diese Instanz, die oft auch als **Injektor** bezeichnet wird, steuert zur Laufzeit die Generierung von einzelnen Objekten und sorgt für deren Verknüpfungen untereinander.

Das Problem der «Create»-Abhängigkeiten innerhalb eines Programms wird zwar auch hierbei nicht vollständig behoben, da der Injektor die Klassen der Objekte, die er erzeugt, kennen muss und damit von diesen abhängig ist. Dennoch wird das Problem durch dieses Konzept zumindest abgeschwächt, da nun die enge Kopplung in Form der «Create»-Abhängigkeiten zwischen den Objekten nicht mehr vorhanden ist. Den Vorgang, einem nutzenden Objekt das von ihm benötigte Objekt über eine außenstehende Instanz zur Verfü-

gung zu stellen, bezeichnet man als Injektion. Anders als bei "Dependency Look-Up", wo das suchende Objekt zusätzlich die Schlüsselinformation zum Auffinden des benötigten Objektes haben muss, kennt bei der "Dependency Injection" das nutzende Objekt nur die Abstraktion der Klasse des benötigten Objekts. Der Injektor hält die benötigten Informationen und übernimmt die Kontrolle von außen. Damit wird zur Laufzeit sichergestellt, dass jedes Objekt, welches ein anderes Objekt benötigt, dieses vom Injektor zur Verfügung gestellt bekommt. Ohne die Injektion der benötigten Abhängigkeiten durch den Injektor ist das abhängige Objekt in diesem Konzept nicht lauffähig. Dabei muss die nutzende Klasse den Vertrag der Klasse des benötigten Objekts kennen. Dieser Vertrag muss bei "Dependency Injection" eingehalten werden.

Für die Umsetzung von "Dependency Injection" gibt es nach Martin Fowler drei verschiedene Möglichkeiten [2]:

- **"Constructor Injection"**
Die Klasse des abhängigen Objekts definiert einen Konstruktor, über den bei der Erstellung des Objekts direkt die Abhängigkeit(en) durch den Injektor injiziert werden kann/können.
- **"Interface Injection"**
Die Klasse des abhängigen Objekts implementiert ein Interface, welches eine bestimmte Methode für die Injizierung definiert, die dem Injektor zur Verfügung steht.
- **"Setter Injection"**
Die Klasse des abhängigen Objekts definiert eine separate set-Methode, welche die benötigten Objekte übergeben bekommt.

Fazit

Bei den Konzepten "Dependency Look-Up" und "Dependency Injection" handelt es sich um wichtige Ansätze, mit denen Abhängigkeiten bei der Erzeugung von Objekten abgeschwächt werden können, wodurch eine größere Flexibilität in einem Softwaresystem erreicht werden kann.

-
- [1] Westphal, Ralf (07.09.2010) "Abhängigkeiten bewusster wahrnehmen": <http://blog.ralfw.de/2010/09/abhaengigkeiten-bewusster-wahrnehmen.html> (Zugriff am 20.10.2016)
 [2] Fowler, Martin (2014) "Injection": <http://martinfowler.com/articles/injection.html> (Zugriff am 20.10.2016)
 [3] Goll, Joachim (2012) "Methoden des Software Engineering"
 Wiesbaden: Springer Vieweg, ISBN 978-3-8348-2433-2

Bildquellen:

- Abbildung 1: Eigener Entwurf

Entwicklung eines webbasierten Shopfloor Management Modul innerhalb der Planungs- und Steuerungssoftware 3Liter-PPS®

Timo Manuel Gutierrez*, Dirk Hesse, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Der andauernde Trend weg von Massenproduktionen hin zu individuellen Produkten bringt einen tiefgreifenden Wandel mit sich. Kleine Losgrößen und vielfältige Produkteigenschaften fordern eine flexible Organisation von produzierenden Unternehmen. Dieser Herausforderung wird mit einer schlanken Fertigung und Dezentralisierung begegnet [1]. Die Teams arbeiten autonom in den Werkstätten.

Diese Philosophie findet sich auch im Planungsansatz PIT-Produzieren im Takt® wieder. Herkömmliche Planungsansätze versuchen, trotz komplexer Strukturen in der Fertigung jeden einzelnen Arbeitsschritt bis auf die Minute genau zu terminieren. Dies nimmt den Teams in den Werkstätten jegliche Flexibilität und weicht somit in den meisten Fällen stark von der Realität ab. Bei PIT-Produzieren im Takt® hingegen werden die Kapazitäten der Teams in Takte eingeteilt. Innerhalb eines Taktes werden die Arbeiten dezentral organisiert. Der Ansatz wird von der Planungs- und Steuerungssoftware 3Liter-PPS® gestützt.

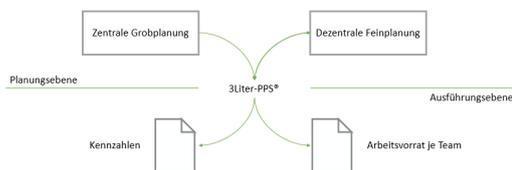


Abbildung 1: Shopfloor Management Komponenten im 3Liter-PPS®

Das Tool sorgt durch kapazitätsgeprüfte Planung und Visualisierung von Engpässen für ein realistischeres Bild der Produktion. Darüber hinaus übernehmen die Verantwortlichen der Fertigungsteams mit dem 3Liter-PPS® selbst die Feinplanung ihrer Arbeit. Somit sind sie innerhalb eines Taktes flexibel und können eigenverantwortlich auf ihr wechselndes Umfeld reagieren.

Aufgabe dieser Bachelorarbeit ist es, innerhalb der Planungs- und Steuerungssoftware 3Liter-PPS®, ein Modul zu entwickeln, das die Aufgaben eines Shopfloor Management Boards übernehmen kann und Mitarbeiter, Teams so-

wie Führungskräfte bei ihrer täglichen Arbeit optimal unterstützt.

Das Shopfloor Management Board ist integraler Bestandteil der dezentralen Organisation in der Fertigung. Dort werden die arbeitsspezifischen Informationen visualisiert und somit kommuniziert. Für jedes Team ist dort ersichtlich, wo deren Aufgaben und Ziele liegen. Des Weiteren werden Kennzahlen aufgezeigt, die die derzeitige Leistung in Zusammenhang mit einem definierten Soll-Zustand setzt. Diese Informationen sollen so zur Verfügung stehen, dass sie bestmöglich zur eigenständigen Problemlösung beitragen.

Shopfloor Management Komponenten im 3Liter-PPS®

Das Shopfloor Management befähigt die Fertigungsteams zum autonomen Arbeiten. In Abbildung 1 sieht man auf welche Art Shopfloor Management Komponenten bisher verwendet werden.

Die Arbeitsvorratslisten zeigen für eine Takt, die vorhandene Arbeit an. Das heißt sie vermitteln einem, bezüglich eines Arbeitsganges Informationen über Endtermine und Mengen. Diese Listen sorgen somit dafür, dass die Mitarbeiter wissen welche Arbeiten innerhalb des Taktes geplant sind. Des Weiteren werden Kennzahlen generiert. Über Parameter kann man Kriterien für Aggregation oder Takte festlegen. Die erstellten Reports können dann als schnelle Informationsquelle genutzt werden. Relevante Kennzahlen für die Produktion sind zum Beispiel Auslastung, Termintreue, Rückstand oder Abarbeitungsgrad.

Das Shopfloor Management Modul

Aufgrund wachsender digitaler Vernetzung steigt die Verwendung mobiler Endgeräte auch in den Fertigungshallen. Diese Entwicklung soll durch das webbasierte Shopfloor Management Modul genutzt werden, um die dezentral organisierten Teams noch besser in ihrer Arbeit zu unterstützen. Das Modul ist also eine Art evolutionäre Folge der bestehenden Kom-

*Diese Arbeit wurde durchgeführt bei der Firma LF Consult GmbH, Stuttgart

ponenten. Durch moderne Konzepte soll damit eine leichter zugängliche und auch interaktivere Anwendung geschaffen werden. Mit dem neuen Modul bekommt der Anwender die Möglichkeit, sich selbst alle benötigten Kennzahlen und Listen auf einer Seite, als eine Art Dashboard zusammenzustellen. Diese Boards können nach Belieben konfiguriert, gespeichert und geladen werden. Mit Konfigurationen sind hier im Allgemeinen Filterbedingungen gemeint, wie zum Beispiel die Begrenzung der Daten auf einen Zeitraum oder einzelne Ressourcen.

Darüberhinaus finden Interaktionen des OLAP wie zum Beispiel „Drill-Down“, die man von Self-Service Business Intelligence Software kennt, hier ihre Anwendung. Dabei geht es darum sich als Nutzer mit wenigen Aktionen durch die Dimensionen der dargestellten Daten bewegen zu können [2]. Für Kennzahlen im Kontext der Produktion wäre das zum einen die zeitliche Dimension mit den Ausprägungen Quartal, Monat, Woche oder Tag. Zum anderen aber auch die Dimension der zu betrachtenden Einheit, diese reichen von einer ganzen Organisationseinheit bis zu einem einzelnen Mitarbeiter oder einer Maschine. Der Nutzer kann somit die Auswahl der Daten spezifisch für seinen Anwendungsfall anpassen. Ausserdem ist es möglich einzelne Komponenten eines Boards miteinander zu „verknüpfen“. Das heißt wenn man beispielsweise einen Datenpunkt in einem Diagramm auswählt, werden in der dazu-

gehörigen „verknüpften“ Liste nur die Arbeitsgänge des ausgewählten Datenpunktes angezeigt. Diese Funktion sorgt dafür, dass man sich auch ohne Änderungen der Konfigurationen einen detaillierteren Einblick verschaffen kann. Wie das Shopfloor Management Modul bedient wird ist letztendlich abhängig von der Aufgabe des Nutzers. Während ein Produktionsleiter Kennzahlen für größere Einheiten und Zeiträume einsehen möchte, wird ein Teamleiter eher am Arbeitsvorrat für einzelne Tage oder Wochen interessiert sein.

Da zeitlich parallel zur Nutzung des Moduls mit den 3Liter-PPS®Clients gearbeitet wird und sich somit die Daten durch Einbeziehungswise Umplanungen kontinuierlich ändern, gibt es im Hintergrund eine Event-Ebene, die per Push-Events für die Aktualität der Daten sorgt.

Ausblick

Die genannten Funktionen und Eigenschaften, bilden den Grundstein des Shopfloor Management Moduls. Für die Zukunft sind weitere Funktionalitäten vorgesehen. Diese werden die reine Visualisierung durch aktive planerische Tätigkeiten ergänzen. Desweiteren sind weitere Webanwendungen geplant, die sich in die bestehende Struktur einbetten werden. Dabei geht es vor allem um Alternativen zu den bestehenden 3Liter-PPS®Clients.

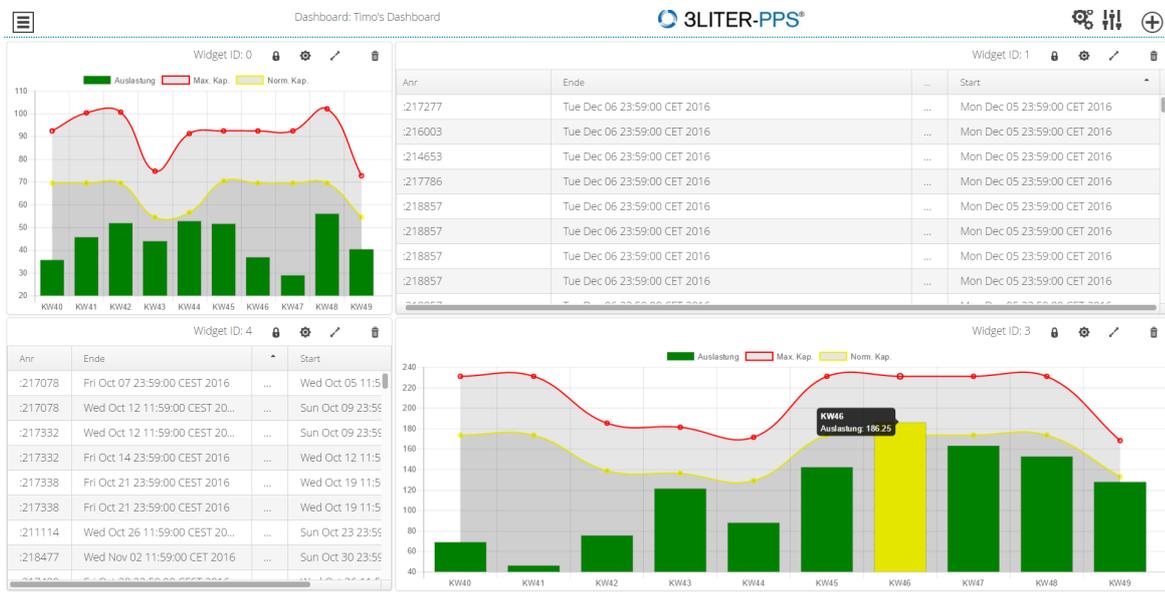


Abbildung 2: Demo des Moduls

- [1] Westkämper, Engelbert; Zahn, Erich (2009): Wandlungsfähige Produktionsunternehmen. Das Stuttgarter Unternehmensmodell. Berlin, Heidelberg: Springer-Verlag.
 [2] Alpar, Paul; Schulz, Michael (2016): Self-Service Business Intelligence. In: Bus Inf Syst Eng 58 (2), S. 151-155

Bildquellen:

- Abbildung 1-2: Eigene Darstellung

Konzipierung und Realisierung eines Analysewerkzeugs für das Monitoring von Telegrammen der synchron-seriellen Schnittstelle SSI im Besonderen unter Aspekten der Funktionssicherheit für Positioniersysteme der Automatisierungstechnik

Felix Heil*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Die Aufgabenstellung der vorliegenden Arbeit war die Entwicklung eines echtzeitfähigen Analysewerkzeugs für SSI-Schnittstellen. Das Werkzeug soll übertragene Positionsdaten zweier SSI-Kanäle jeweils in sich und gegeneinander kontinuierlich auf Fehler überprüfen und im Fehlerfall eine Speicherung der fehlerhaften Daten ermöglichen. Ein Beispielaufbau mit zwei verschiedenen Mastern und zwei Slaves ist in Abbildung 1 veranschaulicht.

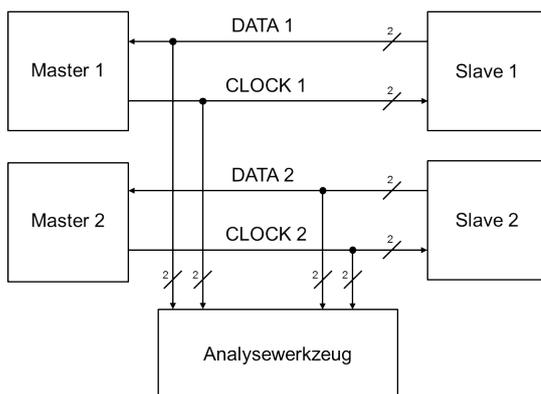


Abbildung 1: Beispielaufbau (2 Master, 2 Slaves)

Das Analysewerkzeug soll in den zu überprüfenden SSI-Signalen zu lange oder zu kurze Taktpausen und Übertragungsperioden, Verletzungen der Monoflop-Zeiten und zu wenige oder zu viele gesendete Datenbits erkennen. Des Weiteren soll der Positionswert pro SSI-Kanal auf ungültige Positionswertsprünge überprüft werden und durch einen Quervergleich der Positionsdaten beider SSI-Kanäle Fehler aufgedeckt werden, wenn die Differenz dieser Positionsdaten nicht innerhalb eines einstellbaren Toleranzbandes liegt.

Momentan am Markt verfügbare Geräte können diese Funktionalitäten nicht zufriedenstellend erfüllen. Einige benötigen außerdem eine ständige PC-Anbindung, welche den

Einsatz im Industrie-Umfeld erschwert, da sich die zu überprüfenden Leitungen an beweglichen Aufbauten befinden können. Somit entstand die Notwendigkeit für ein kompaktes Embedded-Gerät, welches durchgehend Daten überprüfen und aufzeichnen kann.

Darüber hinaus war von Beginn an zu berücksichtigen, dass das Gerät konfigurierbar sein sollte. Diese Anforderung an das Gerät beruht darauf, dass beide SSI-Kanäle komplett unterschiedlich konfiguriert sein können. Entsprechend wurden von Anfang an Design-Entscheidungen getroffen, bei denen auf fest definierte Werte, ausgenommen Maximal- und Minimal-Werte, verzichtet wurde. Zusätzlich zu den fehlerhaften Daten sollen auch eine einstellbare Anzahl an SSI-Telegrammen vor und nach dem Auftreten des erkannten Fehlers in den SSI-Telegrammen mit aufgezeichnet und gespeichert werden.

Die Arbeit wurde mittels eines Evaluationsboards mit bereits im Unternehmen bekannter Hardware durchgeführt. Schon vorhandene SSI-Schnittstellen und zugehörige Software erleichterten die Inbetriebnahme der SSI-Funktionalität. Die Software-Implementierung wurde mittels C++ und einem firmeneigenen Komponenten-Framework realisiert, welches erforderliche Echtzeit-Betriebssystem-Anforderungen erfüllt. Beim Code-Entwurf wurde darauf geachtet, die Anzahl von Transferbefehlen möglichst gering zu halten und hauptsächlich verzeigerte Strukturen zu nutzen. Assembler-Code konnte generell vermieden werden, da sich nach kurzer Zeit herausgestellt hat, dass das System sehr performant ist und alle Aufgaben innerhalb der jeweiligen Zeitanforderungen abgearbeitet werden können.

*Diese Arbeit wurde durchgeführt bei der Firma Leuze electronic GmbH + Co. KG, Owen

Damit eine zyklische Auswertung der gepufferten Daten möglich ist, wurde eine Triple-Puffer-Struktur verwendet, welche nach dem Übergang des Schreib-Zeigers eines Pufferelements auf das nächste Pufferelement einer übergeordneten Manager-Komponente signalisiert, dass das entsprechende Pufferelement nun gelesen und ausgewertet werden kann. Die Auswertung und Erkennung der Fehler wird entsprechend der vorher konfigurierten Werte vorgenommen. Für eine zweikanalige Auswertung muss mehr Aufwand betrieben werden.

Der vereinfachte, konzeptionelle Ablauf ist in Abbildung 2 zu sehen. Es ist an dieser Stelle anzumerken, dass in der Implementierung die „Idle“-Zeit des Monitor-Managers einen wesentlich höheren Anteil besitzt. Die Zeitachse in Abbildung 2 spiegelt keine realen Werte wieder, und die Fehlerfallbehandlung ist sehr vereinfacht dargestellt. Der Monitor-Manager ist die Hauptkomponente, welche Nachrichten empfängt, verarbeitet und weitergibt.

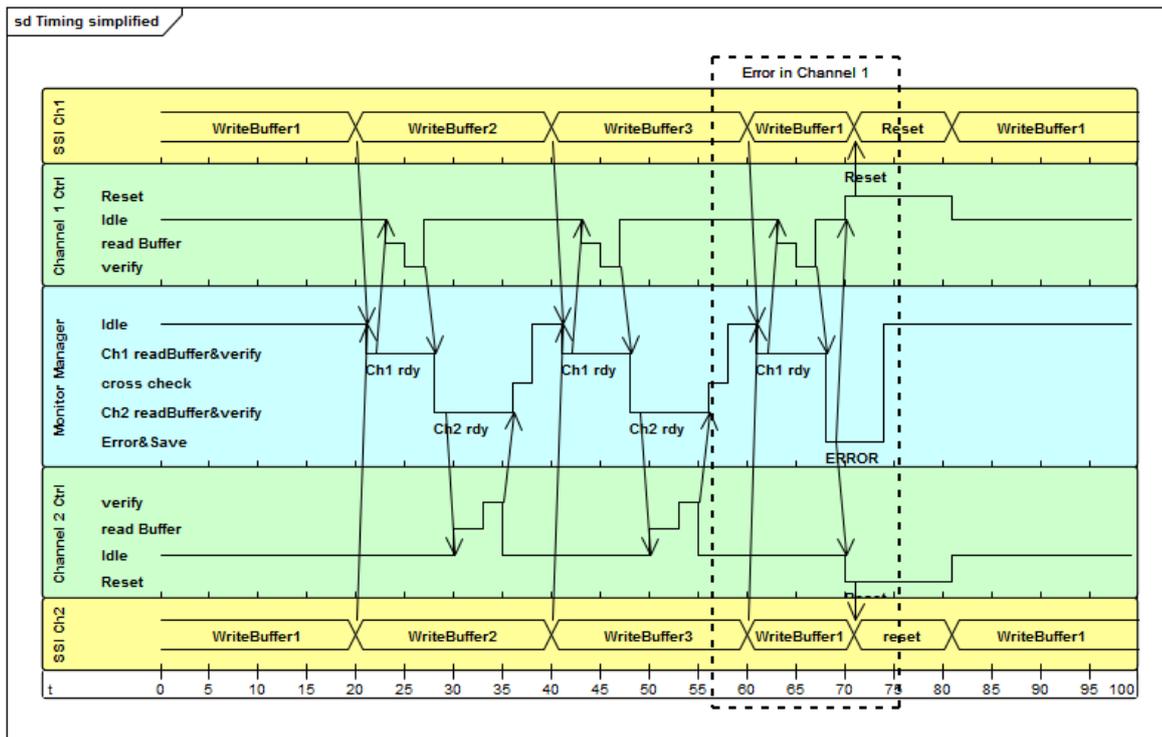


Abbildung 2: Konzeptionelle Funktionalität mit Fehlerfall

Die Auslesung pro Kanal funktioniert prinzipiell weiterhin wie oben beschrieben, jedoch ist es möglich, dass beide Kanäle komplett asynchron laufen. Würde man die Puffer beider Kanäle nicht miteinander in Beziehung bringen, würden diese zu unterschiedlichen Zeiten volllaufen und somit nicht mit den zueinander passenden SSI-Telegrammen belegt sein. Zum Beherrschen dieser Situation wurde eine Kalibrierungsphase direkt nach dem Start des Analysewerkzeugs implementiert. Diese nutzt Zeitstempel und die Anzahl der Einträge pro Pufferelement und errechnet daraus einen Faktor, mit dem der „schnellere“ Puffer vergrößert wird, damit sich dieser mit dem „langsamen“ Puffer synchronisiert. Jedoch reicht dies für eine dauerhafte Synchronisation der Kanäle nicht vollständig aus.

Um die Synchronität weiterhin sicher zu stellen, wurde eine Anpassung der Puffergrößen zur Laufzeit implementiert, welche der Vorgehensweise in der Kalibrierungsphase ähnelt. Damit wird sichergestellt, dass immer

jedes einzelne SSI-Telegramm eines Kanals ausgewertet und ebenfalls mit einem entsprechenden Telegramm des anderen Kanals in Beziehung gesetzt wird und somit verglichen werden kann. Die Speicherung der Daten geschieht mittels USB-Stick in einem mit üblicher Tabellenkalkulations-Software auswertbaren Datenformat.

Ein weiteres implementiertes Feature ist eine Trigger-LED, welche einen aufgetretenen Fehler signalisiert. Parallel wird das Trigger-Signal auf einen Trigger-Ausgang gelegt, der an ein Oszilloskop angeschlossen werden kann. Für die inhaltliche Auswertung der Positionsdaten im SSI-Datenstrom können ein Bit-Offset, die Länge in Bits und die Auflösung konfiguriert werden. Zusätzlich kann das Werkzeug sowohl normale Binärals auch Gray-kodierte Positionsdaten auswerten. Für die zweikanalige Auswertung kann ein Positions-Offset zwischen den zu analysierenden Geräten konfiguriert werden. Diese Konfigurationsmöglichkeiten müssen gegeben

sein, da viele am Markt befindliche Geräte diese ebenfalls bieten und nutzen. [1]

Die Arbeitsleistung der Implementierung entspricht den Erwartungen. Die Auswertung und der Quervergleich von 50 Telegrammen mit 25 Datenbits pro Kanal in einem synchronen Aufbau mit einer Taktrate von 1 MHz und kleinstmöglichen Taktpausen dauert ca. 1,19ms. Dies entspricht grob der Dauer von

24 SSI-Telegrammen auf der Leitung in dieser Umgebung. Der asynchrone Aufbau kann massive Abweichungen der Feldparameter aufweisen. Telegramme können auf einem Kanal mehrere Millisekunden auseinander liegen, während der andere Kanal innerhalb von 100 μ s ein neues Telegramm schickt. Auch hier konnte das Analysewerkzeug seinen Zweck erfüllen und somit seine angeforderte Funktionalität garantieren.

[1] Leuze electronic GmbH + Co. KG (Hrsg.): BPS 307i Barcode-Positionier-System Original-Betriebsanleitung. Leuze electronic GmbH + Co. KG, 2015

Bildquellen:

- Abbildung 1-2: Eigene Darstellung

Konzeption, Realisierung und Qualifizierung eines Moduls zur Nachbildung des Gleichspannungszwischenkreises von Elektrofahrzeugen unter Verwendung der Curve Fitting Toolbox von Matlab

Patrice Heller*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Die Elektrifizierung von Fahrzeugen für den Straßenverkehr ist ein zentrales Thema der aktuellen Automobilentwicklung, so dass der Smart Electric Drive (ED) im Konzept von Car2Go parallel zum Normalangebot großen Absatz findet. Die Batterie stellt einen wichtigen Teil des HV-Systems (High Voltage) dar. Um ein sicheres Abstellen des Fahrzeuges zu gewährleisten, wird der gesamte Zwischenkreis allpolig von der Batterie getrennt, ein Wiedereinschalten der Batterie muss kontrolliert erfolgen.

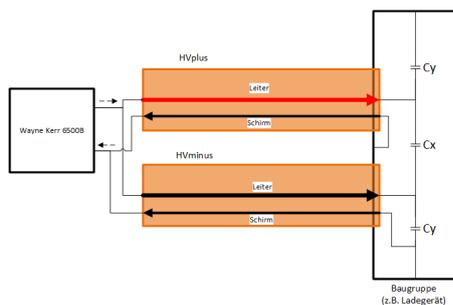


Abbildung 1: Messprinzip Common Mode

Aus Gründen der elektromagnetischen Verträglichkeit (EMV) während des Betriebs sind Filterelemente und Komponenten zur Spannungsstabilisierung im Zwischenkreis in Form von sogenannten Cx-, und Cy-Kapazitäten verbaut. Durch diese Baugruppen entstehen stromtragende Pfade, deren Ströme Auswirkungen auf die EMV-Eigenschaften im Zuschaltmoment haben können. Bestrebungen, EMV-Eigenschaften mit gleichem Verhalten unter allen Betriebsbedingungen zu erreichen, machen umfassende Prüfabläufe im Vorfeld erforderlich. Daher soll für Tests der Batterie bezüglich der Aspekte der parasitären Kapazitäten und Induktivitäten der Zwischenkreises nachgebildet werden können.

Alle Hochspannungskabelverbindungen, die die Batterie ab den HV-Schützen mit dem Umrichter, dem Innenraumheizer, dem elektrischen Kältemittelverdichter (EKMV) und dem

Ladegerät verbinden, bilden wie in Abb. 2 dargestellt zusammen mit den jeweiligen Cx- und Cy-Kapazitäten der einzelnen Baugruppen den Hochspannungszwischenkreis.

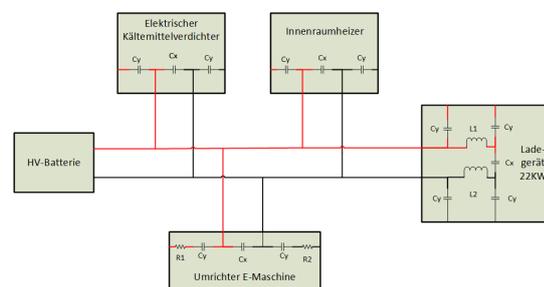


Abbildung 2: Gesamtopologie des HV-Zwischenkreises

Im Vordergrund der Arbeit standen Konzeption, Realisierung und Qualifizierung eines Moduls zur Nachbildung des Gleichspannungszwischenkreises von Elektrofahrzeugen.

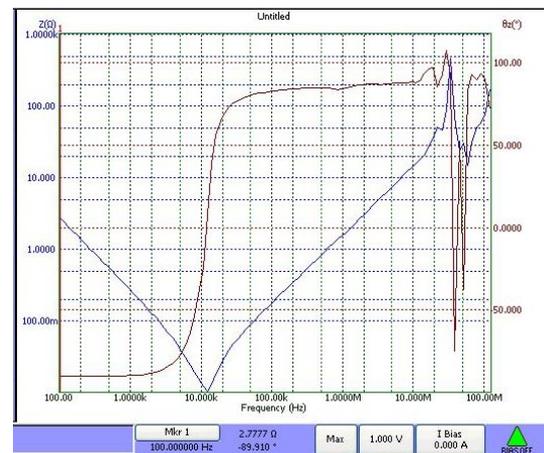


Abbildung 3: Typischer Impedanzverlauf im Differential Mode

Dazu wurden im ersten Schritt die Probleme analysiert, die beim Zuschalten der HV-Batterie in den Zwischenkreis des Elektrofahrzeuges durch diskrete und parasitäre Kapa-

*Diese Arbeit wurde durchgeführt bei der Firma Deutsche ACCUotive GmbH & Co. KG, Kirchheim unter Teck (Nabern)

zitäten und Induktivitäten hervorgerufen werden. Durch Vermessen der einzelnen Baugruppen und dem Gesamtaufbau des HV-Systems konnten die jeweiligen Frequenz-, und Phasengänge bestimmt werden. Im weiteren Verlauf wurden aus diesen Messdaten durch geeignete Interpretation zunächst Ersatzschaltbilder entworfen. Zur möglichst genauen Bestimmung der Bauteilwerte des Simulationsmoduls wurden diese mit Hilfe der Curve-Fitting-Methode an die Impedanzverläufe der realen Komponenten angeglichen [1]. Aus den Erkenntnissen dieser Arbeiten wurden im Anschluss die einzelnen Ersatzschaltbilder zunächst vereinfacht und dann ein gemeinsamer Schaltplan entworfen.

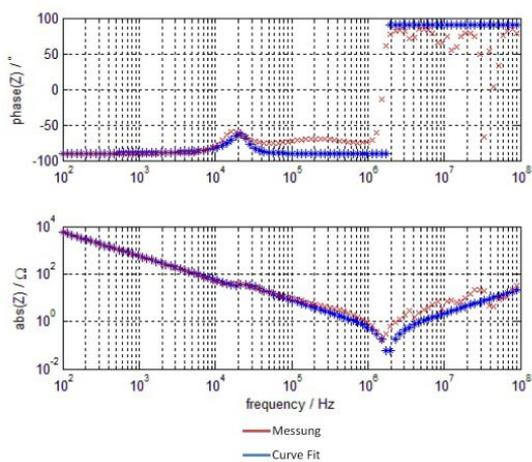


Abbildung 4: Messkurve mit Curve Fitting

Mit den gewonnenen Daten wurden anschließend entsprechende Bauteile beschafft, die die Darstellung jeder einzelnen Baugruppe im Gesamtverbund stufenweise mit Abweichungen von bis zu ca. 200% ermöglicht.

Im Anschluss an den Aufbau wurde das Simulationsmodul für die zu automatisierenden Prüfbläufe vorbereitet. Dazu wurden

je ein Sensor- und Aktormodul mit CAN-Interface integriert. Ersteres dient zur Feststellung des Ladezustands des Zwischenkreises mittels Spannungsmessung, zweiteres ist mit Relais bestückt und wird zur kontrollierten Entladung der installierten Kapazitäten über Leistungswiderstände verwendet. Durch die Programmierung eines CAPL-Knotens unter Verwendung des Tools CANoe wurde ein Demonstrationsablauf zur Integration in einen Prüfstand erstellt.

Im Anschluss daran wurden verifizierende Messungen durchgeführt. Sie dienen der Überprüfung, ob die Nachbildung des Zwischenkreises nahe an dem Verhalten des realen Baugruppenverbundes liegt. Insgesamt konnte mit der vorliegenden Arbeit gezeigt werden, dass die realistische Nachbildung des Zwischenkreises durch die geeignete Ablaufkette aus der Impedanzvermessung von realen Baugruppen, dem Interpretieren von Ersatzschaltbildern und dem Entwickeln eines Ersatzschaltbildes mittels der Curve-Fitting-Methode von Matlab mit anschließendem Aufbau eines Simulationsmoduls möglich ist.

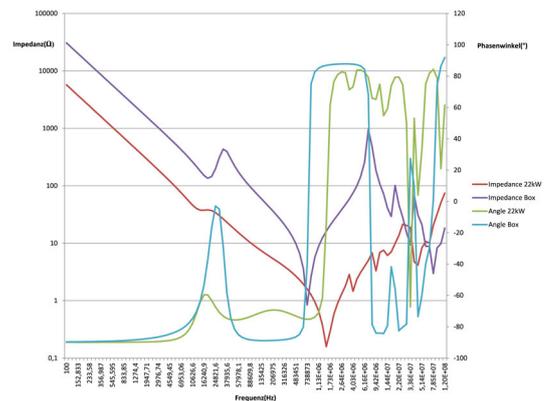


Abbildung 5: Vergleichsmessung von realen Baugruppen und Simulationsmodul

[1] The Mathworks, Curve Fitting Toolbox, 2016, <https://de.mathworks.com/products/curvefitting/>

Bildquellen:

- Abbildung 1: Eigene Messung
- Abbildung 2: Eigener Entwurf
- Abbildung 3: Eigener Entwurf
- Abbildung 4: Eigene Curve Fitting Anwendung
- Abbildung 5: Eigener Vergleichsentwurf

Agiles Testen

Oliver Hummel*, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Kontinuierlich und nicht nachgelagert wie in der spezifikationsorientierten Entwicklung zu testen, ist ein Charakteristikum der agilen Entwicklung. Agil muss kontinuierlich getestet werden, da zyklisch potenziell auslieferfähige Software erzeugt wird.

Eine häufig genutzte Form der agilen Entwicklung ist die Entwicklung mit **Scrum**. Der Kern dieser Entwicklungsmethodik ist der Sprint. Während einer Zeitdauer von zwei bis vier Wochen werden Aufgaben abgearbeitet, die sich das Team vor jedem Sprint selbst zuteilt. Falls nicht alle Aufgaben in einem Sprint erledigt werden können, werden sie in den nächsten Sprint verschoben und das Team kann noch einmal darüber nachdenken, ob es das Arbeitspensum zu hoch angesetzt hat. Das Hauptziel von Scrum ist es, dem Kunden nach jedem Sprint ein potenziell lauffähiges Produkt zu liefern. Um dies zu gewährleisten, gehört das Testen bei Scrum bereits in jeden Sprint eingebaut und darf nicht erst nachträglich durchgeführt werden. Das Testen muss also kontinuierlich erfolgen.

Organisation des agilen Testens

Der Vorgang des Testens wurde allerdings sowohl im agilen Manifest als auch im Scrum-Guide nicht beachtet. Generell können Testvorgänge in eine **innere** und eine **äußere Sicht** des Systems eingeteilt werden. Bei der inneren Sicht steht der gesamte Code als White-Box zur Einsicht bereit. Dies ist die Sicht des Entwicklers. In der inneren Sicht werden Unit Tests und Integrationstests durchgeführt.

Die äußere Sicht des Testens beschreibt die Sicht der Kunden und der Anwender. Der Code ist dabei eine Black-Box und kann nicht eingesehen werden. Zu dieser Sicht gehören Akzeptanztests, Performance Tests, Usability Tests, Lasttests, Security Tests sowie Explorative Testen. Oftmals werden diese Tests von externen Spezialisten durchgeführt.

Da ein agiles System laufend geändert wird, müssen dieselben Tests immer wieder in Form von **Regressionstests** ausgeführt werden. Hierbei wird überprüft, ob die Software trotz Änderungen noch lauffähig ist und keine Fehler infolge von Änderungen eingeführt oder freigelegt wurden. Aufgrund der vielen,

wiederholten Regressionstests einer iterativen, inkrementellen Entwicklung lohnt sich bei einer agilen Entwicklung eine Automatisierung der Tests bereits zu einem viel früheren Zeitpunkt als beim nachgelagerten Testen.

Eine zentrale Rolle in einem agilen Testprozess spielt der teaminterne **Testkoordinator**. Er gilt als Vertreter der Qualität und sorgt für die Einhaltung der vereinbarten Qualitätskriterien sowie für das Gleichgewicht zwischen Produktivität und Qualität. Zu seinem Aufgabenbereich gehören die regelmäßige Refaktorisierung und Nachdokumentierung sowie die Beseitigung von Mängeln im Code [1].

Sollen mehrere agile Projekte nebeneinander herlaufen, wird empfohlen, ein **Test Competence Center** einzurichten, um die verschiedenen Projekte in Fragen des Tests zu Methoden, Techniken und Werkzeugen zu unterstützen. Üblicherweise deckt das Test Competence Center Load-, Performance-, Security-, Safety-, Maintenance-, Usability- und Portabilitytests ab.

Fundamentaler Testprozess nach ISTQB

Das International Software Testing Qualifications Board (ISTQB) veröffentlichte einen fundamentalen spezifikationsorientierten Testprozess als Versuch, den Testvorgang weltweit zu standardisieren. Die Gliederung dieses Prozesses in fünf Hauptaktivitäten ist für agile Projekte allerdings hinderlich, da sie eine Rollenaufteilung und einen Prozess impliziert, die es beim agilen Vorgehen so nicht gibt. Die Hauptaktivitäten nach dem ursprünglichen, spezifikationsorientierten Vorgehen von ISTQB sind

- Testplanung und Steuerung,
- Testanalyse und Entwurf,
- Testrealisierung und Durchführung,
- Auswertung und Bericht sowie
- Abschluss der Testaktivitäten [2].

Passt man die Reihenfolge und Verantwortlichkeiten der Inhalte dieser Aktivitäten etwas an, können sie weitgehend auch bei agilen Projekten verwendet werden. Geisen und Güldali [3] schlagen daher vor, die Testplanung während des Sprint Planning und des Daily Scrums

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

durchzuführen. Die Testanalyse, der Testentwurf wie auch die Testimplementierung sollen während des Sprints erfolgen. Die Testauswertung soll im Sprint Review und der Testabschluss bei der Sprint Retrospektive durchgeführt werden.

Die vier Testquadranten des agilen Testens

Lisa Crispin und Janet Gregory haben die Bedeutung des agilen Testens begründet. Mit Hilfe ihrer vier Testquadranten [4] können Softwaretests im agilen Umfeld in vier Bereiche eingeteilt werden. Die Klassifizierung erfolgt mit Hilfe der Kriterien „fachlich orientiert“, „technisch orientiert“, „teamunterstützend“ oder „produkt hinterfragend“. Diese werden als Matrix mit vier Quadranten dargestellt, wie die folgende Abbildung zeigt:

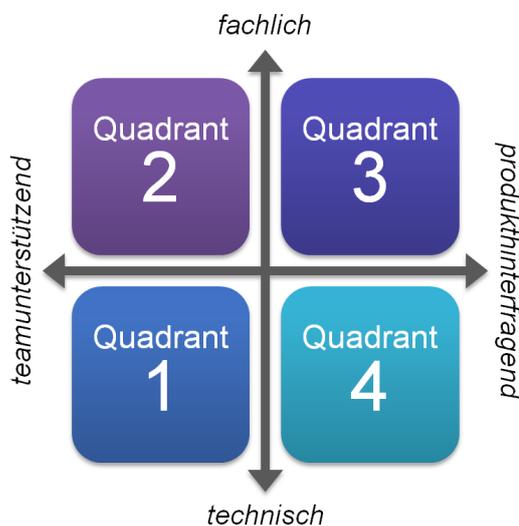


Abbildung 1: Die vier Testquadranten des agilen Testens

Quadrant 1 ist technisch orientiert und teamunterstützend. Darin werden Tests zusammengefasst, die schon in der frühen Phase der Entwicklung durchgeführt werden. Dazu zählen automatisierte Tests wie Unit-Tests sowie Integrationstests, um die korrekte Zusammenarbeit von Teilsystemen sicherzustellen.

Quadrant 2 ist fachlich orientiert und teamunterstützend. Er enthält automatische und ma-

nuelle Tests von Anwendungsfunktionen wie beispielsweise Story Tests, Prototypen und Simulationen. Es wird keine Einsicht in den Code benötigt. Anwendungsfunktionen werden dabei auf die jeweiligen Funktionalitäten getestet, die von den Entwicklern geplant wurden. Daher redet man auch von teamunterstützendem Testen.

Quadrant 3 ist fachlich orientiert, aber produkt hinterfragend. In diese Kategorie fallen alle Tests, die nahe am Endnutzer sind. In diesem Quadranten wird unter anderem die Usability der Anwendung manuell getestet. Außerdem enthält dieser Quadrant Akzeptanztests des Produkts für den Anwender.

Quadrant 4 ist technisch orientiert und produkt hinterfragend. Hier werden Eigenschaften des Programms wie das Verhalten unter Last, Performance und Security getestet. Für diese Tests werden oft Spezialisten eingesetzt [1].

Exploratives Testen

Beim agilen Testen fehlt eine umfassende Testbasis, wie sie normalerweise bei spezifikationsorientierten Projekten verwendet wird, um Testfälle zu finden. Unter einer Testbasis versteht man alle Dokumente, aus denen Anforderungen an das Testobjekt hervorgehen. Daher werden oft explorative Tests verwendet, um Testfälle auszumachen [1]. Zunächst werden dabei nur abstrakte Testziele definiert, wie „Bringe das Programm zum Absturz“. Mit diesem Ziel im Hinterkopf erkundet der Tester das Produkt selbständig. Anhand der Erfahrungen, die während des Testens gemacht werden, werden inkrementell neue Testfälle erstellt. Exploratives Testen ist damit dynamisch angelegt und setzt bewusst auf die Eigenverantwortung und das Wissen des Testers.

Fazit

In der agilen Entwicklung wird viel früher mit dem automatisierten Testen begonnen. Dadurch erhalten die Entwickler schnell Feedback zu den implementierten Funktionen. Im Gegensatz zu spezifikationsorientierten Projekten ist das gesamte Team für das Testen verantwortlich.

- [1] M. Baumgartner, et al. (2013). Agile Testing: Der agile Weg zur Qualität, Carl Hanser Verlag, München
 [2] ISTQB (2011). Foundation Level Syllabus (Zitat vom 18.11.2016) <http://www.istqb.org/downloads/send/2-foundation-level-documents/3-foundation-level-syllabus-2011.html>
 [3] S. Geisen, B. Güldali (2012). Agiles Testen in Scrum - Testtypen und Abläufe (Zitat vom 21.10.2016) http://www.sigs-datacom.de/uploads/tx_mwjournal/pdf/geisen_guedali_OS_Agility_2012_8e65.pdf
 [4] L. Crispin, J. Gregory (2008). Agile Testing: A Practical Guide for Testers and Agile Teams, Addison-Wesley

Bildquellen:

- Abbildung 1: Eigener Entwurf nach [4]

Entwicklung einer Architektur für eine Definition von Bedienoberflächen für Touchdisplay und Implementierung einer Anwendung

Tobias Kiechle*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Als Innovationsführer sieht Kärcher sich in der Pflicht neue Standards in Technologie zu setzen, welche sich optimal am Kunden orientieren [1]. Durch diese Pflicht werden die Produkte zunehmend digitalisiert. Hierbei wird auf übersichtliche Bedienung und benutzerfreundliche Handhabung viel Wert gelegt. Gleichzeitig soll durch eine saubere Architektur und Modularität der Anwendungen die Wiederverwendbarkeit und Wartung der grafischen Benutzeroberfläche (Abk. GUI) garantiert werden.

Das Ziel dieser Arbeit ist mit modularer Softwareentwicklung der einzelnen Komponenten eine solche Architektur zu entwickeln. Hierfür wurde der aktuelle Stand der Architekturmuster für Oberflächen gesichtet und kam zum Entschluss, dass das Model-View-Controller-Muster sehr passend ist. Dieses wird im folgenden Abschnitt näher erläutert.

Das Muster Model-View-Controller bildet die Grundlage der Architektur und wird mit dem Verhaltensmuster Zustand und dem Strukturmuster Kompositum kombiniert [2]. Die Wiederverwendbarkeit und Anpassungsfähigkeit des Model-View-Controller-Musters ist für die angestrebte Anwendungsarchitektur ideal, denn die verschiedenen Aufgaben einer interaktiven Anwendung werden modular auf die drei Komponenten aufgeteilt. Hierdurch können später die einzelnen Komponenten neuen Anforderungen angepasst werden, ohne dabei die anderen Komponenten Veränderungen unterziehen zu müssen. Die Aufgaben der Komponenten in einem Model-View-Controller-Muster sind wie folgt:

- **Das Model** kapselt die Datenkommunikation der Anwendung mit dem System. Hierdurch ist es der GUI möglich systemunabhängig zu arbeiten.
- **Die View** enthält die Darstellung für den Anwender. Diese verändert sich durch Signale des Controllers.
- **Der Controller** verarbeitet die Eingaben des Anwenders, interpretiert diese und gibt sie gegebenenfalls weiter.

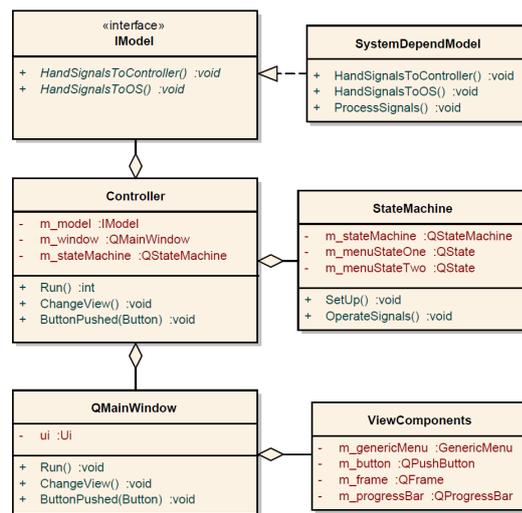


Abbildung 1: Model-View-Controller-Klassendiagramm Beispiel

Aufgezeigt in Abbildung 1 ist ein modifiziertes Model-View-Controller-Architekturmuster, in welchem der Controller die Hauptkommunikationsrolle übernimmt. Um die unterschiedliche Interpretation der Eingabe zu gewährleisten, beinhaltet dieser einen Zustandsautomaten (englisch: state machine). Dieser regelt die unterschiedlichen Behandlungen der Signale, je nach aktivem Zustand.

Abbildung 2 zeigt dies anhand von drei Beispielzuständen.

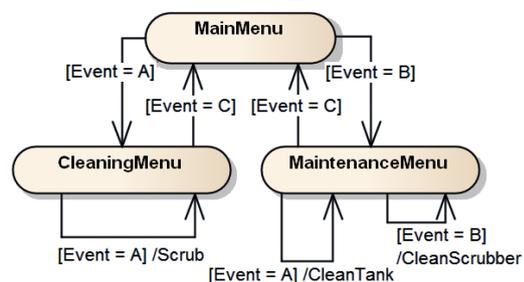


Abbildung 2: Zustandsdiagramm mit 3 Zuständen

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

Die Modularität des Programms gegenüber dem Gerätebetriebssystem wird durch die IModel-Interfaceklasse garantiert. Die View wird mit einem QMainWindow implementiert. Das QMainWindow ist eine Klasse aus dem Qt-Framework. Qt ist ein auf C++ basierendes GUI-Anwendungsframework. Außer einer Vielzahl an GUI-spezifischen Klassen bietet es einen intuitiven SIGNAL- /SLOT-Mechanismus für einfache Kommunikation zwischen Objekten. Zusätzlich ist es einfach auf viele Betriebssysteme portierbar [3]. Für die Arbeit mit Qt bietet sich die integrierte Entwicklungsumgebung (Abk. IDE) Qt Creator an. Diese IDE beinhaltet neben einem Quellcode-Editor und einer Projektverwaltung einen GUI-Designer.

Ausblick der Thesis:

Die Architektur und der damit erstellte „Shared Code“ im Sinne der Wiederverwendbarkeit bieten einen Baukasten für die GUI-Entwicklung an. Somit kann mithilfe generischer Menüklassen schnellstmöglich eine GUI erstellt und individualisiert werden. Diese generischen Klassen folgen den produktübergreifenden Corporate-Designanforderungen, wodurch das Baukastenkonzept ebenso produktübergreifend zum Einsatz kommen kann. Die Kapselung der einzelnen Bauteile unterstützt dies.

Um die Generierung einer GUI zu vereinfachen, wird bereits über Codegenerierung nachgedacht um die Programmierer bei der Erstellung eigener Oberflächen zu unterstützen und Programmierfehler zu vermeiden.



Abbildung 3: Beispielübersicht eines GUI-Menüs

[1] <https://www.kaercher.com/de/inside-kaercher/unternehmen/ueber-kaercher.html> ; Stand 24.11.2016

[2] Architektur- und Entwurfsmuster der Softwaretechnik, Joachim Goll, Springer Vieweg, 2. Auflage

[3] C++ GUI - Programmierung mit Qt4, Jasmin Blanchette & Mark Summerfield, Addison-Wesley, 2. Auflage

Bildquellen:

- Abbildung 1-2: Erstellt von Tobias Kiechle
- Abbildung 3: Screenshot von Teil der GUI, Stand 27.11.2016

Optimierung von Datenbeladungsstrecken und Datenmodellen durch Nutzung von In-Memory Technologien

Christoph Kreibich*, Thomas Rodach, Dirk Hesse

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

In Unternehmen werden verschiedene Informationssysteme eingesetzt, die Daten aus unterschiedlichen Quellen sammeln. Es werden sowohl strukturierte als auch unstrukturierte Informationen in ein System aufgenommen. Strukturierte Daten entstehen innerhalb des Unternehmens, meistens durch ein Enterprise Resource Planning System. Dagegen entstehen unstrukturierte Daten meist außerhalb des Unternehmens, zum Beispiel in sozialen Netzwerken. Damit die gesammelten Informationen sinnvoll eingesetzt werden können, sollten diese in Echtzeit verarbeitet werden können. Dabei stoßen heutige Informationssysteme an ihre Belastungsgrenze. Aus diesem Grund werden die Daten zum Großteil noch nicht in Echtzeit erfasst.

Des Weiteren sind die Datenbanksysteme auf bestimmte Anwendungsfälle zugeschnitten. Es werden unterschiedliche Systeme für das Online Transactional Processing (OLTP) und das Online Analytical Processing (OLAP) eingesetzt. Für OLTP Anwendung ist charakteristisch, dass immer komplette Zeilen aus einer Tabelle gelesen oder in eine Tabelle geschrieben werden. Die Daten können daher sequentiell abgespeichert werden und ein Zugriff auf diese ist relativ einfach zu realisieren. Im Gegensatz dazu nutzen OLAP Anwendungen nicht die Daten einer kompletten Zeile, sondern greifen auf eine begrenzte Anzahl an Spalten zu. Dadurch werden zum Beispiel Summen berechnet. Ein Problem dieser getrennten Datenbanksysteme ist, dass alle Datensätze redundant in beiden Systemen gespeichert werden müssen. Es entsteht ein erheblicher Aufwand bei der Synchronisation der Daten über beide Systeme hinweg. Meistens werden die Daten im OLTP System gesammelt und in Batches in das OLAP System übertragen. OLAP Anwendungen können dadurch nicht immer die aktuellsten Informationen für Analysen nutzen. [1]

Die eingesetzte Hardware hat einen großen Einfluss auf die Geschwindigkeit des gesamten Informationssystems. In den meisten Fällen werden die Daten auf einem Sekundärspeicher,

wie zum Beispiel Festplatten oder Flashspeicher, gespeichert. Die Zugriffsgeschwindigkeit auf den Sekundärspeicher ist ein Flaschenhals, der die Datenverarbeitung in Echtzeit limitiert. Die Performance von Datenbanksystemen, die Festplatten einsetzen, kann verbessert werden. Bei der Parallelisierung werden mehrere Prozessoren eingesetzt, welche die Datenverarbeitung vornehmen. Die anderen Hardwarekomponenten, wie zum Beispiel der Hauptspeicher, können entweder von allen Prozessoren geteilt werden oder sind nur für einen Prozessor zugänglich. Zusätzlich können die Daten mittels Redundant Array of Inexpensive Disks (RAID) über mehrere Festplatten verteilt werden. Je nach RAID Verfahren können Daten parallel abgerufen werden, was den Engpass verringert. Um den Datenzugriff noch effizienter zu gestalten werden Indexstrukturen, wie zum Beispiel B-Bäume eingesetzt. Diese Strukturen sind speziell für den sequentiellen Zugriff auf den Sekundärspeicher entwickelt worden.

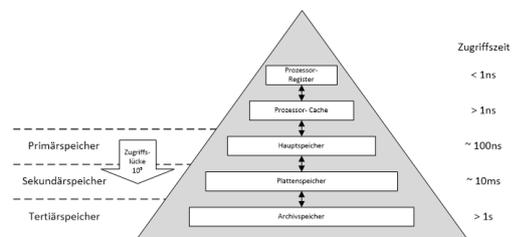


Abbildung 1: Speicherhierarchie

In der Abbildung ist deutlich zu erkennen, dass der Geschwindigkeitsunterschied zwischen dem Sekundärspeicher und dem Hauptspeicher immens ist. Der Grund dafür ist, dass der Prozessor viel schneller auf den Hauptspeicher zugreifen kann als auf eine Festplatte. Eine Verlagerung der Datenhaltung vom Sekundärspeicher in den Hauptspeicher bringt einen nicht unerheblichen Geschwindigkeitsvorteil mit sich. Neben der geringeren Zugriffszeit hat der Hauptspeicher auch noch den Vorteil, dass ein zufälliger Speicherzugriff genau so lange dauert, wie ein sequentieller. Bei Festplatten ist der Unterschied deutlich größer, weswegen

*Diese Arbeit wurde durchgeführt bei der Firma Dymatrix Consulting Group GmbH, Stuttgart

bei Festplatten ein sequentieller Zugriff bevorzugt werden sollte.

Zusätzlich zu der schnelleren Hardware kann das Datenlayout angepasst werden. In einer relationalen Datenbank wird meistens ein zeilenorientiertes Layout eingesetzt. Dabei werden die einzelnen Zeilen in benachbarten Speicherblöcken abgelegt. Das Lesen oder Schreiben einer Zeile ist daher sequentiell möglich. Sollen nur bestimmte Spalten abgerufen werden, wie es bei OLAP Anwendungen häufig der Fall ist, so werden meistens auch überflüssigerweise Daten der Relation abgerufen, die nicht zur Berechnung benötigt werden. Bei einem spaltenorientierten Layout werden die Zeilen zerlegt und die einzelnen Spalten werden in benachbarten Speicherzellen abgelegt. Ein spaltenorientiertes Layout wurde bisher selten eingesetzt, da das Zusammensetzen einer Relation im Sekundär-speicher zu langwierig ist. Bei In-Memory Datenbanken ist ein spaltenorientiertes Layout deutlich besser geeignet, da so OLAP Abfragen sehr schnell beantwortet werden können. Die geringe Zugriffszeit bei einem zufälligen Speicherzugriff erlaubt es die zerteilten Relationen schnell zusammenzusetzen. Ebenso sind Kompressionsverfahren anwendbar, welche die Geschwindigkeit steigern und den Speicherbedarf verringern. [2]

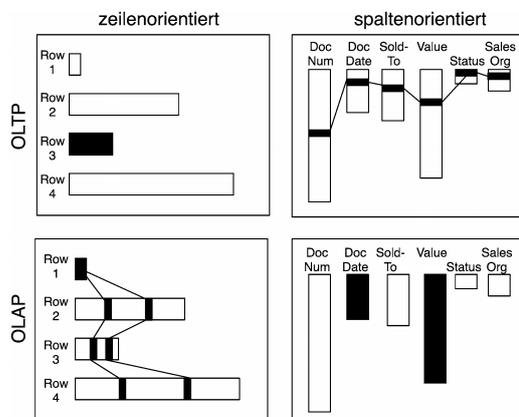


Abbildung 2: Datenzugriff bei einer Zeilen- und Spaltenorientierung

Im Rahmen der Arbeit werden drei Szenarien von Datenbeladungsstrecken getestet. Die Tests werden sowohl mit einer nicht In-Memory und einer In-Memory Datenbank durchgeführt. Der erste Testfall umfasst das Beladen eines Data-Marts mit Daten aus verschiedenen Entitäten. Im zweiten Testfall werden die Kundendaten auf Duplikate überprüft. Hierzu wird ein Dublettenabgleich mit physischen Algorithmen durchgeführt. Falls doppelte Werte erkannt werden, werden diese zu einer Kopfdublette zusammengeführt. Der dritte Testfall umfasst das Berechnen verschiedener Key Performance Indicators (KPI) für das analytische Customer Relationship Management (CRM). Für die Durchführung der Testfälle wird ein Azure Server genutzt, auf dem Microsoft SQL Server 2016 und Data Tools für Visual Studio installiert sind. Microsoft SQL Server 2016 unterstützt unter anderem In-Memory OLTP Tabellen und Columnstores. Die Testfälle sind sowohl im OLTP als auch im OLAP Anwendungsgebiet angesiedelt. Ebenso ist die Mischung aus In-Memory OLTP und Columnstore ein hybrides Datenlayout, welches beide Anwendungsbereiche gut unterstützt. Mit den Testfällen soll überprüft werden, ob eine Performancesteigerung durch den Einsatz von einer In-Memory Datenbank möglich ist. Weitere Optimierungsmöglichkeiten, die mit der Nutzung von In-Memory Datenbanken möglich sind, sollen ebenfalls evaluiert werden. Ein kurzer Test vorab mittels den Microsoft RML Utilities hat ergeben, dass durch den Einsatz von In-Memory Tabellen die Abfragedauer um zwei Drittel reduziert werden konnte. Bei dem Test wurde eine Stored Procedure von 100 Threads jeweils 1000-mal aufgerufen. Wird eine Tabelle verwendet, die auf der Festplatte gespeichert wird, so dauert die Ausführung rund 1 Minute. Bei einer In-Memory Tabelle konnte die Abfrage in ungefähr 20 Sekunden ausgeführt werden. Das Einsparpotential kann je nach Anwendungsfall variieren.

[1] Plattner, H. (2012): In-memory data management. Technology and applications. Second edition. Heidelberg: Springer. ISBN 978-3-642-29574-4

[2] Krüger, J., Grund, M., Tinnefeld, C., Eckart, B., Zeier, A. und Plattner, H. (2010): Hauptspeicherdatenbanken für Unternehmensanwendungen. In: Datenbank-Spektrum, 10(3), S. 143-158. Springer Link. ISSN 1618-2162. Verfügbar unter DOI: 10.1007/s13222-010-0031-5

Bildquellen:

- Abbildung 1: Eigene Darstellung
- Abbildung 2: [2]

Prototypische Umsetzung einer Augmented Reality Applikation zur Markierung von Objekten

Richard Krieger*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Im Jahre 1968 entwickelte Ivan Sutherland das erste Gerät mit welchem es möglich war, durch Überlagerung realer Objekte, einfache Grafiken in Echtzeit auf einem Display anzuzeigen [1]. Da zu dieser Zeit die Entwicklung von Computern gerade in vollem Gange war, setzte sich aufgrund der Tatsache, dass die benötigten Geräte groß, kostspielig und in ihrer Rechenleistung sehr beschränkt waren, diese Technik jedoch zunächst nicht durch. Durch den technischen Fortschritt begünstigt stiegen die Möglichkeiten zur Implementierung komplexer Algorithmen zur Umsetzung von Augmented Reality Anwendungen stark an, sodass um die Jahrtausendwende herum bereits eine Vielzahl solcher Anwendungen existierte. Zeitgleich zu dieser Entwicklung stellte Ronald Azuma 1997 in seiner Arbeit „A survey on Augmented Reality“ [2] eine umfassende Definition des Begriffs „Augmented Reality“ auf, die bis heute Verwendung findet.

Problemstellung

Mit ihrer Software für mobile Endgeräte, wie Smartphones oder Tablet-PCs, bietet die TeamViewer GmbH die Möglichkeit, diese von einem Desktop PC aus zu warten oder fernzusteuern. So kann beispielsweise der Support eines Smartphone Herstellers seinen Kunden, bei der Einrichtung ihrer Geräte, helfen oder, bei aufkommenden Fragen zur Bedienung, Support leisten, indem er sich auf diese über einen Remote-Control (Fernsteuerung) verbindet. Probleme, die nicht direkt etwas mit dem Endgerät zu tun haben, auf welchem TeamViewer installiert ist, können auf diese Weise jedoch nicht behoben werden, da die supportleistende Seite lediglich Zugriff auf das Endgerät besitzt. Um solche Fälle abzudecken und somit den Support in beliebigen Situationen zu gewährleisten, wurden im Rahmen dieser Bachelorarbeit Methoden zur Realisierung einer Augmented Reality Applikation untersucht und basierend auf den Ergebnissen dieser Untersuchung zwei Prototypen für das Android Betriebssystem entwickelt, welche das Markieren von Objekten innerhalb eines Ka-

merabildes ermöglichen, um so dem Anwender Arbeitsschritte zur Lösung seines Problems in Echtzeit anzuzeigen. Hierbei sollten, durch Verwendung von Trackingmethoden, markierte Bereiche über mehrere Bilder hinweg verfolgt werden können, sodass gesetzte Markierungen, selbst bei Bewegung des Anwenders, an der richtigen Position angezeigt werden.

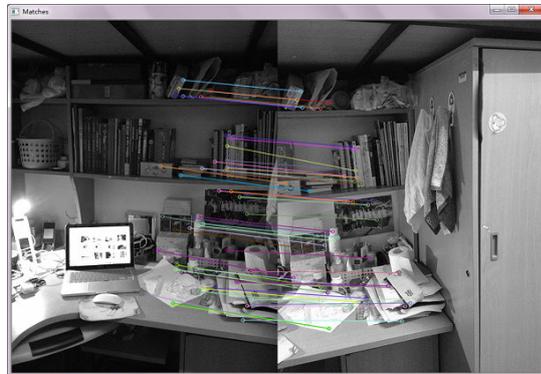


Abbildung 1: Beispiel für den Vergleich von Keypoints

Umsetzung

Zur Umsetzung eines solchen Trackingsystems werden die Kamerabilder, mithilfe verschiedener Algorithmen, auf markante Punkte, auch „Features“ oder „Keypoints“ genannt, untersucht. Durch das Vergleichen der gefundenen Keypoints zweier Bilder lassen sich, falls Übereinstimmungen gefunden werden, beispielsweise Bewegungen, Rotationen oder Blickwinkeländerungen registrieren. Da jedoch die Position solcher Punkte allein nicht ausreicht, um identische Keypoints beim Vergleichen zu erkennen, müssen diese um Zusatzinformationen erweitert und dadurch eindeutig identifizierbar gemacht werden. Aus diesem Grund wird für jeden entdeckten Keypoint ein sogenannter „Feature Descriptor“ berechnet, welcher diesen um die benötigten Zusatzinformationen, wie beispielsweise die Orientierung oder Qualität des Keypoints, erweitert. Abbildung 1 zeigt einen solchen Vergleichsprozess, bei welchem Keypoints die im linken Bild er-

*Diese Arbeit wurde durchgeführt bei der Firma TeamViewer GmbH, Stuttgart-Vaihingen

kannt und im rechten Bild wiedererkannt wurden, mit einer farbigen Linie verbunden sind.

Keypoints, die häufig wiedererkannt werden, lassen sich somit über mehrere Bilder hinweg verfolgen, was die Grundlage für einen Trackingprozess schafft. Um nun eine Markierung im Raum zu platzieren, deren Position sich ständig aktualisiert, muss diese in Relation zu einer Auswahl an gefundenen Keypoints gesetzt werden. Hierdurch wird erreicht, dass die Position der Markierung, im Falle einer registrierten Bewegung der Keypoints, an diese angepasst und somit aktualisiert wird.

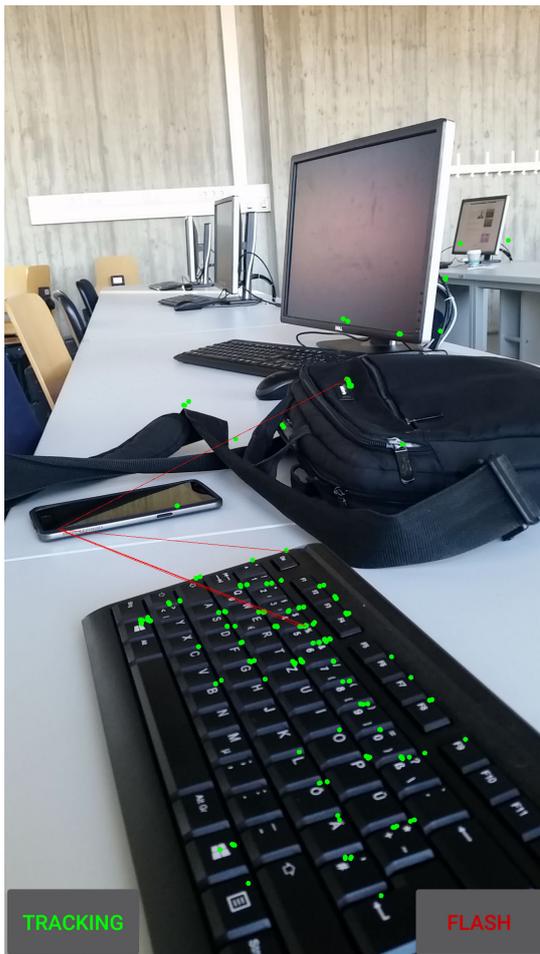


Abbildung 2: Positionierung einer Markierung im ersten Prototyp

Um eine durchgehende Aktualisierung der Markierung zu gewährleisten sollten daher nur Keypoints gewählt werden, deren Wiedererkennungsrates sehr hoch ist. Da bereits eine große Menge an verschiedenen Implementierungen zur Erkennung geeigneter Keypoints existiert, wurde bei der Umsetzung des Trackingsystems auf vorimplementierte Programm Bibliotheken und SDKs zurückgegriffen. Für die Wahl der passenden Implementierung wurden die möglichen Kandidaten, hinsichtlich ihres Funktionsumfangs, miteinander verglichen und bewertet. Anhand dieser Vergleiche wurden zwei Implementierungen

für die Entwicklung der Prototypapplikationen ausgewählt.

Prototyp Version 1 (OpenCV)

Für die Umsetzung des ersten Prototyps wurde die OpenCV Programm Bibliothek verwendet, welche verschiedenste Methoden für die digitale Bildverarbeitung und zur Erkennung und Beschreibung von Keypoints beinhaltet. Mithilfe der Android Camera-API wurde die Gerätekamera angesprochen, um neben dem Bildabgriff auch Zusatzfunktionalitäten, wie beispielsweise das Blitzlicht oder den Autofokus, zu implementieren. Abbildung 2 zeigt die Ausgabe des Prototyps. Neben der Anzeige des Kamerabilds lassen sich deutlich die grünen Keypoints erkennen, die durch die Verwendung von OpenCV berechnet werden. Soll nun eine Markierung gesetzt werden, so kann der Anwender diese durch ein Tippen auf den Bildschirm im Raum platzieren. Die Touchposition wird dann relativ zu einer bestimmten Menge an ausgewählten Keypoints berechnet und abgespeichert. Werden diese Keypoints im nächsten Kamerabild wiedererkannt, so werden rote Linien von diesen zur neu berechneten Position der Markierung gezeichnet (Abbildung 2). Der Schnittpunkt dieser Linien kennzeichnet dann die neue Position der Markierung.

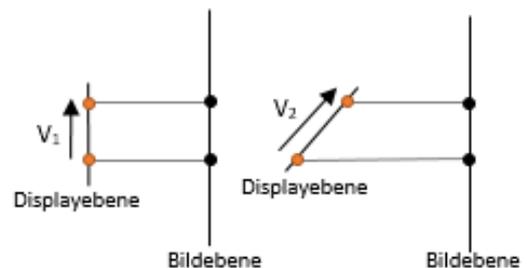


Abbildung 3: Beispiel für das Tiefenwertproblem bei der Verwendung von OpenCV

Da OpenCV jedoch nur zweidimensionale Keypoints aus dem Kamerabild extrahiert, ergeben sich, dadurch, dass sowohl die Keypoints als auch die Markierung ohne Tiefeninformationen betrachtet werden, große Ungenauigkeiten bei der Bestimmung der neuen Markierungsposition, wenn das Gerät gekippt wird. Dieser Zustand soll in Abbildung 3 bildhaft dargestellt werden.

Wie hier deutlich wird, sind die beiden Vektoren v_1 und v_2 der Displayebene nach dem Kippen des Geräts unterschiedlich groß, obwohl die Keypoints sich nicht voneinander wegbewegt haben. Dies hat zur Folge, dass die 2D-Vektoren, die berechnet wurden um die Markierung zu setzen, nach einem Kippen des Gerätes nicht mehr verwendet werden können, da sich die Position der Punkte zueinander auf dem Bildschirm des Gerätes verändert hat. Aufgrund dieser Problematik ist OpenCV, ohne

weitere Berechnungen von dreidimensionalen Informationen zur Umgebung, nur bedingt für die Umsetzung einer solchen Applikation geeignet.

Prototyp Version 2 (Vuforia SDK)

Der zweite Prototyp wurde mit der Vuforia SDK implementiert. Diese bietet, ähnlich wie OpenCV, geeignete Methoden zur Umsetzung eines featurebasierten Trackingprozesses. Der Unterschied hierbei ist jedoch, dass der Such- und Vergleichsprozess sowie das Ansprechen der Gerätekamera bereits vorimplementiert sind, sodass nur gewisse Methoden aufrufen werden müssen, um die gewünschte Funktionalität zu verwenden. Zur Implementierung des Prototyps wurde das „User Defined Targets“ Beispiel der Vuforia Beispielapplikation modifiziert und um benötigte Funktionalitäten erweitert. Hierbei soll angemerkt werden, dass die Entwicklung dieses Prototyps noch nicht abgeschlossen ist, da einige Funktionen, die später näher thematisiert werden, noch entwickelt werden müssen. Die ursprüngliche Funktion des „User Defined Targets“ Beispiels der Vuforia SDK ist es, auf Knopfdruck ein virtuelles Objekt im Raum zu platzieren. Abbildung 4 zeigt, wie eine virtuelle Teekanne mithilfe dieser Beispielapplikation auf einem Tisch platziert wird.

Um dies zu erreichen, wird eine Ebene, die parallel zum Gerät liegt in den Raum ge-

legt, auf welche dann die Teekanne und später auch potentielle Markierungen platziert werden können. Im weiteren Verlauf dieser Bachelorarbeit soll, um die bereits beschriebene Problematik aus Abbildung 3 zu beseitigen, die Transformation zwischen 2D- und 3D-Punkten implementiert werden. So müssen 2D-Punkte, die durch das Tippen auf das Gerätedisplay ausgewählt werden, auf die Ebene, auf welcher ursprünglich die Teekanne gesetzt wurde, übertragen werden. Diese 3D-Punkte liegen dann fix im Raum und spiegeln die Koordinaten der gesetzten Markierung wieder. Durch ständige Projektion dieser Punkte zurück auf die Ebene des Gerätedisplays kann somit, selbst beim Kippen des Geräts, eine korrekte Aktualisierung der Markierungsposition erreicht werden.



Abbildung 4: „User Defined Targets“-Beispiel der Vuforia SDK

-
- [1] Ivan E. Sutherland, A headmounted three dimensional display, <http://www.cise.ufl.edu/research/lok/teaching/ves07/papers/sutherland-headmount.pdf>, 1968, Zugriff am 01.12.2016
 [2] Ronald T. Azuma, A Survey of Augmented Reality, <http://www.cs.unc.edu/~symbol{126}azuma/ARpresence.pdf>, August 1997, Zugriff am 01.12.2016

Bildquellen:

- Abbildung 1: https://academic.levy.at/img/stitch/optimised_keypoint_match.png
- Abbildung 2-4: Eigene Darstellung

Ansätze zur Reduzierung von Installationspaketen auf der mobilen Plattform Android

Sven Lipiz Radalga*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Der Smartphone Markt breitet sich heut zutage auch in Gebieten aus, die nicht gut vernetzt sind oder in denen die Kosten für das Nutzen des Internets sehr teuer sind. Um diese potenziellen Kunden für sich gewinnen kann, muss man ihnen entgegenkommen und seine App anpassen. Damit wird die App für die potenziellen Kunden um einiges attraktiver.

Unattraktiv ist eine App, die zum Beispiel zu lange zum Herunterladen braucht. Bei schlechter Verbindung geht der Download nur stockend voran oder kann unterbrochen werden, wenn die Verbindung zum Internet verloren geht. Ein weiterer Grund, warum eine App unattraktiv für einen Kunden erscheinen kann, ist wenn diese App zu teuer ist. Dies kann auch auf Apps zutreffen, die kostenlos sind. Die Kosten um die es geht, entstehen nicht durch den Kauf einer App, sondern beim Downloaden der App auf das Smartphone. Dies trifft auf den Märkten zu in denen das „pay-by-the-byte“ Prinzip angewendet wird. Die Kunden müssen für jedes Byte, dass sie herunterladen, zahlen und dies kann bei besonders großen Apps sehr teuer werden.

Google hat bereits eine Checkliste erstellt, die die Android Entwicklern helfen soll auf diese besonderen Gegebenheiten aufmerksam zu machen. Die Checkliste nennt sich „Building for Billions“ und bietet einem Möglichkeiten an wie man seine App verbessern kann, damit diese für alle Kunden attraktiv wird. Das Videoportal YouTube entwickelt bereits an einer neuen App, die es ermöglichen soll, YouTube auch offline zu verwenden. YouTube Go sollte vorerst nur in Indien erscheinen sollte, doch der Google-CEO Sundar Pichai hat angekündigt, dass diese auch außerhalb Indiens zu haben sein wird. Die App ermöglicht das Downloaden von YouTube Videos und bietet die Funktion an, diese mit anderen Nutzern der App per Bluetooth zu teilen. Durch dieses Feature ist eine Gute Internetverbindung nicht mehr zwingend notwendig. Videos, die heruntergeladen wurden, können jederzeit ohne Internetverbindung angesehen werden. Das Teilen von Videos ermöglicht es, dass nicht jeder Nutzer

das Video downloaden muss, und kann somit Kosten vermeiden.

In dieser Zusammenfassung wird thematisiert wie das Installationspaket einer Android App aufgebaut ist und welche Möglichkeiten es gibt um die Größe des Installationspakets zu reduzieren. Um zu verstehen, wie man das Installationspaket einer Android App reduzieren kann, ist es wichtig zu wissen wie diese aufgebaut ist.

Das Installationspaket einer Android App ist eine APK Datei. Eine APK Datei ist ein ZIP Archiv, das alle Dateien enthält, die für die App benötigt werden. Die APK Datei ist wie folgt aufgebaut:

- META-INF/: Dieses Verzeichnis enthält, wenn die APP signiert wurde, die Signaturdateien CERT.SF und CERT.RSA und außerdem noch zusätzlich die Manifest Datei MANIFEST.MF
- assets/: Das assers/ Verzeichnis enthält App assets, auf welche die App mithilfe den AssetManager zugreifen kann.
- res/: Dieses Verzeichnis enthält Image Ressourcen und Layouts.
- lib/: In diesem Ordner befinden sich kompilierter nativer Programmcode, der je nach Prozessorarchitektur, in einem eigenen Unterverzeichnis liegt. Beispiele für die Unterverzeichnisse wären: armeabi, x86, mips.
- resources.arsc: Diese Datei enthält XML Ressourcen aus dem res/values Ordner des Projektes. Diese sind meist Strings in allen Sprachen die erstellt wurden und Styles. Des Weiteren enthält es auch Pfade zu Layouts und Imageressourcen.
- classes.dex: Diese Datei enthält Klassen, die in das DEX File Format kompiliert wurde. Unter anderem sind alle Enums in dieser Datei enthalten.
- AndroidManifest.xml: Dieses File enthält eine Reihe von wichtigen Informationen, wie der Name der App und die Versionsnummer. Des Weiteren sind auch alle Be-

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

rechtungen der App eingetragen und welche Features diese App benötigt.

Fast jeder dieser Dateien oder Verzeichnisse kann man bei der Entwicklung beeinflussen. Doch bevor man versucht, alles zu optimieren, ist eine Analyse der App empfehlenswert. Es gibt Tools, die eine APK analysieren kann und ausgibt welche Datei in welchem Verzeichnis

wie viel Speicher verbraucht.

Android Studio bietet seit der Version 2.2 einen APK Analyzer, dem es möglich ist APKs zu öffnen. Dieser APK Analyzer zeigt zu jeder Datei/Verzeichnis die Downloadgröße, die tatsächliche Größe und einen Prozentwert im Verhältnis zur Gesamtgröße an.

File	Raw File Size	Download Size	% of Total Download size
classes.dex	8 MB	3.1 MB	63.7%
res	2 MB	1.2 MB	25.1%
resources.arsc	904.9 KB	219.3 KB	4.3%
classes2.dex	670.2 KB	254 KB	5%
META-INF	186.5 KB	57.3 KB	1.1%
assets	83.3 KB	26.3 KB	0.5%
AndroidManifest.xml	23.7 KB	4.7 KB	0.1%
NOTICE_firebase_jvm	2 KB	1 KB	0%
NOTICE_firebase_android	2 KB	1 KB	0%

Abbildung 1: Android Studio APK Analyzer

Entwickler, die kein Android Studio verwenden, gibt es andere Möglichkeiten eine APK zu analysieren. ClassyShark, eine Java Applikation, ist einer dieser Möglichkeiten.

Die meistens ist es einer oder mehr der folgenden drei Bereiche, die eine App zu groß werden lässt:

1. Ressourcen
2. Quellcode
3. Nativer Code

Ist der Ressourcenbereich zu hoch liegt dies meist an zu vielen oder zu großen Image Ressourcen. Das liegt meistens auch daran, dass

man mehrere Bildschirmen unterstützt wie ldpi, mdpi, hdpi, xhdpi. Eine Folge davon ist, dass man die meisten Image Ressourcen, in den unterstützten Auflösungen, zur App hinzufügt.

Eine Lösung für dieses Problem wäre, dass man nicht alle Bildschirmen unterstützt. Hierbei kann auch eine Analyse helfen was für Android Geräte im Umlauf sind. Zum Beispiel haben, laut folgender Tabelle von der Android Developer Seite, 1,8% der im Umlauf befindlichen Android Geräte eine Bildschirmgröße von ldpi.

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	1.6%						1.6%
Normal		3.1%	0.2%	38.7%	30.4%	15.5%	87.9%
Large	0.2%	3.9%	1.9%	0.4%	0.4%		6.8%
Xlarge		2.8%		0.3%	0.6%		3.7%
Total	1.8%	9.8%	2.1%	39.4%	31.4%	15.5%	

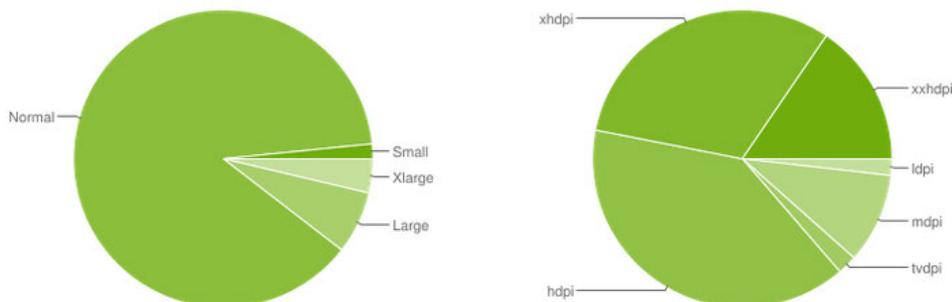


Abbildung 2: in Umlauf befindlich Android Geräten und ihrer Bildschirmgröße und Dichte

Dies könnte ein Grund sein, dass Idpi in seiner APP nicht zu unterstützen und man könnte sich Image Ressourcen sparen.

Eine weitere Methode Ressourcen zu verringern ist es unbenutzte Ressourcen zu entfernen. Dabei handelt es sich nicht nur um Ressourcen, die der Entwickler dem Projekt hinzugefügt hat, sondern auch um Ressourcen, die indirekt in das Projekt gelangen. Dies entsteht zum Beispiel durch das Hinzufügen von Libraries. Die Android Support Library bringt nicht nur Quellcode mit sich, sondern auch String Ressourcen in verschiedenen Sprachen und auch Image Ressourcen. Diese lassen sich ganz leicht wieder entfernen. Gradle kann unbenutzte Ressourcen automatisch entfernen, indem man in die „build.gradle“ Datei „shrink-Resources“ aktiviert.

Eine weitere Methode die Größe einer App zu verringern, ist das Verwenden von APK Splits. APK Splits ist das Erstellen von mehreren kleineren APKs, die jeweils nur einen Teilbereich unterstützen, anstatt einer APK, die alles unterstützt. Es wird von den Android Developer empfohlen nur eine APK zu erstellen, doch im Rahmen von „Building for billions“ wird auch dazu geraten die APK auf zu teilen, um die APK Datei klein zu halten.

Eine APK kann anhand verschiedenen Kriterien aufgeteilt werden. Unter ihnen befindet sich auch die Bildschirmgröße. Es ist möglich, für jede Bildschirmgröße eine eigene APK zu bauen. Diese einzelnen APKs enthalten dann noch die nötigen Ressourcen, die ihrer Bildschirmgröße zugeordnet wurde und zusätzlich die Ressourcen, die keiner Größe zugeordnet wurden. Ein weiteres Kriterium, um eine APK zu teilen, ist die Prozessorarchitektur (ABI). Dabei wird der Native Code reduziert. Das Ergebnis ist mehrere kleine APKs, die jeweils nur eine Architektur in dem „lib/“ Unterverzeichnis aufweisen. Es ist auch möglich, die APK anhand verschiedener dieser Kriterien gleichzeitig auf zu teilen. Als Ergebnis erhalten wir das Kreuzprodukt aller dieser Kriterien.

Im weiteren Verlauf dieser Arbeit werden zusätzliche Methoden, zur Reduzierung des Installationspaketes genannt. Außerdem werden auch Auswirkungen eingegangen, die diese Methoden mit sich bringen. Des Weiteren wird auch auf die mobile Plattform iOS eingegangen und Methoden beschrieben, wie die Installationsgröße einer Apple APP reduziert werden kann.

-
- [1] Android Developers, 2016, „Building for Billions“, [online], <https://developer.android.com/distribute/essentials/quality/billions.html>
 - [2] winfuture, 2016, „YouTube Go: Google start spezielle Offline App („aber“ inside)“, [online], <http://winfuture.de/news,94192.html>
 - [3] Reduce APK Size, 2016, „Understanding the APK Structure“, [online], <https://developer.android.com/topic/performance/reduce-apk-size.html#apk-structure>
 - [4] Configure APK Splits, 2016, „Configure Your Build for APK Splits“, [online], <https://developer.android.com/studio/build/configure-apk-splits.html#configure-split>

Bildquellen:

- Abbildung 1: <https://developer.android.com/studio/build/apk-analyzer.html>
- Abbildung 2: <https://developer.android.com/about/dashboards/index.html#Screens>

Prototypischer Entwurf und Implementierung eines Vehicle-to-X Protokollstacks gemäß dem Standard WAVE einschließlich einer Beispielapplikation unter Verwendung des Nachrichtensets nach SAE J2735 in einer AUTOSAR-Umgebung

Sebastian Lobsinger*, Reinhard Keller, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Bereits seit vielen Jahren sind in Software- und Hardwarekomponenten Möglichkeiten für die Kommunikation innerhalb von Fahrzeugen vorhanden. Die dabei ausgetauschten Informationen ermöglichen eine Vielzahl an neuen Funktionalitäten sowie eine Verbesserung von bestehenden Funktionen des Gesamtsystems. Bislang ist jedoch eine Kommunikation zwischen mehreren Fahrzeugen nicht vorgeesehen. Vehicle-to-X (V2X) ist eine Technologie, welche diese Einschränkung aufhebt und eine Fahrzeug-zu-Fahrzeug-Kommunikation ermöglicht. Neben dieser wird auch eine Kommunikation zwischen Fahrzeugen und Verkehrsinfrastrukturelementen ermöglicht. Als Verkehrsinfrastrukturelemente werden dabei beispielsweise Ampelanlagen und Verkehrszeichen angesehen, die mit V2X Technologie ausgestattet ebenfalls an der Kommunikation teilnehmen können. Der Vorteil von V2X liegt im Informationsgewinn, welcher durch den Informationsaustausch zwischen den Kommunikationsteilnehmern entsteht. Somit steht den Fahrzeugen ein erweitertes Abbild der Umgebung zur Verfügung, welches eine bessere Einschätzung der aktuellen Situation erlaubt.



Abbildung 1: V2X Anwendungsfall Baustellenwarnung

Die Ziele liegen sowohl in der Erhöhung der Sicherheit und der Effizienz des Verkehrssystems als auch in der Steigerung des Fahrerkomforts. So kann beispielsweise das Baustellenschild einer Wanderbaustelle eine Warnung versenden. Der Empfänger weiß damit, dass mit einer Baustelle zu rechnen ist, selbst wenn

diese (z.B. aufgrund der Straßenführung) noch nicht sichtbar ist. Abbildung 1 stellt eine solche Baustellenwarnung über V2X grafisch dar. Ein weiterer Anwendungsfall ist die Übertragung von Informationen zu den Signalphasen einer Ampelanlage. Hierbei wird von der Ampelanlage die verbleibende Zeit bis zum Signalwechsel versandt. Der Nachrichtempfänger kann somit seine Fahrweise der aktuellen Situation entsprechend anpassen.

V2X-Technologien werden durch mehrere Standards festgelegt. Diese unterscheiden sich je nach Entwicklungs- und Einsatzgebiet. Europa, Japan und die USA haben jeweils eigene Spezifikationen erstellt, welche zu großen Teilen nicht untereinander kompatibel sind. Deswegen muss zwischen den einzelnen Standards unterschieden werden. Im Rahmen der vorliegenden Bachelorarbeit wurde ein V2X-Stack nach dem amerikanischen Standard, welcher auch unter dem Namen *Wireless Access in Vehicular Environments* (WAVE) bekannt ist, für die Integration in eine AUTOSAR-Umgebung entworfen und prototypisch implementiert. Der verwendete Protokollstack ist in Abbildung 2 dargestellt und wird im Folgenden beschrieben.

Der WAVE Standard beschreibt eine drahtlose Kommunikation auf Basis des Standards IEEE 802.11, welcher die bekannte WLAN-Technologie spezifiziert. Dieser wird für den Einsatzzweck im Automobilbereich angepasst und erlaubt eine Kommunikation zwischen Fahrzeugen, welche sich mit Geschwindigkeiten von bis zu 250 km/h bewegen. Außerdem werden mehrere Kanäle definiert, wobei ein Kanal für die dedizierte Nutzung für sicherheitskritische Informationen verwendet wird. Eine Erweiterung der MAC-Schicht (WMAC) ermöglicht die Zuweisung eines Kanals samt Priorisierung für jedes einzelne Paket.

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

Auf dem *Logical Link Control Level* (LLC) werden Nachrichten des WAVE-Standards, sogenannte *WAVE Short Messages* (WSM), anhand des *Ethernet-Headers* erkannt und an das darüber liegende *WAVE Short Message Protocol* (WSMP) weitergeleitet. Das WSMP, welches vom *Institute of Electrical and Electronics Engineers* (IEEE) spezifiziert wird, beschreibt das Netzwerk- und das Transportprotokoll des WAVE-Stacks. Übertragungsparameter wie die Sendeleistung und der verwendete Kanal können dabei auf Paketbasis konfiguriert werden [1].

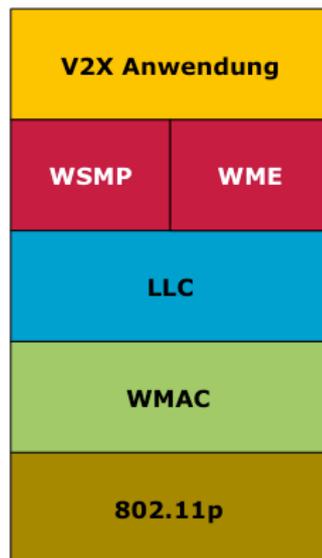


Abbildung 2: Protokollstack nach WAVE Standard

Die Adressierung erfolgt über die MAC-Adresse, wobei sicherheitsrelevante Informationen via Broadcasts versendet werden, so dass alle Teilnehmer in Empfangsreichweite die Informationen erhalten. Auf Transportebene werden die Nachrichten mittels eines Identifiers verarbeitet. Dieser Identifier hängt von dem Verwendungszweck der Nachricht ab und ermöglicht eine Weiterleitung an eine bestimmte Applikation. Neben dem WSMP wird noch ein Managementmodul (WME) definiert, welches die Verwaltungsfunktionalitäten des WSMP realisiert.

Oberhalb des WSMP sind die Applikationen angesiedelt, welche die eigentliche Anwendungslogik implementieren. Die Kommunikation erfolgt dabei über ein einheitliches Nachrichtenset, welches von der *Society of Automotive Engineers* (SAE) spezifiziert wird. Der Vorteil dieses standardisierten Nachrichtensets ist die problemlose Kommunikation zwischen Fahrzeugen unterschiedlicher Hersteller [2]. Für eine sicherheitskritische Kommunikation wird die *Basic Safety Message* (BSM) verwendet. Diese Nachricht wird periodisch versandt und enthält unter anderem Informationen zur Position und zur Fahrtrichtung des Fahrzeugs. Informationen über aktuell aufgetretene sicherheitskritische Ereignisse können ebenso versendet werden.

Ziel der Bachelorarbeit war der Entwurf eines V2X-Protokollstacs nach dem WAVE-Standard für den Einsatz innerhalb der AUTOSAR-Architektur. Dabei war sowohl die Umsetzung der Anforderungen des WAVE-Standards, als auch die Verwendung der von AUTOSAR definierten Konzepte entscheidend. AUTOSAR fordert beispielsweise, dass alle Speicherpuffer statisch allokiert werden. Basierend auf dem Entwurf erfolgte anschließend eine prototypische Implementierung des Stacks. Die Implementierung erfolgte in C für ein Universalsteuergerät auf PowerPC-Basis. Zusätzlich wurde eine Applikation implementiert, welche einen Anwendungsfall des V2X-Systems darstellt. Der dargestellte Anwendungsfall ist das sogenannte *Emergency Electronic Brake Light* (EEBL). Dabei wird bei starken Bremsvorgängen, wie beispielsweise einer Vollbremsung, eine BSM versandt, in welcher der Bremsvorgang gekennzeichnet ist. Auf Empfangsseite wird die BSM ausgewertet und bei Relevanz der Fahrer informiert. Die physikalische Anbindung an die V2X-Kommunikation erfolgte über ein Hardware-Interface, welches die notwendige Funktechnik enthielt und über Ethernet angesteuert wurde. Eine Testumgebung bestehend aus einem weiteren Interface und CANoe als Analyse- und Testwerkzeug erlaubte eine anschließende Validation des Systems.

[1] IEEE 1609.0: Guide for Wireless Access in Vehicular Environments (WAVE) Architecture, 2013

[2] SAE International: Dedicated Short Range Communications (DSRC) Message Set Dictionary, 2016

Bildquellen:

- Abbildung 1: Vector Informatik GmbH
- Abbildung 2: Eigene Darstellung

Programmierung eines Rapid-Prototyping Universalboards und einer GUI für die Inbetriebnahme und den Betrieb unterschiedlicher DCDC-Wandler Topologien

Kay Müller*, Werner Zimmermann, Karlheinz Höfer

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Als Zulieferer wird es immer wichtiger, dem Kunden in Projekten erste Ergebnisse zu liefern. Vor allem in Gebieten der Elektromobilität, welchen in Anbetracht des Umstiegs auf alternative Energien immer mehr Relevanz zukommt, muss man sich mit ersten Prototypen gegen die Konkurrenz behaupten. Da bei Simulationen meist nicht alle Randbedingungen erfasst werden können, benötigt es für schnelle Lösungen eine sichere und flexible Möglichkeit, die Hardware in Betrieb zu nehmen. Der Mikrocontroller der Bachelorarbeit soll dies für verschiedene DCDC-Topologien gewährleisten. Der Aufbau des Projekts sieht wie folgt aus.

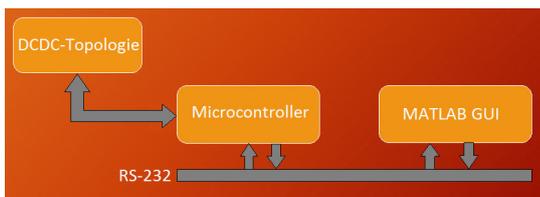


Abbildung 1: Aufbau des Projekts

Der Mikrocontroller wird von einer grafischen Benutzeroberfläche angesteuert. Die Kommunikation zwischen der GUI und dem Controller ist durch eine serielle Schnittstelle realisiert. Nach vollständiger Konfiguration durch den Benutzer kann dieser die Pulsweitenmodulatoren des Controllers aktivieren, um die MOSFETs auf der DCDC-Topologie anzusteuern. Funktionen, wie das Wechseln einer Topologie oder das Ändern von Grenzwerten, werden deaktiviert, um einen sicheren und stabilen Betrieb zu ermöglichen. Wird ein Grenzwert überschritten oder bricht die Verbindung zur GUI ab, geht der Controller in den sicheren Zustand und ein Fehlerbericht wird für den Benutzer bereitgestellt. Die Programmierung des Controllers ist in MATLAB Simulink durch Codegenerierung vorgenommen. Dafür benötigt man entsprechend das vorgesehene Hardwarepaket, worin beispielsweise Register, Speicher und Takte deklariert sind. Die einzelnen Module werden dann mit Simulink-Blöcken dargestellt. Reichen diese nicht aus

oder muss zeitkritisch auf Events reagiert werden, kann C-Code durch Simulink Coder Blöcke eingebettet werden. Aufgrund der Codegenerierung wird ein Struct mit allen Konstanten des Simulink-Modells erstellt, welches zur Laufzeit beschrieben und gelesen werden kann. So wird beispielsweise auf eingehende serielle Nachrichten reagiert und folglich werden die Werte des Modells angepasst. Die Erstellung der Benutzeroberfläche ist mithilfe von MATLAB GUIDE realisiert.

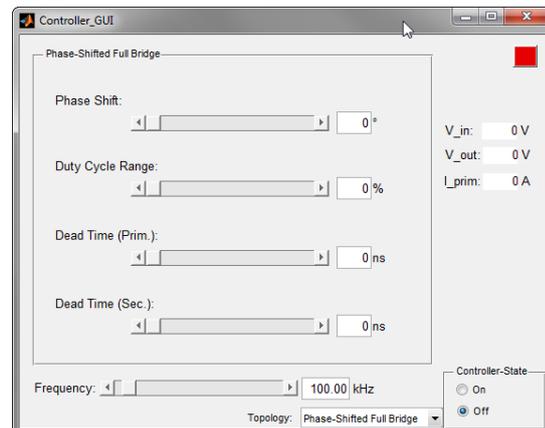


Abbildung 2: Grafische Benutzeroberfläche

Man kann Knöpfe, Textfelder und Schieberegler platzieren, ihnen Eventhandler zuteilen, sie ausblenden und einblenden. Erfolgt durch den Nutzer eine Änderung der Werte, so wird diese in einem Eventhandler auf ihre Sinnmäßigkeit überprüft, bearbeitet und anschließend an den Controller gesendet. Erfolgt nach einer bestimmten Zeit keine Antwort des Controllers oder ist die Antwort korrupt, so wird die Kommunikation abgebrochen. Da bei aktiver Verbindung die GUI in regelmäßigen Zeitschritten ein Lebenssignal an den Controller schickt, erkennt auch dieser den Verbindungsabbruch.

Um die Sicherheit zu gewähren, dass auch bei Hardwarefehlern nichts zu Schaden kommt, wurde eine Abschaltung für Kurzschlüsse und für Überschreitung der Grenzspannung eingerichtet. Für Kurzschlüsse werden Komparatoren verwendet, um per

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

Hardware-Interrupt die PWMs auszuschalten und somit die Verbindungen zu unterbrechen. Bei Überschreitung der Grenzspannung wird erst beim nächsten Lesen der ADCs per Software reagiert, daher ist diese Abschaltung verglichen mit der auf Hardware relativ langsam. Da die Spannungsänderung aber weniger zeitkritisch ist, reicht diese Abschaltung aus. Für einen kleinen Einblick, wie die PWM-Signale aussehen können und was insbesondere beachtet werden muss, wird hier beispielsweise auf die Ansteuerung einer Phase-Shifted Full Bridge eingegangen. Es wird ein Transformator verwendet, um eine galvanische Trennung zwischen Hoch- und Niedrigvoltseite zu erreichen. Diese ist gewünscht, um den DC-Anteil der Spannung zu isolieren [1].

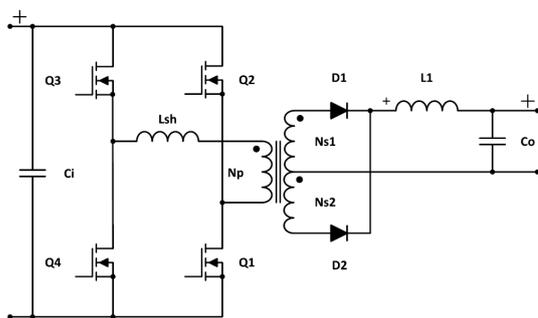


Abbildung 3: Phase-Shifted Full Bridge

Da eine Wechselspannung für die Energieübertragung nötig ist, werden abwechselnd Q1 mit Q3 und Q2 mit Q4 geschaltet. Durch eine Phasenverschiebung kann man nun die Länge des effektiven Pulses verringern. In der Abbildung ist dieser effektive Puls durch den schraffierten Bereich gekennzeichnet. Man muss allerdings darauf achten, keinen Kurzschluss zu generieren, daher wird eine kleine Totzeit zwischen

den Pulsen gehalten, um einen heißen Pfad zu verhindern.

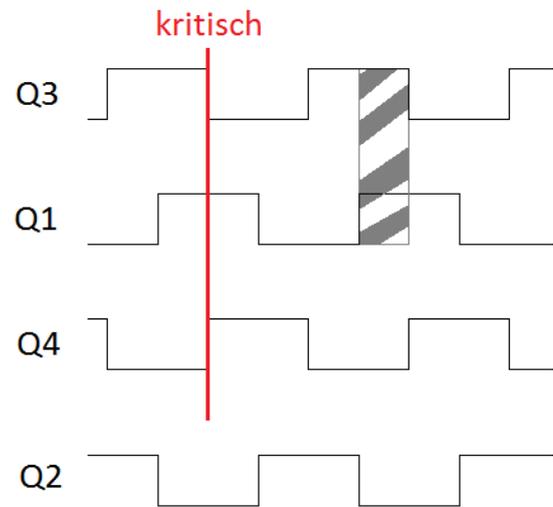


Abbildung 4: PWM Signale der Brücke

Abschließend sind noch Erweiterungsmöglichkeiten des Projekts zu erwähnen. Der verwendete Mikrocontroller enthält eine parallele Prozessoreinheit, die unabhängig vom Hauptprozessor agieren kann. Dadurch kann man extrem zeitkritische Ausführungen wie Spitzenstrommessungen oder Regelschleifen für diese realisieren. Um das Projekt unabhängig von MATLAB zu machen, müsste eine EEPROM Emulation auf dem Flash umgesetzt und die GUI in einer unabhängigen Sprache implementiert werden, damit Einstellungen über die GUI auf dem Flash abgespeichert werden könnten. So würde ein Standalone-Betrieb ohne Simulink möglich sein.

[1] Robert W. Erickson – DC-DC Power Converters

Bildquellen:

- Abbildung 1,2,4: Eigene Abbildung
- Abbildung 3: Ti – Power Topologies Poster (Rev. F)

Prozessuale Einführung von Kanban (IT) in einem nach Scrum agierenden Team in Hinsicht auf Continuous Delivery

Benjamin Munzinger*, Kai Warendorf, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Durch die sich ständig ändernden Anforderungen der Märkte an die gelieferte Software, gerade im Web-Development, wird in der Entwicklung dieser ein äußerst agiles Vorgehen immer notwendiger. Daher werden hier sehr reaktionsfreudige Prozesse, welche kleine Änderungen in kürzester Zeit dem Kunden verfügbar machen können, immer weiter gefördert. Inzwischen hat sich Scrum, als solch ein reaktionsfreudiges System, in den meisten Teams im Web-Development etabliert. So auch in der Entwicklung der smart.com der Daimler AG. Jedoch können auch diese nach Scrum agierenden Prozesse an ihre Grenzen stoßen, an denen Änderungen nicht noch schneller dem Kunden präsentiert werden können. Daher erfreut sich eine neue Prozessidee, Kanban (IT) (im Folgenden nur „Kanban“), in den letzten Jahren an einer stetig steigenden Beliebtheit. Hiermit soll die Time-to-Market einzelner Änderungen stark reduziert werden, um schneller auf wechselnde Marktanforderungen reagieren zu können.

Ist-Zustand

Derzeit wird die Website smart.com nach den etablierten Scrum-Prozessen entwickelt. Jede Änderung (im Folgenden „Ticket“ genannt) muss hierbei alle definierten Prozessschritte und Status durchschreiten, um vom Endnutzer sichtbar und nutzbar zu sein. Die zu durchschreitenden Status wurden hierbei für jeden einzelnen Prozessschritt in einen aktiven und einen passiven Zustand aufgeteilt. Befindet sich ein Ticket in einem aktiven Zustand, so wird im Moment auch wirklich daran gearbeitet. Wird der Prozessschritt abgeschlossen, wird das Ticket in den nächsten passiven Zustand verschoben, und signalisiert damit, dass es für den nächsten aktiven Prozessschritt zur Verfügung steht. Dies ist für die Spezifikation und ein initiales Schätzen des Aufwands der Tickets in einem Pull-System verwirklicht, in dem jeder Arbeiter die für ihn zur Verfügung stehenden Aufgaben proaktiv beginnt und nicht direkt zugewiesen bekommt.

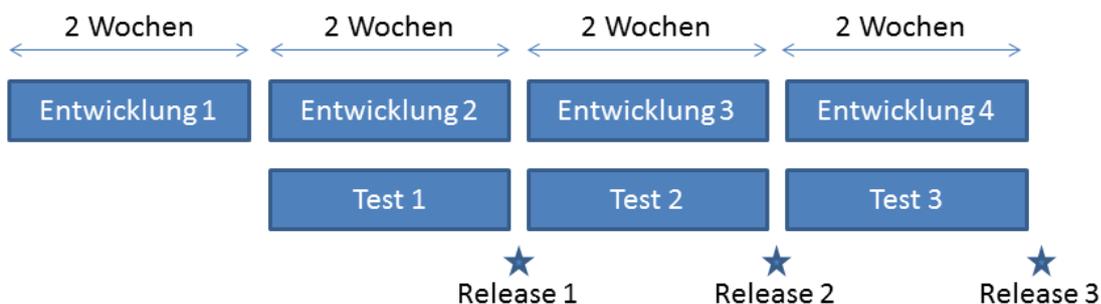


Abbildung 1: Sprint-Release-Zyklen

Für die eigentliche Entwicklung und Umsetzung der Tickets ist dieses Pull-System derzeit jedoch noch nicht realisiert. Hier werden Scrum Sprints mit einer Länge von zwei Wochen durchgeführt. Nach einer zweiwöchentlichen Entwicklungsphase, in der optimalerweise alle geplanten Tickets umgesetzt wurden, folgt eine erneut zwei Wochen lange Testpha-

se (Abbildung 1: Sprint-Release-Zyklen). Hier werden alle umgesetzten Tickets erneut überprüft, um herauszufinden, ob diese Realisierung wirklich, wie geplant und spezifiziert, geschehen ist. Erst wenn in dieser Testphase keine kritischen Fehler auftreten, wird die Erlaubnis gegeben, die durchgeführten Änderungen dem Endnutzer zu präsentieren.

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

Im besten Fall benötigt ein Ticket, nachdem es spezifiziert und geschätzt wurde, vier Wochen, bis es vom Markt gesehen wird. Häufig wird jedoch von Änderungen erwartet, dass diese innerhalb weniger Tage nach dem Aufkommen der Idee auf der Website verfügbar sind. Eine so schnelle Umsetzung ist mit solch einem Prozess mit definierten Sprintlängen nicht möglich.

Lösung

Abhilfe soll daher Kanban bieten. Hierbei werden alle Tickets einzeln betrachtet. Jedes Ticket steht schlussendlich für ein Release. Nach David J. Anderson [1], der als Begründer von Kanban in der IT gilt, muss bei der Nutzung von Kanban auf fünf Kernpraktiken Rücksicht genommen werden:

- Visualisiere den Arbeitsfluss und die Arbeit
- Limitiere den Work in Progress
- Steuere und messe den Arbeitsfluss
- Mache Prozessregeln explizit
- Verbessere durch bewährte Modelle und wissenschaftliche Methoden

Diese Praktiken sollen nun auch in der Entwicklung der smart.com Einfluss finden. Damit jedoch jede kleine Änderung für sich der Öffentlichkeit zugänglich gemacht werden und somit das volle Potential von Kanban ausgenutzt werden kann, muss zuvor noch Continuous Delivery eingeführt werden. Durch Automatisierung von Tests und Softwareauslieferungsprozessen werden hier alle Prozessschritte, die nach der Entwicklung durchgeführt werden müssten, vollautomatisiert ausgeführt. Nach der Entwicklung ist also, überspitzt ausgedrückt, nur noch ein Knopfdruck nötig, um ein neues Feature dem Endnutzer zugänglich zu machen – ein Aufwand, der nur einen Bruchteil der heutzutage notwendigen Arbeit darstellt und damit auch häufiger eingegangen werden kann. Entsprechend der Einführung von Continuous Delivery muss eine geplante Umstellung der Prozesse von Scrum auf Kanban also in zwei Phasen aufgeteilt werden – die erste Phase vor der Einführung von Continuous Delivery, und die zweite Phase danach.

Phase 1

In der ersten Phase sind die technischen Voraussetzungen für eine Umsetzung von Continuous Delivery noch nicht gegeben, weshalb es noch nicht möglich ist das volle Potential, welches Kanban in der Softwareentwicklung bieten kann, abzurufen. Durch die daher noch nur selten möglichen Deployments sollte vorerst eine Mischung aus den aktuellen Scrum-Prozessen und einigen über Kanban eingeführte Neuerungen umgesetzt werden. Diese Mischkultur dient zum einen als

Eingewöhnung an die Arbeitsweise in Kanban, soll jedoch auch schon verschiedene Vorteile gegenüber dem derzeitigen Scrum-Prozess mit sich bringen. Wichtig in einem Kanban-Umfeld ist, dass jedes Ticket, das von einem Arbeiter angegangen wird, auch von diesem selbst gezogen wird. Arbeit wird hier nicht zugewiesen, sondern sie wird von der Person gewählt und bearbeitet, die gerade eben Zeit dafür hat. Daher dürfen auch keine direkten Sprint-Ziele entworfen werden, welche den Entwicklern direkt vorschreiben, welche Arbeit in welcher Zeit abgeschlossen sein muss. Die zwei-Wochen-Sprints werden jedoch aufgrund der nur selten möglichen Deployments vorerst noch beibehalten. Am Ende eines solchen Sprints muss jedoch nicht gezwungenermaßen all das umgesetzt worden sein, was zu Beginn des Sprints definiert wurde, sondern es wird das ausgeliefert, was auch wirklich fertig und qualitativ hochwertig entwickelt wurde. Dieses Verhalten ermöglicht eine schnelle Umpriorisierung auch inmitten eines Sprints, um kurzfristig aufgekommene kritische Tickets problemlos noch in den laufenden Sprint mit aufzunehmen. Für diese kurzfristige Priorisierung der Tickets muss sich auch über die benötigten Meetings Gedanken gemacht werden. Es stellt sich die Frage, ob diese übernommen werden können, oder ob diese durch weitere Meetings ergänzt werden sollten, welche eine schnellere Umpriorisierung der Tickets begünstigen würden. So wäre zum Beispiel die Einführung eines Daily-Standup-Meetings, in welchem bestehende Probleme und potentielle Umpriorisierungen von den Stakeholdern angesprochen werden sollen, eine gute Idee. Außerdem sollten Retrospektiven beziehungsweise Kaizen-Meetings, in welchen grundlegende und prozessuale Probleme angesprochen und nach nachhaltigen Lösungen dieser gesucht werden sollten, eingeführt werden. Bei diesem Meeting handelt es sich um das eigentliche Herzstück einer Kanban-Kultur, welche eine kontinuierliche Verbesserung des Arbeitsflusses anstrebt. Ein weiterer wichtiger Teil von Kanban stellt die Visualisierung und Begrenzung der aktuell laufenden Arbeit dar.

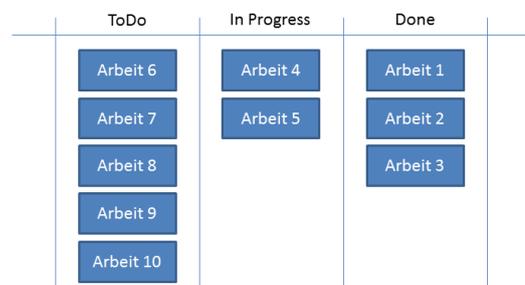


Abbildung 2: einfaches Kanban Board

Im einfachsten Fall hat ein Kanban-Board drei Spalten (Abbildung 2: einfaches Kanban Board), in denen die Zustände, in denen sich die aktuell bearbeiteten Tickets befinden, so-

fort erkennen lassen. Bewegt sich ein Ticket für einige Zeit nicht, fällt hier schnell auf, dass eine Blockade besteht. Im Fall der smart.com werden hier vorerst die einzelnen Prozessschritte als Spalten herangezogen. Außerdem sollte die Work-in-Progress, also jene Arbeit, die gerade bearbeitet wird, explizit begrenzt werden, wodurch ein zu häufiges Wechseln der Aufgaben und damit ein unnötiger Zeitverlust beim Umsetzen vermieden wird. In Verbindung mit dem zuvor erwähnten Kanban-Board bieten solche WiP-Limits jedoch noch den Vorteil, dass Engpässe im Prozess deutlich sichtbar werden. Mit Serviceklassen, verschiedenen Metriken, Arbeitstypen und -größen und weiteren Hilfsmitteln aus dem Kanban-Umfeld liegen hier noch weitere Möglichkeiten, die auch schon in diesem frühen Stadium der Prozessänderung viele Vorteile der Priorisierung und Schätzungen bieten können, offen, welche auch in dieser Bachelorarbeit tiefergehend behandelt werden.

Phase 2

Die zweite Phase der Prozessänderungen kann beginnen, sobald alle technischen Voraussetzungen für die Umsetzung von Continuous Delivery gegeben sind. Es wird also angenommen, dass nach der Entwicklung eines Tickets dieses ohne nennenswerten Aufwand automatisch in vollem Umfang getestet

und bei erfolgreichem Test direkt dem Endnutzer verfügbar gemacht werden kann. Nun kann sich über die einzelnen Prozessschritte, die von einem Ticket durchlaufen werden, Gedanken gemacht werden. Ein wichtiger Ansatz, um die Durchlaufzeiten und die Übersichtlichkeit der Prozesse zu gewährleisten und zu verbessern, beschreibt das KISS-Prinzip, welches zu der einfachsten Lösungsmethode rät. Daher sollten alle überflüssigen Prozessschritte abgeschafft werden. Durch die Automatisierung der Tests, stellt sich zum Beispiel die Frage, ob alle testbezogenen Prozessschritte irrelevant sind und aus dem Life-Cycle eines Tickets entfernt werden können. Weitere Überlegungen betreffen hierbei zum Beispiel auch Schätzungen und verschiedene Abnahmeprozesse. Der wichtigste Teil der zweiten Phase ist jedoch die Automatisierung der Softwareauslieferungsprozesse, welche zur Folge hat, dass Sprintzyklen überflüssig werden. Jede noch so kleine Änderung kann als eigenes Release angesehen und behandelt werden. Solch ein Verhalten führt zu bedeutend geringeren Durchlaufzeiten und weniger Blockaden. Natürlich müssen auch hier wieder die verschiedenen Meetings überdacht werden. So muss die Review zum Beispiel aufgrund der kontinuierlichen Deployments deutlich häufiger durchgeführt werden. Jedoch ziehen diese häufigeren Auslieferungen auch noch weitere prozessuale Änderungen mit sich, die in dieser Bachelorarbeit behandelt werden.

[1] David J. Anderson, 2007

Bildquellen:

- Abbildung 1: Eigene Abbildung: Sprint-Release-Zyklen
- Abbildung 2: Eigene Abbildung: einfaches Kanban Board

Entwicklung und Umsetzung eines Konzepts zum Data Management von referenz- und applikationsspezifischen Daten für iDome-Applikationen

Konstantinos Papadopoulos*, Manfred Dausmann, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Laut einer Studie von ClusterHQ [1] verbringen 43 Prozent der Softwareentwickler bis zu 25 Prozent ihrer Zeit mit der Fehlersuche in Systemen, die bereits in Produktion laufen. Eine Ursache dafür ist die Unfähigkeit, die Produktumgebungen neu zu erstellen. So kann es passieren, dass Daten- und Konfigurationsreste übrig bleiben und so fälschlicherweise Fehler entstehen. Weitere Probleme entstehen bei den Integrationstests, in denen die gegenseitige Abhängigkeit externer Systeme die Ursache für Bugs ist. Tests, die mit unrealistische Daten laufen, können ebenfalls zu Problemen führen. So werden oft Randfälle nicht betrachtet und führen letztendlich zu Softwarefehlern.

Problemstellung

Diese Bachelorarbeit wurde für die inTension GmbH angefertigt und dient dazu, einfacher realistische Daten zu erstellen. Ziel war es, einen geeigneten Mechanismus zu entwerfen und zu entwickeln, um von einer Anwendung benötigte Daten sinnvoll zu definieren und in eine Datenbank schreiben zu können, die außer dem Schema keine Daten enthält. Hierbei spielt es keine Rolle, ob die Daten für Testumgebungen genutzt werden sollen, für die Initialisierung einer neuen Software beim Kunden benötigt werden oder ob es relevante statische Daten für den Produktionsbetrieb sind. Hierfür sollen die Daten in einer einfachen und lesbaren Form in einer Datei geschrieben werden. Es sollte möglich sein, diesen Mechanismus in einer Continuous Integration Umgebung zu nutzen und auch aus einer Entwicklungsumgebung heraus auszuführen. Weiterhin musste analysiert und ein Prozess entworfen werden, wie versionsspezifische Teile einer Datenbank-Konfiguration oder auch nur einzelne, ganze oder partielle Tabelleninhalte im Rahmen von Deploymentprozessen automatisch von einer Umgebung auf eine andere weiter propagiert werden können.

Analyse

Es musste ein Dateiformat, das die Daten enthält, ausgewählt werden. Da im Vorfeld nicht bekannt war, welche Beziehungen unterstützt werden sollten, wurde entschieden, dass das einzusetzende Framework Beziehungen zwischen Datenobjekten definieren sollte. So konnte jeder Beziehungstyp abgedeckt werden. Anschließend musste der Weg, wie die Daten in die Datenbank eingespeist werden, geklärt werden. Danach musste die Versionierung der Dateien innerhalb eines Continuous Integration Prozesses und deren Weitergabe in andere Umgebungen geklärt werden. Zum Schluss mussten noch die Parameter definiert werden, die dem Tool beim Start mitgegeben werden, um beispielsweise die Ordnerstruktur mit den Dateien finden zu können.

Realisierung

Die Wahl des Dateiformats fiel auf das JSON-Format. Es ist ein kompaktes Datenformat in einer einfach lesbaren Textform. Jede Tabelle, die mit Daten befüllt werden soll, besitzt eine eigene JSON Datei. In den Dateien wird der jeweilige Entitytype-Name definiert und die Entitäten angelegt. Eine Entität wird dabei zeilenweise definiert, indem das Attribut gefolgt von seinem Wert gespeichert wird. Da die Datenstruktur schlank gehalten und lediglich die Attribute gespeichert werden sollen, bekommen die Zeilen in den JSON-Dateien keinen „internen“ eindeutigen Bezeichner.

Um eine Relation zu definieren, werden die Zeilennummern der betroffenen Einträge hinzugezogen. So wird eine 1-zu-n Beziehung zwischen Datensätzen, symbolisch durch ein Hashtag gefolgt von der Zeilennummer des zu referenzierenden Dateneintrags definiert. Das bedeutet beispielsweise, wenn eine Person mehrere Accounts besitzt, hat der Account ein Attribut mit dem Namen „owner“, indem die verlinkte Person stehen soll. Steht nun im Attribut der Wert „#1“, bedeutet das, dass der

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

erste Eintrag in der JSON-Datei der Person dort verlinkt ist. Bei einer n-zu-m Relation werden hintereinander mehrere Hashtags gefolgt von der Zeilennummer gesetzt.

Ein Laden kann beispielsweise verschiedene Produkte haben, genauso kann aber das Produkt auch in verschiedenen Läden verkauft werden. Das Attribut „läden“ des Produkts wird den Wert „#1#2#3“ haben und das Attribut „produkte“ des Ladens den Wert „#4#5#6“. Die Überführung der Daten in die Datenbank findet über die Businesslogik von iDome statt. Hier gibt es den Vorteil, dass iDome mit seinen Entity Objekten einen unabhängigen Zugriff auf das Datenbankschema und die Datentypen der Attribute hat. Wird das oben genannte Beispiel der 1-zu-n Beziehung genommen, dann wird im Attribut „owner“ des Accounts einfach das Entity Objekt der zutreffenden Person gesetzt. Bei der n-zu-m Beziehung wird in den Attributen jeweils ein Set von Entitäten abgelegt. iDome löst diese Beziehungen anschließend richtig auf und speichert alles in die richtigen Tabellen der Datenbank ab. Der oben beschriebene Prozess der Datenverarbeitung wurde in Abbildung 1 als Prozessor zusammengefasst.

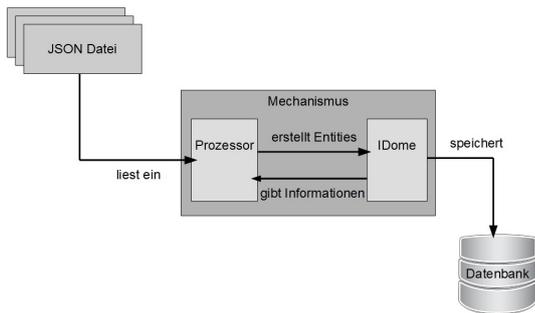


Abbildung 1: Mechanismus zur Datenverarbeitung

Die Versionierung der JSON Dateien kann über Maven stattfinden. Ein Maven Projekt enthält stets eine aktuelle Versionsnummer. Diese kann ausgelesen und benutzt werden, um die JSON Dateien zu versionieren. Sobald der Continuous Integration Prozess gestartet wird, wird mit der aktuellen Versionsnummer ein Ordner erstellt. Wurden die Datensätze aus den JSON Dateien in die Datenbank geschrieben, werden die Dateien in den neu erstellten

Ordner kopiert. Die alten Dateien werden anschließend bis auf das Grundgerüst des JSON-Formats geleert, um für den nächsten Build-Prozess verfügbar zu sein. Hier müssen dann lediglich die neu hinzugekommenen Datensätze eingefügt werden. Damit die Komplexität der Konfiguration minimal bleibt, wurde das Softwaredesign-Paradigma „Konvention vor Konfiguration“ [2] benutzt.

Es wurde eine Standardkonfiguration definiert, damit die JSON Dateien und die Konfigurationsdateien für den iDome Start automatisch gefunden werden. Trotzdem wurden Parameter für den Pfad zu den JSON Dateien, sowie den Konfigurationsdateien für den iDome Start eingeführt. Denn so ist das Tool flexibler und kann auch bei älteren Projekten, die keine Standardform besitzen, zum Einsatz kommen.

Fazit

Ziel der Bachelorarbeit war es einen geeigneten Mechanismus zu entwerfen und zu entwickeln, um von einer Anwendung benötigte Daten sinnvoll zu definieren und in eine Datenbank schreiben zu können. Es sollte möglich sein, diesen Mechanismus in einer Continuous Integration Umgebung zu nutzen und auch aus einer Entwicklungsumgebung heraus auszuführen. Außerdem sollten die Dateien versioniert werden können und in verschiedene Deployprozesse durchpropagiert werden. Dabei ergab sich, dass das JSON-Format sehr gut als Datenformat geeignet ist, denn es ist ein kompaktes Datenformat in einer einfach lesbaren Textform.

Was die Datensätze betrifft, so konnten diese mit Hilfe von den Entity Objekten von iDome sehr einfach in die Datenbank geschrieben werden. Bei der Versionierung wurde eine einfache Struktur gewählt, in der die JSON Dateien pro Continuous Integration Durchlauf in einen Ordner mit der aktuellen Versionsnummer als Namen kopiert werden. Bei den Parametern mussten einerseits Defaultwerte definiert werden, aber andererseits musste das Tool flexibel genug gehalten werden, um auch in älteren Projekten zum Einsatz zu kommen.

[1] <https://clusterhq.com/2016/11/03/devops-testing-survey/> (Stand: 03. November 2016)

[2] https://de.wikipedia.org/wiki/Konvention_vor_Konfiguration (Stand: 19. Juni 2015)

Bildquellen:

- Abbildung 1: Eigene Darstellung

Untersuchung und Anwendung von Algorithmen in Matlab zur Erkennung von Verkehrsschildern für das autonome Fahren

Christian Rautenberg*, Karlheinz Höfer, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Was ist eigentlich autonomes Fahren?

Autonomes Fahren beschreibt ein Fahrzeug, das sich komplett selbstständig fortbewegen kann, ohne dass der Mensch aktiv in die Bewegung des Fahrzeuges eingreift. Dabei kann es sich um einen normalen Personenkraftwagen, einen Lastkraftwagen oder ein Modellfahrzeug handeln. Ein populärer Teilbereich des autonomen Fahrens ist die Einparkhilfe. Hierbei kann ein Fahrzeug, fast ohne menschliches Einwirken, selbstständig einparken. Die nächste Stufe des autonomen Fahrens ist, dass das Fahrzeug komplett ohne Eingreifen des Menschen eine gewisse Strecke zurücklegen kann. Hierbei ist das Fahrzeug in der Lage, die Verkehrsschilder und Ampeln zu erkennen und daraufhin richtig zu reagieren und weiter noch Verkehrsteilnehmer, Passanten und Hindernisse wahr zu nehmen und diesen auszuweichen. Mit bestimmten Fahrzeugen der Marke Tesla ist dies heute schon durch den Autopilot auf deutschen Straßen möglich. Überschattet wird der Autopilot von Tesla [1] jedoch durch einen tragischen Unfall. Ein Tesla Fahrzeug hatte dabei einen Lkw für ein Straßenschild gehalten und aufgrund dessen falsch reagiert, wodurch der Insasse starb [2].

Nachteile des autonomen Fahrens

Der tragische Unfall des Tesla zeigt, dass die Sensoren noch nicht komplett fehlerfrei arbeiten und man weiterhin aufpassen muss, wie das Fahrzeug sich verhält um kurzfristig reagieren zu können. Ein Nachteil kann auch eine schlecht geschützte Funkkommunikation sein, die Hacker ausnützen könnten, um einem Fahrzeug falsche Daten zu übermitteln, die daraufhin auf Grundlage dieser fehlerhaften Informationen falsch reagieren und Unfälle verursachen.

Vorteile des autonomen Fahrens

Laut einer Statistik von Google sind 94 % aller Verkehrsunfälle durch menschliche Fehler verursacht worden [3]. Seien es Ablenkungen oder das Missachten von Verkehrsregeln. Einem autonomen Fahrzeug werden solche Fehler nicht passieren, sofern die Soft.- und Hardware fehlerfrei funktioniert. Ist die Funkkommunikation

gut geschützt, kann die Kommunikation zwischen den Fahrzeugen große Vorteile bieten. Beispielweise fahren Future Trucks von Daimler [4] nicht nur autonom, sie sind auch in der Lage untereinander zu kommunizieren. Diese Technik hat den Vorteil, dass die Fahrzeuge enger auffahren können und somit keine nutzlosen Lücken entstehen. Weiter haben somit Drängler, die versuchen durch schnelleres Fahren andere zu provozieren um schneller voran zu kommen, keine Chance mehr. Hierdurch können Staus vermieden werden, da der Verkehr flüssiger laufen kann [5].

Anforderungen

Die Anforderungen an solch ein System sind sehr hoch. Die Sicherheit der Kommunikation der Fahrzeuge muss so geschützt werden, dass sich kein Dritter in die Kommunikation hacken kann und somit das Fahrzeug manipuliert. Auch an die Hardware sind sehr hohe Anforderungen gestellt. Wo ein einfaches Radar mit seinem Sichtwinkel von ca. 130° ausreichen würde um die Fahrzeuge voraus zu erkennen, muss aus Sicherheitsgründen und der vollständigen Abdeckung des Sichtbereichs noch zwei weitere Radars verbaut werden. Das Radar muss außerdem dazu im Stande sein Personen von Objekten unterscheiden zu können. Da ein Radar keine Schilder lesen kann, muss zusätzlich noch eine Kamera verbaut werden. An diese Kamera werden wiederum ebenfalls sehr hohe Anforderungen gestellt. Diese darf weder ausfallen noch blockiert sein. Allein dichter Nebel würde ausreichen Schilder und Personen nicht mehr rechtzeitig erkennen zu können. Jedoch kann das Radar im Notfall zumindest Personen rechtzeitig erkennen, sofern diese nicht auch ausfällt.

Beschreibung der Bachelorarbeit

In dieser Bachelorarbeit geht es darum verschiedene Algorithmen zu untersuchen, um festzustellen, welcher am besten dafür geeignet ist, Verkehrsschilder zu erkennen. Hierfür wird versucht, 12 verschiedene Verkehrsschilder zu erkennen. Diese müssen für den Carolo-Cup erkannt werden, damit das Fahr-

*Diese Arbeit wurde durchgeführt an der Fakultät Informationstechnik

zeug am Wettbewerb teilnehmen darf. Bei der Untersuchung wird darauf geachtet, welcher Algorithmus schnell genug ist, um das jeweilige Verkehrszeichen zu erkennen. Meist hat man nur wenige Sekunden von der Erkennung des Schildes bis zur Reaktion darauf.

Was ist der Carolo-Cup eigentlich?

„Der Hochschulwettbewerb „Carolo-Cup“ bietet Studententeams die Möglichkeit, sich mit der Entwicklung und Umsetzung von autonomen Modellfahrzeugen auseinander zu setzen. Die Herausforderung liegt in der Realisierung einer bestmöglichen Fahrzeugführung in unterschiedlichen Szenarien, die sich aus den Anforderungen eines realistischen Umfelds ergeben.“[6]

Grundlegendes Vorgehen

Es sollen Algorithmen wie „imfindcircles“ getestet werden. Hierbei lässt sich zuerst ein Kreis erkennen. Mit einem Farbgleich lässt sich daraufhin bestimmen, um welches Schild es sich genauer handeln könnte. Eine weitere Möglichkeit wäre das Lernen und Speichern der Verkehrszeichen in ein künstliches neuronales Netz. Hierbei werden gleiche Schilder aus verschiedenen Winkeln, Größen und Fehlern gelernt, so dass dieses Netzwerk bei Übereinstimmung das Schild erkennen kann. Auch besteht die Möglichkeit eine über ein theoretisches Lernen mit Markov-Ketten eine

Wahrscheinlichkeitstabelle für die jeweiligen Schilder aufzustellen, da sich zwar viele Schilder ähneln aber an vielen Punkten wie Farbe, Form und Formen auf dem Schild unterscheiden.

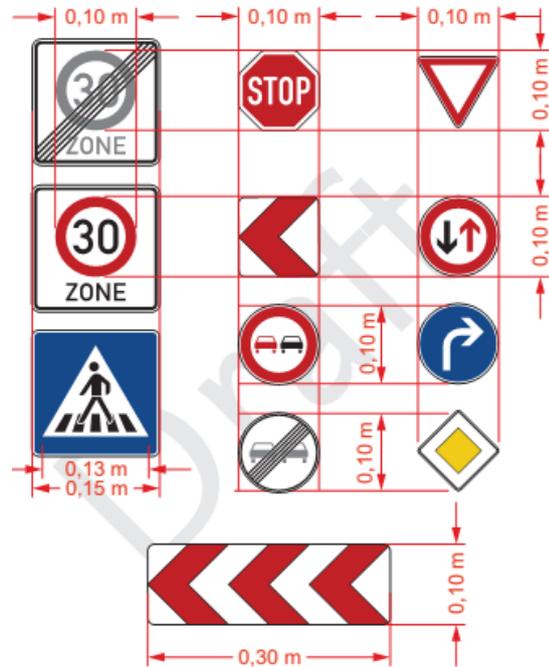


Abbildung 1: Verkehrschilder

[1] Tesla setzt voll auf Roboterautos,

<http://www.sueddeutsche.de/auto/elektroautos-tesla-baut-technik-fuer-autonomes-fahren-in-alle-modelle-ein-1.3214038>, 02.12.2016

[2] Tesla-Autopilot hielt Lkw für Verkehrsschild,

<http://www.zeit.de/mobilitaet/2016-07/autonomes-fahren-tesla-unfall-model-s-autopilot-software>, 02.12.2016

[3] Google Self-Driving Car Project, <https://www.google.com/selfdrivingcar/>, 02.12.2016

[4] Autonom unterwegs, <https://www.daimler.com/innovation/autonomes-fahren/>, 02.12.2016

[5] Wie entsteht eigentlich ein Stau? Fünf grobe Fehler der Autofahrer,

http://www.t-online.de/ratgeber/auto/fahrverhalten/id_48139086/wie-entsteht-eigentlich-ein-stau-fuenf-grobe-fehler-der-a.html, 02.12.2016

[6] admin, Wettbewerbsziel, <https://wiki.ifr.ing.tu-bs.de/carolocup/carolo-cup>, 02.12.2016

Bildquellen:

- Abbildung 1: TU Braunschweig, Carolo-Cup Regelwerk 2017, S. 43

„Favour Composition over Inheritance“

Alessandro Valentino Ribul Moro*, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

„Favour Composition over Inheritance“ (FCOI) ist ein Prinzip der Softwareentwicklung, das bei polymorphem Code in Konkurrenz zur Vererbung steht. Übersetzt heißt das Prinzip: „Ziehe Objektkomposition der Klassenvererbung vor“. Der Name nennt zwar eine Komposition, es ist aber eine Aggregation einer Abstraktion zu verwenden. Zur Laufzeit tritt bei FCOI dann an die Stelle der Abstraktion ein Objekt, dessen Klasse die Abstraktion implementiert hat. Der Vorteil dieses Prinzips ist, dass Abhängigkeiten durch Vererbung vermieden werden und die Verknüpfungen und Abhängigkeiten von konkreten Objekten erst zur Laufzeit gesetzt werden. Bei der Vererbung wird die Verbindung zwischen Basisklasse und abgeleiteter Klasse bereits während der Kompilierzeit festgelegt. Sie kann während des Betriebs nicht mehr geändert werden. Das Prinzip FCOI dient zur Reduktion von Abhängigkeiten und wurde bereits 1994 im Buch „Design Patterns. Elements of Reusable Object-Oriented Software“ [1] der „Gang of Four“ beschrieben. Zusammen mit dem Prinzip „Programmiere gegen Schnittstellen und nicht gegen Implementierungen“ stellt dieses Prinzip die Grundlage für die berühmten Entwurfsmuster von Gamma et al. dar.

Vererbung vs. Objektkomposition

Für die Erzeugung von Polymorphie kennt die Objektorientierung zwei generelle Mechanismen:

- die statische Vererbung (Klassenvererbung) und
- die Aggregation einer Abstraktion (die sogenannte „Objektkomposition“).

Vergleicht man die beiden Mechanismen miteinander, so stehen sich grundlegende Prinzipien gegenüber. Bei der Vererbung wird durch Unterklassenbildung eine **White-Box-Wiederverwendung** zur Kompilierzeit genutzt. Das heißt im Falle der Vererbung, dass die Daten und Methoden einer Oberklasse für die Unterklasse infolge der Vererbung zum Teil sichtbar sind, nämlich dann, wenn sie in der Basisklasse nicht mit dem Zugriffsmodifikator `private` geschützt sind. Dagegen

ist die Objektkomposition eine **Black-Box-Wiederverwendung**. Hierbei referenziert eine entsprechende Klasse eine Abstraktion, die polymorph implementiert werden kann. Es werden keine internen Details der aggregierenden Klasse sichtbar und somit sieht man nicht ins Innere der schwarzen Box.

Im Allgemeinen:

White-Box-Wiederverwendung -> Wiederverwendung der Implementierung.

Black-Box-Wiederverwendung -> Wiederverwendung der Schnittstelle.

Liskovsches Substitutionsprinzip

Das liskovsche Substitutionsprinzip [2] (LSP) ist Grundlage der Objektkomposition und findet auch in der Vererbung Anwendung. LSP beschreibt eine Bedingung der objektorientierten Programmierung. Die Forderung lautet: Ein Programm mit Referenzen auf Objekten einer Basisklasse darf den Austausch eines Objektes einer Basisklasse durch das einer abgeleiteten Klasse nicht bemerken. Das heißt, Kundenobjekte dürfen es nicht bemerken, dass an der Stelle eines Objekts der Basisklasse ein Objekt einer abgeleiteten Klasse steht. Beispiel: Das Objekt der Basisklasse A wird durch das Objekt der abgeleiteten Klasse B ersetzt, ohne dass der Vertrag der Basisklasse gebrochen wird. In Verbindung mit Design by Contract [3] von Bertrand Meyer können Vorbedingungen nur aufgeweicht und Nachbedingungen verschärft werden.

Vorgehensweisen

Bei der **Vererbung** tritt zur Kompilierzeit eine abgeleitete Klasse, die das LSP einhält, an die Stelle einer Basisklasse. Durch die Vererbung entsteht eine starke Abhängigkeit einer abgeleiteten Klasse von ihrer zugehörigen Basisklasse, wenn die Elemente der Basisklasse in der abgeleiteten Klasse sichtbar sind. Bei der **Objektkomposition** kann man zwischen zwei Vorgehensweisen wählen. Bei der ersten Vorgehensweise kann als Abstraktion eine Schnittstelle verwendet werden. Dann hängt das Objekt, welches zur Laufzeit an die Stel-

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

le der Schnittstelle tritt, nur vom Vertrag der Schnittstelle ab. Dagegen kann bei der zweiten Vorgehensweise eine abstrakte Klasse als Abstraktion benutzt werden. Bei deren Verwendung hat man jedoch wieder das Problem der Erzeugung von Abhängigkeiten durch Ver-

erbung. Allerdings kann man die Objekte dieses Typs zur Laufzeit austauschen und muss nicht neu kompilieren.

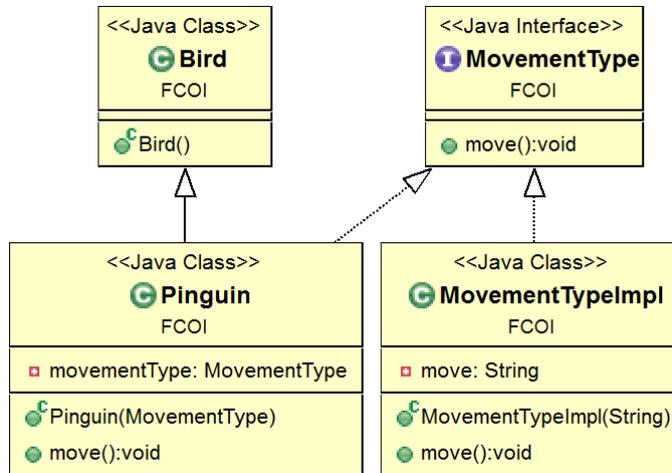


Abbildung 1: UML-Diagramm Objektkomposition

Abbildung 1 zeigt die Vererbung der Klasse Pinguin, die eine abgeleitete Klasse der Basis-Klasse Bird ist. Zudem die Objektkomposition bei der zur Laufzeit ein Objekt einer Klasse genutzt wird, welches das Interface Movement-Type implementiert.

Vorteile von FCOI

Vorteile, die FCOI allgemein und gegenüber der Vererbung aufweist, sind:

- **Einhaltung der Datenkapselung**
Durch die Black-Box-Sicht wird die Kapselung der Klassen geschützt, da die Objekte ausschließlich über die Abstraktion ihrer Klasse verwendet werden.
- **Bindung zur Laufzeit statt zur Kompilierzeit**
Bei der Objektkomposition erfolgt der Austausch der Abstraktionen zur Laufzeit. Bei der Vererbung werden die Bindungen schon zur Kompilierzeit festgelegt.
- **Simulation von Mehrfachvererbung**
Programmiersprachen, die nur eine Einfachvererbung unterstützen, können durch die Verwendung von Objektkomposition eine Mehrfachvererbung simulieren.
- **Leichteres Testen**
Durch die Verwendung einer Abstraktion kann man beim Testen leichter mit Stubs oder Mock-Objekten arbeiten.

Nachteile von FCOI

Nachteile von „Favour Composition over Inheritance“ sind:

- **Keine direkte Wiederverwendung von Code**
Vererbung führt zu einer direkten Wiederverwendung von Programmcode, falls er nicht überschrieben wird. Bei der Objektkomposition muss jede API implementiert werden, auch wenn dabei nur eine Delegation zum Einsatz kommt.
- **Rasche Zunahme der Schnittstellen**
Durch die Erwartung der Objektkomposition, dass aggregierende Objekte eine Schnittstelle als Abstraktion verwenden, müssen diese Schnittstellen natürlich bereitgestellt werden. Da das „Interface-Segregation Principle“ fordert, dass Schnittstellen jeweils nur an einen Client angepasste Aufgaben haben, wächst die Anzahl der Schnittstellen mit jedem Objekt stetig.

Trotz der vielen Vorteile, die die Objektkomposition bietet, gibt es auch Anwendungsmöglichkeiten, in denen die statische Vererbung von Vorteil ist.

Statische Vererbung ist sinnvoll, wenn:

- tatsächlich eine „is a“-Beziehung modelliert werden soll und es sich nicht nur um Wiederverwendung von Programmcode handelt,
- sich der polymorphe Anteil nicht lokalisieren und heraustrennen lässt,
- die Klassenhierarchie einfach bleibt und
- durch Umgestaltung der Basisklasse alle abgeleiteten Klassen verändert werden sollen.

Fazit

Die Objektkomposition weist sehr viele Vorteile gegenüber der Vererbung auf, jedoch wird in der Praxis meist die Vererbung verwendet. Hier spielen zwei Faktoren eine Rolle. Zum einen wird Polymorphie in der Regel anhand der Vererbung gelehrt, zum anderen wird die Vererbung durch die Unterstützung der Programmiersprache leicht gemacht. Entscheidend für die praktische Anwendbarkeit

des Prinzips FCOI ist, dass es möglich ist, einen Teil der Klasse, der polymorph auftreten kann, zu lokalisieren und als Abstraktion herauszutrennen. Dieser Teil ist dann separat zum restlichen Teil zu betrachten, jedoch noch immer über eine Aggregation mit diesem verbunden. Dieser herausgetrennte Teil der Klasse wird als Abstraktion referenziert und kann polymorph realisiert werden.

-
- [1] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: „Design Patterns: Elements of Reusable Object-Oriented Software“. Addison-Wesley, 1994
 - [2] Barbara H. Liskov, Jeannette M. Wing: Family Values: A Behavioral Notion of Subtyping. Pittsburgh 1993.
 - [3] Meyer, Bertrand: Design by Contract, Technical Report TR-EI-12/CO, Interactive Software Engineering Inc., 1986

Bildquellen:

- Abbildung 1: Eigene Darstellung

Analyse und Lösung von OpenAI Environments mit Hilfe von Künstlichen Neuronalen Netzen und Reinforcement Learning

Chris Schäfer*, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Während viele Personen und vor allem Unternehmen der Entwicklung Künstlicher Intelligenz großes Potential für die Bewältigung technischer und wirtschaftlicher Probleme zuschreiben, ruft sie bei anderen Unbehagen hervor. IT Größen, wie etwa Bill Gates, oder anerkannte Wissenschaftler wie Steven Hawking warnen sogar vor einer unkontrollierten Entwicklung der KI. Der CEO der Firma Tesla, Elon Musk, setzte sich bei der Gründung des OpenSource Projektes OpenAI, seinen Angaben zufolge, deßhalb zum Ziel, die künstliche Intelligenz in eine gemeinnützige und nicht profitorientierte Richtung zu lenken[1]. Das Ziel der Bachelor Arbeit ist die Analyse und Lösung von OpenAI Environments mit Hilfe der KI Technologien der Künstlichen Neuronalen Netze und des Reinforcement Learnings. Es sollen geeignete Strategien und Kombination dieser Verfahren verwendet werden.

diese Videos hochgeladen werden können. Anschließend erfolgt eine Evaluierung der Lernfähigkeit des Agenten gemäß den Ansprüchen der Environments. Die Evaluierungen können jederzeit über einen Link aufgerufen und heruntergeladen werden. Bei Bedarf kann auch der Sourcecode zur Verfügung gestellt werden, um andere Entwickler beim Erstellen ihrer Agenten zu unterstützen. Das Bereitstellen des Sourcecodes ist von OpenAI gewünscht und soll zur generellen Entwicklung der KI beitragen[1].

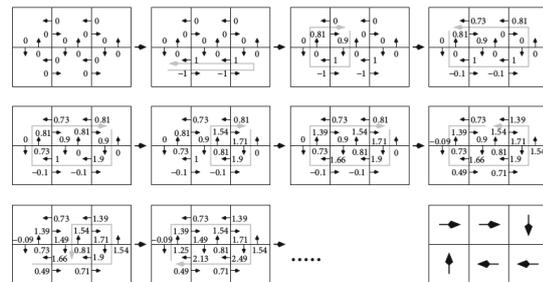


Abbildung 2: Q-Learning Vorgang, Zustände, Zustandsübergänge und zugehörige Q-Werte



Episode 0

Abbildung 1: Das OpenAI Environment CartPole-v0

OpenAI stellt eine Plattform zum Testen und zur Evaluierung von KI Methoden bereit. Diese Plattform besteht aus zwei Komponenten. Die OpenAI Gym library und dem OpenAI Gym service. Die Gym library umfasst eine Vielzahl an Umgebungen(Environments), für deren Lösung intelligente Software Agenten(autonom agierende Computer Programme) entwickelt werden sollen. Die Resultate der Agenten können in Form von Videodateien aufgenommen und gespeichert werden. Der OpenAI Service umfasst die Homepage von OpenAI, auf welcher

OpenAI Gym. Python ist die vom OpenAI Gym aktuell einzig unterstützte Programmiersprache. Weitere Schnittstellen zu anderen Programmiersprachen sollen jedoch folgen. In der Trainingsumgebung können Agenten an verschiedenen Environments ihre Lernfähigkeit testen. Das Gym umfasst elf Kategorien mit insgesamt über 100 Environments. Unter den Kategorien befinden sich Klassiker wie ATARI®und DOOM®. In der Kategorie ATARI®müssen Agenten ATARI®Spiele der 80er Jahre, wie etwa PacMan oder Breakout bewältigen. Jedes dieser Environments hat andere Anforderungen mit jeweils anderen Inputs und Outputs. Die Inputs werden in Form eines Action Space realisiert, ein Array, welches alle Aktionsmöglichkeiten beinhaltet. Das Observation Space beinhaltet die Outputs, welche sich nach Ablauf von Zeiteinheiten verändern können. Die Aufgabe des zu entwickelndem Agenten ist es, aus den Outputs des Environments selbstständig Entscheidungen zu treffen

*Diese Arbeit wurde durchgeführt an der Fakultät Informationstechnik

und diese durch Ausführung von Aktionen zu realisieren. Ziele. In erster Linie soll das Ziel des Entwicklers sein, einen Agenten zu entwickeln, der in der Lage ist ein Environment zu lösen. Ein übergreifenderes Ziel ist es, einen Agenten zu entwickeln, welcher in der Lage ist

mehrere Environments ohne bzw. mit wenigen Codeänderungen zu erfüllen. Dies entspräche dem Verständnis des Deep Learnings[1].

Aktion Zustand	A	B	C	D	E	F
A	-	0	-	0	-	-
B	0	-	0	-	0	-
C	-	0	-	-	-	0
D	0	-	-	-	0	-
E	-	0	-	0	-	0
F	-	-	0	-	0	-

Aktion Zustand	A	B	C	D	E	F
A	-	1.39	-	-0.09	-	-
B	0.73	-	1.54	-	1.49	-
C	-	1.39	-	-	-	1.71
D	1.25	-	-	-	0.49	-
E	-	0.81	-	2.13	-	0.71
F	-	-	1.54	-	2.49	-

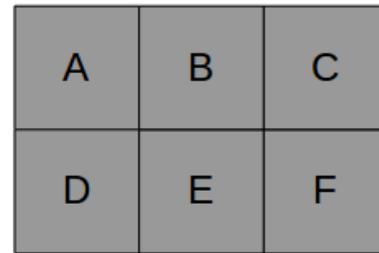


Abbildung 3: Zu Abbildung 2 zugehörige Q-Matrix, Anfangszustand oben, Konvergierte Matrix unten

CartPole. Das Environment Cartpole(-v0/-v1), in Abbildung 1 dargestellt, soll als eine Experimentelle Basis der Bachelor Arbeit dienen. Das Environment besteht aus einem zweidimensionalen Raum, in welchem ein Rechteck, das Cart, durch das Zuführen von horizontalen Kräften nach links oder rechts bewegt werden kann. Auf diesem Cart ist in der Mitte ein weiteres längliches Rechteck angebracht, welches durch den Agenten balanciert werden soll. Da das Model den Gesetzen der Trägheit und der Schwerkraft folgt, wird das längliche Rechteck ohne Einwirken des Agenten nach einer kurzen Zeit nach unten fallen.

Erreicht das längliche Rechteck einen Winkel, der mehr als 15 Grad zur vertikalen Achse abweicht, gilt die Episode(ein Versuch) als beendet. Eine weitere Herausforderung des Environments ist die Begrenzung der Position. Überschreitet die Position des Carts den Wert 2.4 ,bzw. -2.4, gilt die Episode ebenfalls als beendet. Das Environment gilt als bestanden(solved), wenn das längliche Rechteck über 100 Episoden hinweg mit einem Durchschnittswert von 195 Zeiteinheiten balanciert wurde[2].

Vorgehensweise. In der aktuellen Phase der Bachelorarbeit, wird versucht das Environment Cartpole mit Hilfe der Reinforcementstrategie des Q-Learnings zu lösen. Das

Q-Learning benötigt zur Ausführung ein Zustandsmodell der Umgebung. Da Cartpole jedoch keine Zustände sondern lediglich physikalische Werte, wie Position, Winkel des länglichen Rechtecks, Geschwindigkeit des Carts und Winkelgeschwindigkeit des länglichen Rechtecks als Output ausgibt, ist es nötig, die Umgebungswerte in Zustände umzuwandeln.

Um dieses Problem zu lösen, wird eine Matrix erstellt, welche alle Zustände beinhaltet. Desweiteren werden für das Durchführen des Q-Learning Algorithmus Informationen über die Zustandsübergänge zwischen den Zuständen benötigt. Dafür wird eine Zustandsübergangsmatrix angelegt. Zustandsübergänge sind Aktionen, die in einem Zustand x ausgeführt werden, um in einen anderen Zustand y überzugehen. Zustandsübergänge müssen vom Agenten jedoch vorab erst erkundet werden. Der Agent wird in der Exploration Phase durch zufällige Aktionen versuchen so viele Zustände wie möglich anzunehmen, um die Zustandsübergänge zu bestimmen.

In Abbildung 2 werden Zustände und ihre zugehörigen Übergänge und Q-Werte dargestellt. Das „Gehirn“ des Agenten stellt die Q-Marix dar(Abbildung 3), die in ihrer Dimension der Zustandsübergangsmatrix gleicht. In

ihr werden beim Erreichen eines Ziels Q-Werte gemäß einer Q-Gleichung aufeinander addiert. So wird der Agent im Laufe der Zeit lernen, welcher Zustandsübergang ihn zu seinem Ziel führt. Eine Evaluation des aktuellen Agenten

der Bachelorarbeit, welcher den Q-Learning Algorithmus implementiert hat, wird in Abbildung 4 dargestellt.



Abbildung 4: Evaluation eines Durchlaufs eines Q-Learning Agenten

[1] <https://openai.com/blog/introducing-openai/>

[2] <https://gym.openai.com/envs/CartPole-v0>

Bildquellen:

- Abbildung 1: <https://gym.openai.com/envs/CartPole-v0>
- Abbildung 2: Springer Fachmedien Wiesbaden 2013, Grundkurs Künstliche Intelligenz, Wolfgang Ertel, ISBN 978-3-8348-1677-1 S.300
- Abbildung 3: Eigene Darstellung
- Abbildung 4: https://gym.openai.com/evaluations/eval_SO0QB1FETfiC32gIkuJGA

Konzipierung und Realisierung eines Test-Access-Points für die PC-basierte Diagnose industrieller Kommunikationssysteme auf Basis eines FPGAs und Embedded-Linux

Ulf Schmelzer*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Feldbusse sind das Nervensystem moderner Industrieanlagen. Sie dienen zum Austausch von Daten zwischen Steuerungen, Sensoren und Aktoren sowie weiteren Feldbus-Komponenten. Der Datenaustausch zwischen den Bus-Teilnehmern erfolgt über umfangreiche und für Echtzeitanwendungen optimierte Kommunikationsprotokolle. Um bei Problemen eine detaillierte Diagnose von Feldbussen und deren Teilnehmern zu ermöglichen, ist es wichtig, die übertragenen Daten der Kommunikationsprotokolle unverfälscht und in Echtzeit aufzeichnen zu können. Diese Aufzeichnungen dienen als Basis für weitere Analysevorgänge und zur Eingrenzung des Problems auf bestimmte Feldbus-Teilnehmer. Aufgrund der großen Datenmengen sowie der Komplexität der Protokolle wird die weitere Analyse mit PC-basierten Diagnoseprogrammen durchgeführt. Ein Problem hierbei ist, dass PCs nicht über die Feldbus-spezifischen physikalischen Schnittstellen verfügen. Des Weiteren erfüllen die auf den PCs eingesetzten Betriebssysteme meistens nicht die Echtzeitanforderungen, um die eingehenden Feldbus-Telegramme in ihrem korrekten zeitlichen Zusammenhang zu setzen.

Um diesen Problemen zu begegnen werden für die Feldbus-Diagnose so genannte Test Access Points (TAPs) eingesetzt. Diese verfügen über die notwendigen physikalischen Schnittstellen und Echtzeitfähigkeiten, um Feldbus-Telegramme zu empfangen. Über eine gängige Kommunikationsschnittstelle, wie z.B. Ethernet, werden die Telegramme anschließend an einen PC weitergeleitet. Dort können diese mit entsprechenden Diagnosetools, wie z.B. Wireshark, grafisch dargestellt und analysiert werden. Indem beim Empfangen jedes Telegramm mit einem Zeitstempel durch den TAP versehen wird, können am PC die exakten zeitlichen Zusammenhänge ermittelt werden. In Abbildung 1 ist ein Anwendungsfall für einen solchen TAP dargestellt.

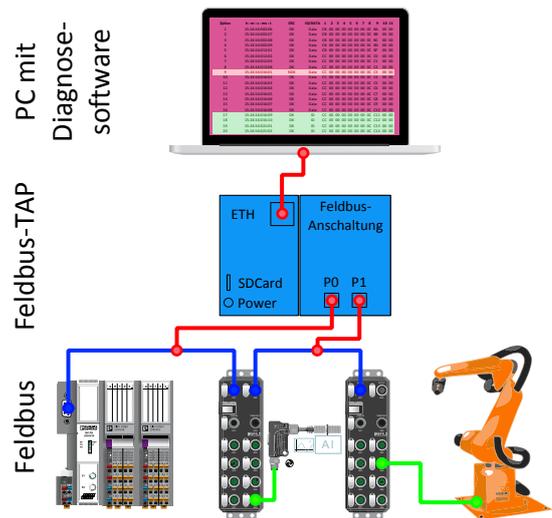


Abbildung 1: Schematische Darstellung eines TAPs zum Abhören eines Feldbus

Um eine umfangreichere PC-basierte Diagnose von industriellen Kommunikationsprotokollen zu ermöglichen, entwickelt die Steinbeis Embedded Systems Technologies GmbH gegenwärtig einen TAP. Dieser kann für verschiedenartige Feldbusse mit unterschiedlichen physikalischen Schnittstellen eingesetzt werden. Im Rahmen dieses Entwicklungsprojekts wurde die vorliegende Abschlussarbeit mit folgenden Zielen durchgeführt:

1. Konzeption und Implementierung einer Softwarearchitektur für die Feldbus-übergreifende Aufzeichnung und Weiterleitung von Feldbus-Telegrammen
2. Realisierung eines Moduls für die Aufzeichnung von INTERBUS-Telegrammen

Die Softwarearchitektur

Der zu entwickelnde TAP soll die Aufzeichnung von Telegrammen unterschiedlicher Feldbusse unterstützen. Daher musste bei der Konzeption der Softwarearchitektur darauf geachtet werden, dass die gesamte

*Diese Arbeit wurde durchgeführt bei der Firma Phoenix Contact Electronics GmbH, Bad Pyrmont, in Kooperation mit Steinbeis Embedded Systems Technologies GmbH, Esslingen

Feldbus-spezifische Logik für die Telegramm-Erkennung in eigenen Modulen gekapselt wurde. Für die zur Aufzeichnung relevanten Teile der Software mussten folgende Anforderungen realisiert werden:

- Verknüpfen der eingehenden Feldbus-Telegramme mit Zeitstempeln
- Zusammenfassen der Telegramme in Summentelegramme zur Reduzierung des zu übertragenden Datenvolumens
- Weiterleiten der Summentelegramme über Ethernet an einen PC

Zusätzlich war die Softwarearchitektur für folgende Erweiterungen auszulegen:

- Einsatz von Filtern für Feldbus-Telegramme zur Reduzierung des Datenvolumens
- Einsatz von Triggern zur Auslösung bestimmter Ereignisse, wie z.B. dem Zurücksetzen der Systemzeit des TAPs

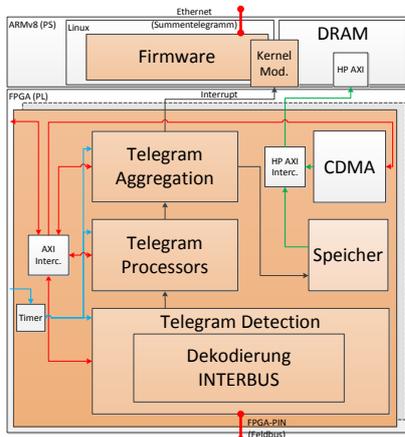


Abbildung 2: Softwarearchitektur des TAP

In einer früheren Arbeit [1] hat sich gezeigt, dass für diese Anforderungen die *Performance* eines Mikrocontrollers nicht ausreichend ist. Deshalb wird für die Realisierung des TAPs ein FPGA mit integriertem leistungsstarkem Prozessor und Embedded-Linux eingesetzt. Im Rahmen der Konzeption der Softwarearchitektur wurde zu Beginn festgelegt, welche Funktionen aufgrund von deren Echtzeitanforderungen und notwendiger Parallelisierung in Form von VHDL-Code in FPGA-Logik realisiert werden müssen. Die weniger Echtzeitkritischen und komplexeren Funktionen wurden der Ausführung im Embedded-Linux System zugeordnet. Zum Verarbeiten eingehender Feldbus-Telegramme für die FPGA-seitige Erkennung wurden die Funktionen in

weitere Module unterteilt und dann umgesetzt. Die dazu notwendigen Module sind in Abbildung 2 dargestellt. Das Bus-Signal wird eingelesen und gemäß dem Feldbus-Protokoll im *Telegram Detection Module* interpretiert. Anschließend werden die Bus-Telegramme im *Telegram Processor Module* mit einem Zeitstempel verknüpft und ggf. Filter und Trigger Ereignisse angewendet. Im *Telegram Aggregation Module* werden die Bus-Telegramme zu einem Summentelegramm zusammengefasst und durch den CDMA in den Speicher des Linux-Systems kopiert. Das *Linux Kernel Module* stellt eine einfache Schnittstelle zum Lesen der Summentelegramme für die *Firmware* bereit. Von dort werden die Summentelegramme über Ethernet an einen PC gesendet.

INTERBUS-Modul

Für die Umsetzung der INTERBUS-Feldbus-Signale wurde ein *Telegram Detection Module* realisiert, das einen INTERBUS-IP-Core von Phoenix Contact verwendet. Der IP-Core wurde so angepasst, dass die dekodierten INTERBUS-Telegramme kompatibel mit der Architektur sind. In Abbildung 3 ist der TAP mit einer INTERBUS-Karte als physikalische Schnittstelle dargestellt. Diese wird genutzt, um mit dem TAP INTERBUS-Signale in das Telegram Detection Module einzuspeisen.

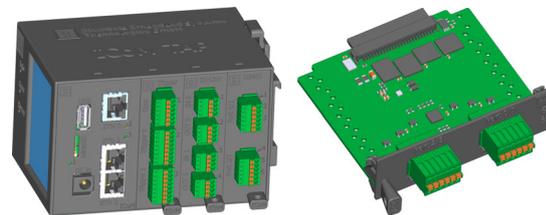


Abbildung 3: Prototypischer TAP und INTERBUS-Karte

Fazit

Im Rahmen dieser Bachelorarbeit wurde die Software für einen Feldbus-TAP entwickelt, der INTERBUS-Telegramme über Ethernet für Diagnosezwecke an einen PC weiterleitet. Die Software des TAPs wurde dabei so ausgelegt, dass sie mit sehr geringem Entwicklungsaufwand für das Aufzeichnen anderer Feldbusse eingesetzt werden kann.

[1] Fabig, Clemens (2016) – Konzeption und prototypische Umsetzung eines Mikrocontroller-basierten Converters zur Umsetzung der Protokolle verschiedener Feldbussysteme auf Ethernet für eine PC-gestützte Analyse

Bildquellen:

- Abbildung 1–2: Eigene Abbildung – enthält Grafiken von Phoenix Contact Electronics GmbH und Cliparts von <https://openclipart.org>, mit Inkscape erstellt
- Abbildung 3: Steinbeis Embedded Systems Technologies GmbH

Gamification zur Verbesserung der Prozess- und Datenqualität in Unternehmen

Benedikt Julian Schrade*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Fast jeder ist schon einmal mit einem Computerspiel in Berührung gekommen. Laut dem deutschen Branchenverband Bitkom spielen 42% der Deutschen und das durch alle Altersschichten [1]. Computerspiele sind in der Gesellschaft angekommen. Obwohl sie zum Teil sehr komplex sind, üben sie trotzdem einen hohen Reiz auf die Spieler aus. Seit einigen Jahren wird deshalb versucht, Prozesse in Unternehmen mehr wie Spiele zu gestalten, um den Nutzer so zu motivieren, diese unter anderem korrekter und schneller auszuführen. Diese Idee heißt Gamification.

In großen Unternehmen wird aufgrund der Vielzahl von Abläufen versucht, diese in Prozessen zu standardisieren und zu vereinheitlichen. Das bietet den Unternehmen viele Vorteile, wie z.B. einfachere Schulungen, wird jedoch von vielen Mitarbeitern als störend empfunden. Sie müssen sich an einen vorgegebenen Ablauf halten, der unter Umständen mehr Zeit kostet, aber keinen direkt spürbaren Nutzen für den Mitarbeiter bringt.

Name	Punkte	Status
101	02.200	
102	02.200	
103	02.200	
104	02.275	Spätkommer
105	02.200	
106	02.200	
107	02.200	

Abbildung 1: Mockup einer Rangliste

Zielsetzung der vorliegenden Arbeit ist eine Analyse der Möglichkeiten Gamification-Ansätze in Unternehmensprozesse zu integrieren. Gamification soll dabei die Prozess- und Datenqualität einer Unternehmensanwendung zu erhöhen. Hohe Datenqualität bedeutet unter anderem, dass die Datensätze vollständig, aktuell und redundanzfrei sind. Von einer hohen Prozessqualität spricht man, wenn Prozesse vollständig und korrekt ausgeführt wer-

den, sie nur einmal ausgeführt werden müssen und der gesamte Prozessablauf sauber dokumentiert wird.

Bei der für diese Arbeit betrachteten Anwendung handelt es sich um eine Verkaufsanwendung. In ihr werden alle Prozesse des Verkäufers während des Verkaufs dokumentiert. In der Anwendung wurden nach eigenen Recherchen und mehreren Experteninterviews folgende Probleme festgestellt: Viele Nutzer führen manche Prozesse gar nicht oder nur unvollständig aus. Da in vielen Prozessen Daten eingegeben werden müssen, sind viele Datensätze der Anwendung unvollständig oder nicht aktuell. Um einige Prozesse abzukürzen, werden außerdem teilweise Datensätze neu angelegt, obwohl sie bereits im System existieren. Außerdem sind den meisten Nutzern viele Funktionen der Anwendung unbekannt. Zu guter Letzt werden auch Bugs nicht immer gemeldet. Häufig wird lieber ein Workaround gesucht, um den Bug zu umgehen, anstatt ihn zu melden [2].

Lösungsansätze

Es gibt eine Vielzahl von unterschiedlichen Ansätzen, den Nutzer mit Hilfe von Gamification zu motivieren. Zu den häufigsten gehören: Einführung eines Punktesystems, bei dem der Nutzer für seine Tätigkeit mit virtuellen Punkten belohnt wird. Das Führen einer Rangliste, um den Wettbewerb zwischen den Nutzern zu fördern und sie so zu motivieren. Oder die Einführung von Quests, sodass der Nutzer immer wieder kleine Aufgaben bekommt, die er in der Anwendung zu erfüllen hat.

Am Anfang der Arbeit wurde eine Nutzeranalyse durchgeführt: Die Nutzer der Anwendung sind meist männlich und sehr wettbewerbsorientiert. Sie vergleichen ihre Verkaufszahlen bereits in regelmäßigen Abständen, um zu ermitteln, wer der beste Verkäufer ist. Auf dieses Verhalten sollte auch bei der Entwicklung eines Ansatzes zur Verbesserung der Prozess- und Datenqualität eingegangen wer-

* Diese Arbeit wurde durchgeführt bei der Firma Caggemini Deutschland GmbH, Stuttgart-Degerloch

den. Anhand der Probleme und der Nutzeranalyse wurde ein Ansatz erstellt, der die Probleme der Anwendung lösen und Prozess und Datenqualität der Anwendung verbessern soll.

Alle Probleme lassen sich eigentlich auf ein Grundproblem zurückführen: Den Nutzern fehlt ein Anreiz, sich intensiver mit der Anwendung zu beschäftigen und die Prozesse korrekt auszuführen. Das liegt vor allem daran, dass die Nutzer nur am Verkauf ihrer Produkte interessiert sind, für die sie eine Prämie erhalten. Um diesem Anreiz entgegen zu wirken, wurde der starke Wettkampfgedanke aufgegriffen, um den Nutzern in der Anwendung die Möglichkeit zu geben, sich mit anderen Nutzern zu vergleichen. Das bringt ihnen zwar kein Geld ein, es befriedigt jedoch das Bedürfnis, besser als die anderen Nutzer zu sein. Zu diesem Zweck soll ein Punktesystem eingeführt werden, das alle Aktionen eines Nutzers mit Punkten belohnt. Je mehr Daten der Nutzer in das System eingibt, desto mehr Punkte bekommt er als Belohnung. Diese Punkte kann der Nutzer dann in einer Rangliste mit der Punktzahl anderer Nutzer vergleichen. So wird der Wettbewerb zwischen den Nutzern gefördert und das Eintragen der Daten nicht mehr als lästige Pflichtaufgabe empfunden, sondern von den Nutzern als sinnvoll erachtet.

 The image shows a detailed mockup of a web application interface. It features a top navigation bar with a logo and several menu items. Below this, there are several panels and form elements:

- A left sidebar with a search bar and a list of items.
- A main content area with a header and a large form containing multiple input fields, checkboxes, and buttons.
- A right sidebar with a list of items and a small chart or progress indicator.
- At the bottom, there are several small pop-up windows or notifications.

 The overall design is clean and modern, with a focus on user interaction and data entry.

Abbildung 2: Mockup einer Eingabemaske mit Gamification-Elementen inklusive Pop-up-Fenstern

Dabei gilt es jedoch einige Einschränkungen im Hinblick auf das deutsche Datenschutzrecht zu beachten. So darf man in Deutschland nicht einfach die Daten von Nutzern sammeln und

diese dann nutzen, um öffentlich die Leistungen der Nutzer zu vergleichen. In Deutschland muss hier eine Anonymisierung stattfinden, was dem Wettbewerb natürlich ein wenig die Spannung nimmt, da man nicht mehr genau weiß, mit wem man sich eigentlich misst. Eine Lösung hierfür wäre nicht der Vergleich von einzelnen Personen, sondern von ganzen Büros oder Standorten. Hierbei muss der Vergleich aber so errechnet werden, dass das Ergebnis des einzelnen Nutzers nicht mehr heraus gerechnet werden kann.

Das Punktesystem ist Ausgangspunkt für weitere „gamifizierte“ Elemente in der Anwendung. So soll noch ein Quest-System eingeführt werden, das den Nutzer motivieren soll, sich mehr mit der Anwendung zu beschäftigen und auch die neuen, ihm unbekanntem Funktionen kennen zu lernen. Außerdem ermöglicht das Quest-System dem Management eines Standorts gezielt bestimmte Produkte zu verkaufen, indem es die Verkäufer für den Verkauf eben dieser Produkte belohnt.

Fazit

Gamification ist mehr als das Einführen von Punkten und Badgets (virtuelle Orden) in eine Anwendung. Dies motiviert die Nutzer vielleicht im ersten Moment, doch auf lange Sicht macht das die Anwendung nicht besser. Man muss sich der ganzen Anwendung eher aus der Sicht eines Spieleentwicklers nähern. Zuerst sollte man sich überlegen, wo eigentlich die Probleme der Anwendung liegen und was die Ursachen für diese Probleme sind. Dann sollte man sich die Nutzer genau ansehen und ausgehend davon, nach gamifizierten Anwendungen suchen, die die Nutzer auch wirklich ansprechen. Erst dann kann man sich Gedanken darüber machen, wie die Gamification wirklich realisiert wird. Gamification ist nichts, was man schnell mal in seine Anwendung einfügen kann, sondern es bedarf etwas Vorarbeit, um sie sinnvoll zu integrieren. Das erfordert natürlich mehr Aufwand, doch nur dann kann man sich sicher sein, dass sich das Verhalten der Nutzer auch wirklich langfristig ändert.

[1] Bitkom (29.07.2015): Gaming hat sich in allen Altersgruppen etabliert. Online verfügbar unter <https://www.bitkom.org/Presse/Presseinformation/Gaming-hat-sich-in-allen-Altersgruppen-etabliert.html>, zuletzt geprüft am 19.09.2016.

[2] Experteninterview

Bildquellen:

- Abbildung 1-2: Eigene Abbildung

Informationsextraktion durch Methoden des maschinellen Lernens und Anwendung auf einen ETL-Prozess für Fahrzeugdaten

Christoph Stähler*, Dirk Hesse, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

ETL-Prozesse stellen eine einfache Modellierungsmethode dar um Daten aus unterschiedlich strukturierten Datenquellen in einer Zieldatenbank zu vereinigen.

In der Extraktionsphase werden Daten aus verschiedenen Quellen selektiert. Werden Daten aus unterschiedlich semistrukturierten Quellen bezogen, muss der Teil der Anwendung zur Extraktion der Daten verschiedene Datenstrukturen verarbeiten können. Für jede dieser Struktur benötigt die Anwendung die Information, wo die zu selektierenden Daten in der Datenstruktur abgelegt und zu selektieren sind. Werden die Datenstrukturen regelmäßig geändert, ist dieser Prozess mit einer gewissen Fehlerhäufigkeit und einem ständigen Wartungsaufwand verbunden.

Im Rahmen dieser Bachelorarbeit soll untersucht werden, ob eine Extraktion aus semistrukturierten Daten mit Methoden des maschinellen Lernens vereinfacht werden kann. Als Anwendungsfall dient ein Teil eines ETL-Prozesses der Daimler AG, welcher Kerncharakteristiken aus semistrukturierten Fahrzeugdaten selektiert.

Maschinelles Lernen

Mithilfe maschinellen Lernens wird versucht menschliche Entscheidungen nachzubilden ohne jede Situation und der daraus resultierenden Entscheidung programmatisch vorgeben zu müssen. Das Ziel ist es, Programme zu entwickeln welche eigenständig Probleme lösen können.

Maschinelles Lernen wird grob in drei verschiedene Lernverfahren kategorisiert [1]:

- **Überwachtes Lernen:** das Programm lernt Entscheidungen auf Basis von Trainingsdaten, welche zu einer Menge an Fragen die dazugehörigen korrekten Antworten enthält.
- **Unüberwachtes Lernen:** das Programm erkennt selbständig Muster in vorliegenden Daten.
- **Bestärkendes Lernen:** das Programm spielt gegen sich selbst und erlernt so anhand des Ergebnisses optimale Strategien.

In dieser Arbeit werden Verfahren des überwachten Lernens betrachtet, da Trainingsdaten vorhanden waren und so eine höhere Genauigkeit erreicht werden konnte. Im Folgenden werden mehrere dieser aufgezeigt.

Entscheidungsbäume

Bei Entscheidungsbäumen handelt es sich um ein sogenanntes Klassifikationsverfahren. Klassifikationsverfahren ordnen Objekte einer bestimmten Klasse zu.

Das Entscheidungsbaumverfahren bildet Entscheidungen über eine Baumstruktur ab. Von einem Wurzelknoten aus werden die Charakteristiken der zu klassifizierenden Objekte jeweils als Knoten abgelegt. Jeder ausgehende Ast steht dabei für eine Ausprägung der Charakteristik. Die Blätter bezeichnen die Klasse der Objekte auf welche die Entscheidungen des Pfades zutreffen. Basierend auf den Ausprägungen der Charakteristiken des zu klassifizierenden Objekts wird an jedem Knoten für einen weiteren Ast entschieden, worüber ein weiterer Knoten oder ein Blatt erreicht wird. Durch ein schrittweises Durchlaufen des Baumes wird das Objekt so klassifiziert.

Die Baumstruktur der Entscheidungsbäume wird basierend auf den Trainingsdaten erstellt. Somit gehören Entscheidungsbäume zu den überwachten Lernverfahren.

Naïve Bayes

Naïve Bayes gehören wie Entscheidungsbäume zu den Klassifikationsverfahren. Naïve Bayes geben die Wahrscheinlichkeit an, mit welcher ein Objekt zu einer Klasse gehört. Die Methode basiert auf dem Theorem von Bayes, welche dem Verfahren seinen Namen gibt. Mit der Formel von Bayes wird die bedingte Wahrscheinlichkeit eines Ereignisses berechnet.

Anhand der Trainingsdaten, welche zu einer Menge an Charakteristiken von Objekten die zugehörige Klasse enthalten, wird für jede Charakteristik die bedingte Wahrscheinlichkeit für eine Klasse berechnet. Also die Wahrscheinlichkeit für eine Ausprägung einer Charakteristik, wenn ein Objekt zu einer bestimmten Klasse gehört. Mithilfe des Theorems von Bayes wird dann für die Menge an Charakteris-

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

tiken des neuen zu klassifizierenden Objekts die Wahrscheinlichkeit für die Klasse des Objekts berechnet.

Künstliche neuronale Netze

Künstliche neuronale Netze sind der Funktionsweise des menschlichen Gehirns nachempfunden. Nervenzellen des Gehirns werden über sogenannte künstliche Nervenzellen nachgebildet. Diese lassen je nach Eingabe eine Verarbeitung der Daten zu. Eine Kombination aller künstlichen Nervenzellen ermöglicht eine Verarbeitung der Daten. Durch verschiedene Strukturen dieser Netze, wie einer mehrschichtigen Verarbeitung durch sogenannte versteckte Ebenen, können komplexere Handlungsweisen des künstlichen neuronalen Netzes erreicht und die Lernfähigkeit verbessert werden.

Vorgehen

Um zu entscheiden, welches Lernverfahren für die Informationsextraktion geeignet ist, wurden mehrere Verfahren am Beispiel von Fahrzeugdaten miteinander verglichen. Die Verfahren mussten dazu erkennen ob eine Zeichenkette aus einem Text eine gesuchte Kerncharakteristik ist.

Eine prototypische Umsetzung erfolgte auf Basis von Apache Spark. Apache Spark ist ein Framework für Datenanalysen in einem Rechnerverbund. Durch die Bereitstellung von Basisklassen für jedes der zu betrachtenden Lernverfahren ermöglichte es eine schnelle Prototypentwicklung.

Die Lernverfahren Entscheidungsbaum, Naïve Bayes und künstliche neuronale Netze wurden in unterschiedlichen Ausprägungen, wie beispielsweise unterschiedlicher Baumtiefe bei Entscheidungsbaumverfahren oder unterschiedlicher Anzahl an versteckten Ebenen in einem künstlichen neuronalen Netz, getestet. Abbildung 1 stellt die Ergebnisse dieser Evaluation dar, wobei nur jeweils die Ausprägungen der Verfahren mit der geringsten Fehlerquote gezeigt werden. Als Fehler gilt eine Zeichenkette, welche einer andern Klasse zugeordnet wurde, als der Klasse der die Zeichenkette eigentlich angehört.

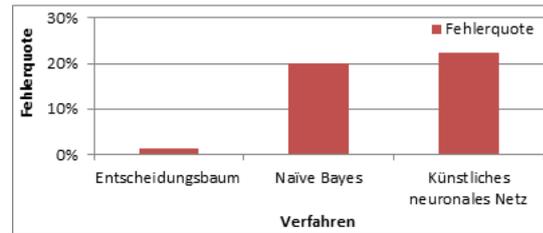


Abbildung 1: Vergleich der Verfahren nach Fehlerquote

Das Entscheidungsbaumverfahren wurde aufgrund der geringen Fehlerquote für die Umsetzung ausgewählt. Die finale Implementierung erfolgte ebenfalls auf Basis von Apache Spark, da es im Vergleich zu anderen Frameworks und Bibliotheken für Verfahren des maschinellen Lernens mehrere Vorteile hat: Neben einer einfachen Integration von Sparkanwendungen in bestehende Hadoopumgebungen bietet Apache Spark Performancevorteile gegenüber anderen Implementierungen [2,3]. Zum anderen deutet sich durch eine starke Entwicklergemeinschaft [4] und kommerziellen Vermarktung [5,6] eine stabile Weiterentwicklung und Support des Frameworks an.

Die Ergebnisse einer Klassifikation werden in einer Hive-Datenbank gespeichert. Von dieser Datenbank aus können die Daten beispielsweise innerhalb eines ETL-Prozesses weiterverarbeitet werden.

Bei der Umsetzung wurde besonders auf die Modularisierung der Anwendung geachtet, um einzelne Komponenten auch in anderen Anwendungsfällen verwenden zu können.

Zusammenfassung

Ziel der Arbeit war es Kerncharakteristiken aus Fahrzeugdaten mithilfe von Methoden des maschinellen Lernens zu extrahieren.

Dazu wurden mehrere Klassifikationsverfahren einander gegenübergestellt und verglichen. Durch ein Training mit 50.000 Datensätzen konnte eine Erkennungsrate von über 95% bei Testdaten mit dem ausgewählten Entscheidungsbaumverfahren erreicht werden.

Die Anwendung wurde auf Basis von Apache Spark realisiert.

[1] Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006.

[2] Xiangrui Meng, MLlib: Machine Learning in Apache Spark, 2016, URL: <http://www.jmlr.org/papers/volume17/15-237/15-237.pdf> (besucht am 29.11.2016)

[3] Tom White, Hadoop: The Definitive Guide - 4th Edition, O'Reilly Media, 2015.

[4] Spark Commit Aktivität, URL: <https://github.com/apache/spark/graphs/commit-activity> (besucht am 29.11.2016)

[5] Derrick Harris, Databricks raises \$14M from Andreessen Horowitz, wants to take on MapReduce with Spark, 2013, URL: <https://gigaom.com/2013/09/25/databricks-raises-14m-from-andreessen-horowitzwants-ato-take-on-mapreduce-with-spark/> (besucht am 30. 11. 2016).

[6] Ron Miller, Databricks Snags \$33M In Series B And Debuts Cloud Platform For Processing Big Data, 2014, URL: <https://techcrunch.com/2014/06/30/databricks-snags-33m-in-series-b-and-debuts-cloud-platform-for-processing-big-data/> (besucht am 30. 11. 2016).

Bildquellen:

- Abbildung 1: Eigene Abbildung

IT Prozess Stabilisierung & Optimierung für das MFTBC SAP System im Finanz & Controlling Umfeld

Waldemar Stenski*, Kai Warendorf, Catharina Kriegbaum-Kling

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Diese Bachelorarbeit bezieht sich auf die Verbesserung und Stabilisierung von IT Prozessen, daher wurden verschiedene Analysen für das Mitsubishi Fuso Truck & Bus Corporation (MFTBC) SAP System vorgenommen. Diese basieren auf den Strategien von Business Intelligence und Business Analytics. Während die versendete Datenmenge und das Internet der Dinge stets wachsen, sind Fahrzeuge heutzutage immer mehr mit der Außenwelt verbunden und senden wichtige Informationen an den Fahrer um beispielsweise einen Stau zu umgehen. Um die IT Landschaft von MFTBC an den aktuellen Stand der Technik anzupassen [1], werden erforderliche Business und IT Prozesse optimiert und stabilisiert.

Big Data & Business Intelligence

Big Data beschreibt eine riesige Menge an digitalen Daten in unserer Welt. Diese Daten sind für die meisten Personen unzugänglich und werden sehr oft von Unternehmen genutzt um Erfahrungen über Kundenvorhaben zu sammeln um davon zu profitieren. Heutzutage werden diese Daten durch verschiedene Quellen und in verschiedenen Mustern generiert. Diese wären zum Beispiel: Mobile Daten Nutzung, Soziale Netzwerke, Cloud-Computing, Streaming von Medien, GPS Tracking und viele andere.



Abbildung 1: Big Data [2]

Business Intelligence & Big Data geben einen Ausblick über einen höheren Standard der Projekte. Dabei werden Daten von Kunden, Dienstleistern und Fahrzeugen gesammelt und visuell dargestellt, damit Manager Entscheidungen einfacher und effizienter treffen

können. Zusammenfassend geht es darum, relevante und zuverlässige Informationen zum richtigen Zeitpunkt an die richtigen Personen zu bringen, um bessere Entscheidungen schneller treffen zu können. BI sammelt und strukturiert Daten mit Methoden und Programmen, um sie in Informationen zu konvertieren und so Unternehmensentscheidungen zu verbessern. [2]

Prozess Optimierung

Zehn Design Heuristiken

Heutzutage sind die Begriffe User Experience (UX) und Usability in der ganzen Welt vertreten. Dabei wird eine Software in Hinsicht auf benutzerfreundlich erstellt und dient zur einfachen Interaktion mit dieser. Auch im Jahr 1990 hatte sich Jacob Nielsen mit der Usability von Software Programmen auseinandergesetzt und einen Standard eingebracht der heute, über 25 Jahre später, noch in der Welt von UX Verwendung findet. Insgesamt gibt es zehn Heuristiken, die in drei Schichten aufgeteilt werden. Diese können am Anfang eines Projekts zum Beispiel in einem Papierprototyp oder auch während der Testphase verwendet werden, wenn Softwareentwickler durch neue Geschäftsanforderungen die Software anpassen müssen [3].

Nutzer Wahrnehmung

- Einheitliche Dialoge
- Zeige das Erwartete
- Denken wie der Nutzer

Interaktionen Anzeigen

- Am laufenden halten
- Fehler vermeiden
- Erholung nach Fehlern
- Helfende Hand & Spaß

Verfahrensweise

- Spielraum erlauben
- Vielseitigkeit
- Standardvorgaben

*Diese Arbeit wurde durchgeführt bei der Firma Mitsubishi FUSO Truck & Bus Corporation, Japan, Kawasaki

Prozess Stabilisierung Software Test

In diesem Abschnitt wird erklärt, warum ein Testplan erforderlich ist und wie dieser zu erstellen ist. Außerdem werden zwei Teststrategien, die Black-Box & White-Box-Tests, eingeführt und danach neun verschiedene Testprinzipien erläutert an denen sich ein Softwaretester orientieren kann. Nach einem Rollout neuer Software ist der nächste Schritt eines Unternehmens, funktionale Tests zu starten um Bugs und Fehler ausfindig zu machen. Jedoch haben die meisten Entwickler und Tester eine falsche Definition von Tests. Dabei wird in den meisten Fällen angenommen: „ Tests werden durchgeführt um zu zeigen das alle Funktionen eines Programms korrekt ausführbar sind“. Diese Definition ist falsch, eher ist davon auszugehen, dass eine Software Fehler hat und der Tester diese finden muss. Glenford J. Myers definiert in seinem Buch „ The Art of Software Testing“ Testen wie folgt: “Testen ist der Prozess der Ausführung eines Programms mit der Absicht, Fehler zu finden.“ [4]



Abbildung 2: Business Intelligence [4]

Incident Management

Schließlich ist es unvermeidlich durch Incident Management zu gehen, da Ausfälle überall entstehen können und es ist fast unmöglich ist diese zu vermeiden ohne hohe Verfügbarkeit Server bereitzustellen. Testen ist dabei ein kreativer und phantasievoller Prozess. Auch nach Stunden der Ausgaben für die Prüfung der Software können in der finalen Version immer noch Fehler zu finden sein. Daher hat das Incident Management seine Verwendung,

die den Prozess nach dem Testen oder einer Fehlerbehebung steuert.

Zusammenarbeit Verschiedener Abteilungen

Darüber hinaus können einzelne Abteilungen wie die IT-, Business-, Design- oder das Management nicht für sich selbst entscheiden, welche Prozesse eingeführt werden sollen. Die verschiedenen Teams müssen in Meetings vorerst Anforderungen definieren und sich miteinander abstimmen. Dabei hat jede Abteilung ihren eigenen Zustimmungsbereich. Dementsprechend ist die IT verantwortlich für die Entwicklung einer technischen Struktur, Einrichtung von Backup Servern oder für die Verwaltung von Rechenzentren. Die Business Abteilung erstellt neue Anforderungen und steuert Prozesse. Ein Designer hat verschiedensten Anforderungsspezifikationen gerecht zu werden. Es reicht vom Verstehen der Anforderungen eines Kunden bis zum Erstellen benutzerfreundlicher Software. Letztlich das Projektmanagement, das für das gesamte Projekt haftet und sein Team zum Erfolg führen soll. Dabei wird auf genaue Projektziele in Zeit, Geld, Qualität und Effizienz geachtet. Um diese Ziele zu erreichen, muss der Manager strenge Regeln beachten.

Schlussfolgerung

Nach einer Umstellung vom MFTBC Legacy System auf die neue SAP Standard Software, werden die Prozesse und damit die Arbeitserfahrung für Anwender verbessert, um Zeit und Geld zu sparen und die Handarbeit zu reduzieren. Mit den eingeführten Design-Heuristiken von Jakob Nielsen wird die Software in allen Hinsicht um die heutige Mensch Computer Interaktion Usability Erfahrung erweitert. Aber neue Releases bringen immer Probleme mit sich, daher ist ein breites Spektrum von Tests und Benutzer Trainings unter keinen Umständen zu vernachlässigen. Werden die Regeln für Incident Management befolgt und wird dabei auf eine reibungslose Koordination geachtet, wird das neue SAP System im Falle eines Softwarefehlers in der Lage sein sich von diesem zu erholen ohne große Schäden zu hinterlassen.

- [1] Stand der Technik, Dr. Gabriel Meckel, <http://wirtschaftslexikon.gabler.de/Definition/stand-der-technik.html>, URL zuletzt aufgerufen 2016/November/29
- [2] What is Business Intelligence (BI)?, <https://www.youtube.com/watch?v=hDJdkcdG1iA>, Hitachi Solutions Canada, URL zuletzt aufgerufen 2016/November/28
- [3] 10 Usability Heuristics for User Interface Design, <https://www.nngroup.com/articles/ten-usability-heuristics/>, Jakob Nielsen, 1995/January/01, URL zuletzt aufgerufen 2016/September/15
- [4] The Art of Software Testing, Glenford J. Myers, 2012 3rd Edition, page 6, ISBN 978-1-118-03196-4

Bildquellen:

- Abbildung 1: 4 Ways Big Data is Changing Inventory Management, Dan Dowling, 01 Sep. 2015, <http://www.eazystock.com/blog/2015/09/01/4-ways-big-data-is-changing-how-companies-manage-inventory/>
- Abbildung 2: Tableau against the world: The battle of Business Intelligence heats up, Juan F. Cía, 01 Apr. 2015, <https://bbvaopen4u.com/en/actualidad/tableau-against-world-battle-business-intelligence-heats>

Analyse und prototypische Implementierung von Schrittzähleralgorithmen mit einem MEMS Inertialsystem bestehend aus Beschleunigungs- und Drehratensensor

Georgios Terzoglou*, Kai Warendorf, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Activity-Tracker[1] erfahren eine immer größere Beliebtheit unter Sportlern und generell körperbewussten Menschen. Die Möglichkeit über den gesamten Tag hinweg eine Aufzeichnung und Analyse der eigenen Aktivitäten wie zum Beispiel Joggen, Gehen, Liegen, Sitzen, Schwimmen oder auch Radfahren kann eine Unterstützung sein um gesetzte Trainingsziele besser zu erreichen.

Neben der Aktivitätserkennung kann auch der Herzschlag erfasst, die Körpertemperatur gemessen, die zurückgelegte Strecke ermittelt und in Summe der Energieumsatz berechnet werden. Der Anwender kann sich auch durch ein akustisches Signal oder eine Vibrationsmeldung über zu geringe Aktivität informieren lassen.



Abbildung 1: Fitness-Tracker Jawbone up3

Seitens der Sensorik kommen meist Beschleunigungs-, Drehraten-, Temperatur-, Luftdruck-, Licht- oder auch GPS Sensoren zum Einsatz. Die aufgezeichneten Daten kön-

nen dann an einem Rechner oder Smartphone übertragen, ausgewertet und über Soziale Netzwerke geteilt werden.

Eine zentrale Funktionalität von Activity-Trackern ist das Zählen von Schritten beim Gehen oder Joggen. Dies wird maßgeblich durch die Erfassung von Beschleunigungs- und oder Drehratensensordaten mit entsprechender Algorithmen erreicht.

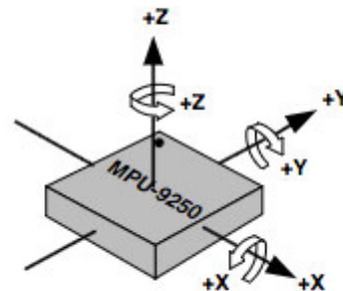


Abbildung 2: InvenSense Inc. MPU-9250

Im Rahmen dieser Bachelorthesis wurden Schrittzähleralgorithmen analysiert und auf einem Entwicklungsboard prototypisch implementiert.

Hierzu wurde der Sensor MPU-9250 von InvenSense Inc. und das Entwicklungsboard Nucleo 411re von STMicroelectronics verwendet. Der MPU-9250 besteht aus einem dreiaxigen Beschleunigungs-, Drehraten- und Magnetfeldsensor. Das Entwicklungsboard beherbergt einen ARM Cortex M4 Mikrocontroller (STM32F411RE).

In dieser Arbeit wurde ausschließlich der Beschleunigungs- und Drehratensensor herangezogen.

Das Grundlegende Prinzip und die damit einhergehende Herausforderung der Schritterkennung von Personen ist die Ermittlung und Zuordnung von Signalausschlägen, aus dem Signalrauschen des Inertialsystemes, wel-

*Diese Arbeit wurde durchgeführt bei der Firma BASICWORX ENGINEERING GmbH, Stuttgart

che durch die tatsächlichen Schritte entstanden sind. Einen hochpräzisen Ansatz bieten Künstliche Neuronale Netze welche allerdings sehr rechenlastig sind. In dieser Arbeit sollten die möglichst einfachsten Methoden eingesetzt werden um mögliche Portierungen auf Mikrocontrollern realisierbar zu halten.

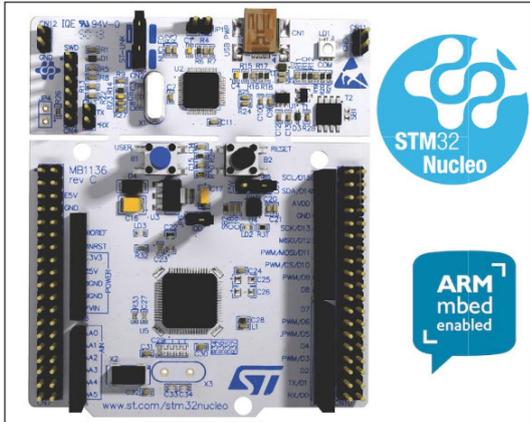


Abbildung 3: STMicroelectronics Nucleo F411re

Zu Beginn wurden die Rohdaten vom Beschleunigungs- und Drehratensensors in Echtzeit per Bluetooth Verbindung an einen Messrechner übertragen und in Matlab/Simulink ausgewertet. In diesem Umfeld wurden dann unterschiedliche Methoden erprobt und ein eigener Algorithmus entwickelt mit dem die Schritte sehr gut erkannt werden können. Im Anschluss erfolgte die Portierung des Algorithmus in C und die Integration auf das Entwicklungsboard, so dass die Schritterkennung autark genutzt werden kann.

[1] https://de.wikipedia.org/wiki/Activity_Tracker

Bildquellen:

- Abbildung 1: <https://jawbone.com/fitness-tracker/up3>
- Abbildung 2: <http://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf>
- Abbildung 3: http://www.st.com/content/ccc/resource/technical/document/data_brief/c8/3c/30/f7/d6/08/4a/26/DM00105918.pdf/files/DM00105918.pdf/jcr:content/translations/en.DM00105918.pdf

Entwicklung eines verteilten Datenbanksystems mit Webfrontend für die Verwaltung und Analyse von Diagnosedaten im Automobilbereich

Marcel Trinks*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Kombiinstrumente zeigen dem Fahrer alle fahrrelevanten Kennzahlen wie aktuelle Geschwindigkeit, Drehzahl oder aktive Fahrassistenzsysteme. Die Robert Bosch GmbH entwickelt und programmiert solche frei programmierbaren Kombiinstrumente (FPK) unter anderem für den Kunden BMW. Die Digitalisierung von Kombiinstrumenten erhält Einzug in die Fahrzeuge von heute. Dadurch müssen nicht mehrere Varianten davon produziert werden, sondern es gibt ein Kombiinstrument mit verschiedenen Codierungen. Damit werden länderspezifische Regelungen und Standards sowie technische Unterschiede einzelner Fahrzeuge eingehalten, wie z.B. die verschiedene Anzeige der Drehzahl von Benzin- und Dieselfahrzeugen.

Im Angesicht der Qualitätssicherung müssen die FPK abgesichert werden. Das geschieht an Prüfplätzen im Labor oder in Erprobungsfahrzeugen. Bei den Prüfplätzen ist das Kombiinstrument an einem Testrechner angeschlossen und bekommt Signale über eine spezielle Software namens Vector CANoe. Dabei wird der Fahrzeugbus in einer Restbussimulation dargestellt. Dies bedeutet, dass viele anderen Steuergeräte im Fahrzeug simuliert werden, um ein komplettes System und deren Zusammenspiel abzubilden [1]. Diese Tests sind durch die CAPL Programmierung hochgradig automatisiert. Im Erprobungsfahrzeug werden Tests in einer realen Umgebung durchgeführt. Die Daten werden geloggt und können mit einem Computer ausgelesen werden. Dazu gehören auch Screenshots oder Videos des Kombiinstrumentes. Diese werden über eine GoPro-Kamera aufgenommen. Wichtig sind neben den Signalen auch die Software- und Hardwareversionen der verbauten Steuergeräte, um einen kompletten Aufbau des Systems nachvollziehen zu können. In Abbildung 1 wird der Prozess einer Erprobung inklusive der angestrebten Weiterverarbeitung der Daten dargestellt.

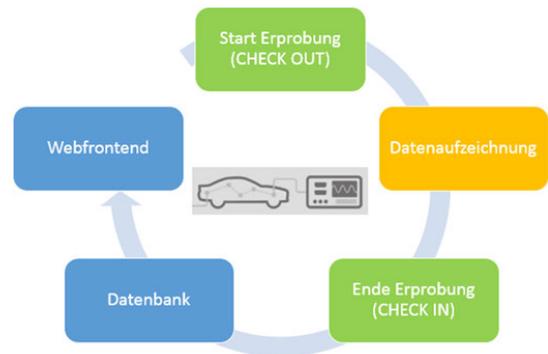


Abbildung 1: Prozess Erprobung

Im Rahmen dieser Bachelorarbeit soll ein System entwickelt werden, welches Daten der Kombiinstrumente in einer Datenbank speichert. Diese Datenbank soll große Mengen an Daten speichern und verarbeiten können. Bei einer Erprobungsfahrt fallen ca. 5GB Daten/Stunde an. Zu beachten ist ein Sonderfall im Erprobungsfahrzeug, da der Rechner dort nicht permanent mit dem Bosch-Netzwerk verbunden ist. Hier muss es eine Möglichkeit geben, die Daten zwischen zu speichern. Zudem sollen alle Daten wieder über eine Anwendung sichtbar gemacht werden. Die Anwendung muss die Daten logisch anzeigen und soll mit einer Suche ausgestattet werden.

Zuerst muss die Auswahl der Datenbank erfolgen. Dabei gibt es Vorgaben wie eine große Speicherkapazität, Multiuserzugriff und einer möglichst kostengünstigen Lösung. Aufgrund dieser Parameter wurde nach gründlicher Abwägung MySQL als Datenbanksystem gewählt. Die Open Source Lösung hat keinerlei Einschränkungen in der Speicherkapazität. Diese wird nur durch das Betriebssystem und die Speicherkapazität der Festplatte des Servers beschränkt. Dadurch hat MySQL einen entscheidenden Vorteil gegenüber anderen kostenfreien Systemen. Zudem unterstützt sie den Multiuserzugriff und bietet Programmierschnittstellen für mehrere Programmiersprachen. Wichtig ist hierbei die Unterstützung von PHP, welche bei der Entwicklung der Webanwendung zum Einsatz kommt.

Durch das spätere Einsatzgebiet der An-

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

wendung muss zwischen einer Web- oder Desktopanwendung entschieden werden. Da die Anwendung parallel nutzbar sein soll und eine spätere Wart- und Erweiterbarkeit durch eine Webanwendung einfacher wird, wurde dieser Ansatz gewählt. Unter dem Gesichtspunkt der Offline-Verfügbarkeit entsteht dabei eine verteilte Architektur, die mit Microservices arbeitet. Microservices gehören dabei zu den verteilten Systemen. Sie arbeiten nach dem Prinzip, dass ein Service genau eine Aufgabe erledigt. Vorteile bieten diese durch den einfachen Austausch einzelner Komponenten und der Unabhängigkeit der verwendeten Technologie pro Service [2]. Abbildung 2 zeigt das Zusammenspiel der Microservices im entwickelten System.

Die Architektur arbeitet folgendermaßen: Durch einen Prüflauf werden Daten gesammelt, die als Dateien in einen definierten Ordner der Testumgebung gelegt werden. Der Service „FileWatcherAndCopier“ überwacht diesen Ordner. Sobald Dateien in den Ordner geschrieben werden, kopiert sich der Service diese von der Testumgebung in ein temporäres, lokales Verzeichnis auf dem Testrechner. Somit besteht keine Abhängigkeit mehr zur Testumgebung. Wenn der Testrechner eine Verbindung zum Bosch-Netzwerk hat, sendet der Service eine Nachricht an den Message-Queue-Server. Hierbei wird der RabbitMQ Server aufgrund seiner Unabhängigkeit der Programmierschnittstellen gewählt. Dabei kann auf mehrere Programmiersprachen zur Implementierung der Clients zurückgegriffen werden, um die Flexibilität im Rahmen der Microservice-Architektur aufrecht zu erhalten. Ein zweiter Service „FileUploader“ überwacht die Message-Queue auf eine Nachricht, dass Dateien lokal verfügbar sind. Danach sendet er diese über einen http-POST Request an eine PHP-Schnittstelle auf der Serverseite. Diese Schnittstelle nimmt die Dateien entgegen und schreibt diese in die Datenbank.

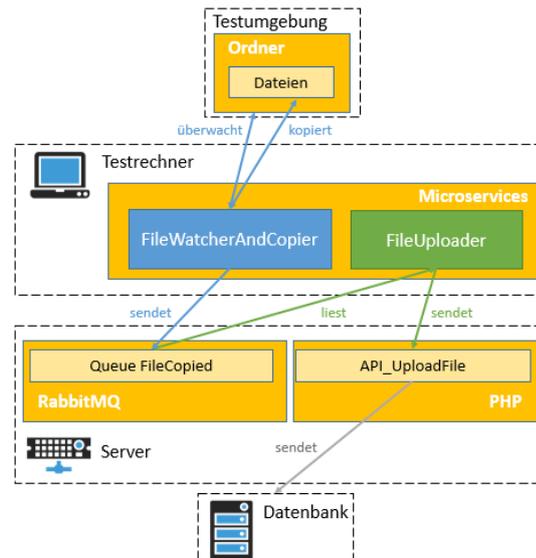


Abbildung 2: Verteiltes System – Aufbau der Microservice Architektur

Der Philosophie nach erledigt ein Microservice nur genau eine Aufgabe. Dies kann beim Service „FileWatcherAndCopier“ nicht komplett gewährleistet werden, da sonst keine Offline-Verfügbarkeit der Daten existiert. Der Grund ist folgender: Da der Testrechner im Falle der Erprobungsfahrt nicht mit dem Bosch-Netzwerk verbunden ist, müssen die Daten zwischengespeichert werden. Die Nachrichten für den Message-Queue-Server werden dabei in einem Textfile gepuffert. Sobald eine Verbindung vom Rechner zum Server besteht, werden die Nachrichten aus dem Textfile gelesen und versendet.

Ein Ausblick zeigt, wie die Daten weiter verarbeitet werden können. Bisher erstellen Tester manuell Fehlertickets in einem internen Ticketsystem. Diese enthalten eine Beschreibung und haben Anhänge wie Screenshots oder Traces. Dadurch, dass im entwickelten System alle Daten korrekt und gebündelt zur Verfügung stehen, können daraus automatisiert Tickets erstellt werden. Dies bietet Optimierungspotenziale beim Fehlermanagementprozess.

[1] Werner Zimmermann und Ralf Schmidgall, 2011, Bussysteme in der Fahrzeugtechnik, Springer Fachmedien Wiesbaden GmbH, 4. Auflage, S.260–263

[2] Sam Newman, 2015, Building Microservices: designing fine-grained systems, O'Reilly Media, S.4–7

Bildquellen:

- Abbildung 1: Marcel Trinks
- Abbildung 2: Marcel Trinks – enthält Icons der Robert Bosch GmbH

Konzipierung und Realisierung eines Plug-Ins als Schnittstelle zwischen einem Nachtfahrsimulator und dem Ansteuerungsalgorithmus eines adaptiven hochauflösenden Multipixel-Scheinwerfersystems für den Einsatz in zukünftigen Fahrzeuggenerationen

Konstantin Wacker*, Reinhard Keller, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Straßenverkehrssicherheit ist eines der wichtigsten Themen in der Automobilbranche. Durch die Entwicklung neuer Technologien und Systeme werden Autos immer sicherer. Während früher versucht wurde, die Folgen von Unfällen abzuschwächen, steht heutzutage die Prävention von Unfällen durch aktive Sicherheitssysteme im Vordergrund.

Ein relativ neues Feld innerhalb der aktiven Sicherheitssystemforschung befasst sich mit adaptiven Scheinwerfersystemen. Diese passen die Lichtverteilung je nach Streckensituation und Fahrbedingungen an, um einerseits die Sichtverhältnisse bei Nacht zu erhöhen und andererseits die Blendung von anderen Verkehrsteilnehmern zu vermeiden.

Dazu entwickelt und setzt die Daimler AG in ihren Fahrzeugen ein sogenanntes *Intelligent-*

Light-System (ILS) ein.

Die in aktuellen Serienfahrzeugen von Mercedes-Benz erhältliche ILS Generation verfügt mit ihren *Multibeam-LED*-Scheinwerfern bereits jetzt über 84 Hochleistungs-LED-Chips pro Scheinwerfer, die ein rein elektronisch gesteuertes und präzises Teilfernlicht ermöglichen [1].

Die nächste Generation von Scheinwerfersystemen sieht eine deutliche Steigerung der Pixelauflösung durch hochauflösende μ AFS-Module (*Mikro Adaptive Frontlighting System*) vor. Damit soll die Leuchtkraft und Präzision des ILS-Systems im zentralen Bereich vor dem Fahrzeug verbessert und die Anzahl der einzeln abblendbaren Objekte weiter erhöht werden.



Abbildung 1: Nachtfahrsimulator Autobahnszenario

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

Um das Potential dieser neuen Scheinwerfermodule voll ausschöpfen zu können, wird der Algorithmus zur Ansteuerung der Scheinwerfermodule in der Vorentwicklung komplett überarbeitet und auf die Möglichkeiten der neuen Scheinwerfer angepasst. Da dieser Prozess auf ausgiebiges Testen und Überprüfen des generierten Lichtbilds angewiesen ist, physische Prototypen der tatsächlichen Scheinwerfer allerdings teuer und Testfahrten in Versuchsfahrzeugen zeitaufwändig sind, setzt die Daimler AG nun eine Simulationsumgebung ein, um Änderungen am ILS-Algorithmus schon früh während der Entwicklung testen zu können.

Zu diesem Zweck wurde ein Nachtfahrssimulator zur Echtzeitsimulation hochdynamischer Lichtverteilungen von hochauflösenden Lichtsystemen entwickelt. Dieser ist in der Lage auf entsprechenden Monitoren die Simulation in einer *High-Dynamic-Range*-Darstellung (HDR) auszugeben, wodurch die Ausleuchtung der Umgebung und Blendeffekte mit realitätsnaher Helligkeit und Kontrast wiedergegeben werden, was eine Bewertung des Lichtbilds mithilfe der Simulation ermöglicht.

Für das Testen des im ILS verwendeten Algorithmus mit dem beschriebenen Nachtfahr-

simulator wurde im Rahmen der Bachelorarbeit die Schnittstelle zwischen Simulator und ILS-Algorithmus konzipiert und entwickelt.

Um während der Simulation die für das ILS relevanten Daten aus dem Simulationsszenario auszulesen, existiert als Schnittstelle des Simulators ein Sensor Plug-In, das in der Lage ist einen Sensor wie z. B. eine Kamera an einem Fahrzeug zu simulieren und gefilterte Umgebungsinformationen in Echtzeit zu liefern.

Ziel der Arbeit war es, dieses Sensor Plug-In zu erweitern, um die Funktionalität der später tatsächlich im Fahrzeug verbauten *Multi-Purpose-Camera* für die Simulation zu emulieren.

Als Rahmen für die im Plug-In umzusetzende Funktionalität diente die Spezifikation der *Multi-Purpose-Camera*, in der der gewünschte Funktionsumfang der Kamera und das Interface zum ILS-Algorithmus beschrieben sind. Dabei übersteigt die Kameralogik die reine Objekterfassung.

Die umzusetzenden Funktionen umfassten zudem situationsabhängige Sicherheitsgrenzen um erfasste Objekte, Dimmen von Verkehrsschildern, Vorhersage der Position von vom Sensor verlorener Fahrzeuge und Zusammenfassen dicht beieinander liegender Objekte, mit dem Ziel einer homogeneren Lichtverteilung zwischen den einzelnen Objekten.



Abbildung 2: Nachtfahrssimulator Landstraßenszenario

[1] <http://mercedes-benz.com>

Bildquellen:

- Abbildung 1-2: Daimler AG

Entwurf, Simulation und Implementierung von Hard- und Software eines Flugreglers für das Versuchs-Segelflugzeug Discus 2C des DLR

Patrick Wieland*, Werner Zimmermann, Karlheinz Höfer

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Zur Erforschung aerodynamischer, aeroelastischer und flugdynamischer Effekte in ihrer Reinform eignen sich Segelflugzeuge besonders gut, da sie – typischerweise – keine Störungsquellen wie Propeller oder Motoren aufweisen. Zu diesem Zweck unterhält das Deutsche Zentrum für Luft- und Raumfahrt (DLR) zwei Versuchs-Segelflugzeuge. Das neuere der beiden ist ein Schempp-Hirth Discus 2C (genannt „der Heilige“), welcher schon in der Bauphase speziell für seine Forschungsaufgaben vorbereitet wurde [1]. Der Heilige ist in seinen Flugeigenschaften komplett vermessen und in einem MATLAB SIMULINK Modell abgebildet.



Abbildung 1: Der Heilige auf dem Vorfeld mit geöffnetem Einbaukasten der Messanlage.

Ausgerüstet mit einer ausgeklügelten Messanlage, wird dieses Flugzeug jedes Jahr auf dem Sommertreffen der Interessengemeinschaft deutscher akademischer Fliegergruppen e.V. (Idaflieg) als Vergleichsflugzeug für die Flugleistungsmessung von Prototypen und bereits zugelassenen Mustern verwendet [2]. Dabei fliegen das zu vermessende- und das Vergleichsflugzeug in Formation diverse Geschwindigkeitsstufen ab, wobei durch hochgenaues GPS die Relativbewegung bestimmt wird. Mit Hilfe der bekannten Polare des Vergleichsflugzeugs kann so die Polare des zu vermessenden Flugzeuges bestimmt werden.

Die Aussagekraft der so ermittelten Daten hängt primär davon ab, wie genau die Geschwindigkeitsprofile erflogen wurden, wobei das menschliche Leistungsvermögen und die Phygoide der Flugzeuge (niederfrequente Flugbahnschwingung mit geringer Dämpfung zwischen potentieller und kinetischer Energie) die Hauptfehlerquellen sind.



Abbildung 2: Discus 2C (der Heilige) und DG 300 Elan-17 des DLR im Vergleichsflug.

Auch bei anderen Experimentalflügen ist eine genaue und reproduzierbare Ausführung der Manöver essenziell. Hierbei wird häufig durch eine vorgegebene Sequenz an Ruderausschlägen eine Schwingungsanregung vorgegeben und die Reaktion des Flugzeuges darauf gemessen. So können beispielsweise im Bereich der Aeroelastik die Eigenfrequenzen der Flugzeugstruktur bestimmt werden, wodurch auf das Flatterverhalten (Resonanzüberhöhung die zum Strukturversagen führen kann) des Flugzeuges geschlossen werden kann.

Aus der Forderung nach Reproduzierbarkeit der geflogenen Manöver und den Grenzen des menschlichen Leistungsvermögens entstand die Idee, den Heiligen mit einem Fly-By-Wire System auszurüsten. Folglich wurden elektrisch betriebene Linearmotoren an das Rudergestänge angebaut, mit welchen sich sowohl die Ruder steuern als auch deren Stellungen auslesen lassen. Im Fehlerfall kann der Pilot durch Abschaltung der Spannungsversorgung der Motoren problemlos die volle Kontrolle über das Flugzeug wiedererlangen.

Es gilt also ein System zu entwickeln, welches es ermöglicht, vorkonfigurierte Manöver automatisiert, exakt und reproduzierbar zu

*Diese Arbeit wurde durchgeführt bei der Firma Deutsches Zentrum für Luft- und Raumfahrt, Braunschweig

fliegen.

Der Entwurf besteht aus einem Rechner, welcher die Kommunikation mit den Peripheriekomponenten ermöglicht und auf dem der Flugregler ausgeführt wird. Die Datenbasis für den Flugregler bilden das Global Positioning System für die Position und Geschwindigkeit über Grund, eine inertielle Messeinheit für Lage, Beschleunigungen und Rollraten und eine Fünflochsonde, mit der sich die Geschwindigkeit gegenüber der umgebenden Luft, der Anstell- und der Schiebewinkel bestimmen lassen. Der Pilot kann den Flugregler durch ein Eingabegerät steuern und bekommt Zustandsdaten über ein Display im Instrumen-

tenpizl angezeigt. Der Flugregler wird auf Basis des MATLAB SIMULINK Modells des Heiligen erstellt, getestet und anschließend auf die Zielhardware portiert.

Die eingesetzte Hardware muss Schnittstellen für die genannte Peripherie und für die Ansteuerung der Motoren zur Verfügung stellen. Zudem muss das gesamte System robust genug sein, um den Beanspruchungen im Flug sicher stand zu halten, leistungsfähig genug, um alle Aufgaben in Echtzeit abarbeiten zu können und einfach genug, dass es ohne Vorkenntnisse in Programmierung oder Elektrotechnik verwendbar ist.

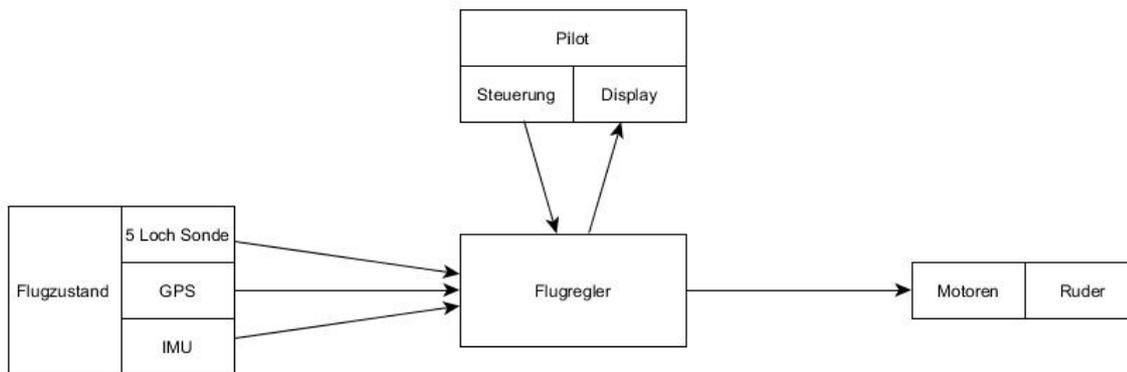


Abbildung 3: Schematische Darstellung des Gesamtsystems.

1 DLR, Forschungsflugzeuge Discus-2c DLR http://www.dlr.de/dlr/desktopdefault.aspx/tabid-10203/339_read-9181/#/gallery/8791

2 Idaflieg e.V. Flugleistungsvermessung <http://idaflieg.info/flugleistungsvermessung/>

Bildquellen:

- Abbildung 1–2: Dipl.–Ing. Kai Rohde-Brandenburger, DLR
- Abbildung 3: Eigene Darstellung

Imitation of Life – Die Autonomie moderner UI-Konzepte vom Skeuomorphismus zum Flat Design

Hatice Yildirim*, Astrid Beck, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Die Benutzeroberfläche (engl. User Interface, kurz UI) stellt die wichtigste Schnittstelle zwischen Mensch und Computer dar, über die der User eine Software oder Hardware kontrollieren kann. Hierbei versteht man die Art und Weise, wie bestimmte Daten und Befehle in den Computer eingegeben und von diesem verarbeitet werden. Die Benutzeroberfläche wird vom Betriebssystem oder den Anwendungsprogrammen bereitgestellt und soll dem User das Interagieren ermöglichen und vereinfachen. Ein ästhetisches graphisches De-

sign wirkt daher positiv auf das Nutzungserlebnis (eng. User Experience) eines Benutzers aus und ist neben einer guten Gebrauchstauglichkeit (engl. Usability) ein weiterer wichtiger Bestandteil für den Erfolg eines Produktes oder einer Software [1].

Im Laufe der Zeit haben sich verschiedene Herangehensweisen an das User Interface Design ergeben. Im Folgenden werden zwei dieser Entwicklungen, die auch die Grundlage für die Bachelorarbeit bilden näher erläutert.



Abbildung 1: Skeuomorphismus

Mit dem Launch des ersten iPhones der Firma Apple wurden im Laufe der Zeit im Bereich des mobilen Interface Designs neue Wege eingeschlagen. Die Designrichtung des iOS-Betriebssystems nannte sich „Skeuomorphismus“ und sollte dem Nutzer den Umgang mit dem Rechner oder dem Smartphone einfacher und interessanter gestalten.

Der Begriff Skeuomorphismus leitet sich von den griechischen Begriffen „Skeuos“ für das Wort Werkzeug und „morph“ für Form ab.

Auf Wikipedia wird Skeuomorphismus wie folgt definiert:

„... eine Stilrichtung hauptsächlich im Design, bei der Objekte in ihrer Gestaltung ein anderes Material

oder eine Form eines älteren, vertrauten Gegenstandes nachahmen, ohne dass diese durch ihre Funktion gegründet ist [2].“

Naturalistisch beschreibt Skeuomorphismus die Strategie, bekannte Gegenstände und Werkzeuge aus dem analogen Leben digital in gleicher Weise abzubilden. Um den Computer für Neueinsteiger besser zugänglich zu machen, entwickelte Apple eine grafischen Benutzeroberfläche, die eine visuelle Metapher zwischen neuen Konzepten und vertrauten Objekten bildete. Als Ergebnis wurde die GUI mit Darstellungen von vertrauten Objekten gefüllt, die es in der digitalen Welt zuvor noch nie gegeben hat. Nicht verwendete Dateien gehören

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

in den Papierkorb, Notizen hält man auf Notizzettel fest, wichtige Termine schreibt man in einen Terminkalender und Bücher kommen in das Bücherregal. Da neue User etwas ängstlich an neue Entwicklungen herangingen, vereinfachten solche Metaphern den Einstieg in das neue System.

Vom Skeuomorphismus zum Flat Design

Skeuomorphismus diente seinerzeit als die beste Designlösung für Smartphones, insbesondere für die Smartphone Produktreihe von Apple. Viele User Interface Designer folgten diesem Konzept und gestalteten ihre Produktoberflächen nach diesen Prinzipien, jedoch machte sich die schlechte Usability hinter diesem System schnell bemerkbar. Ebenso waren die Ladezeiten der einzelnen Elemente viel zu lang, sowie der gestalterische Aufwand viel zu hoch.

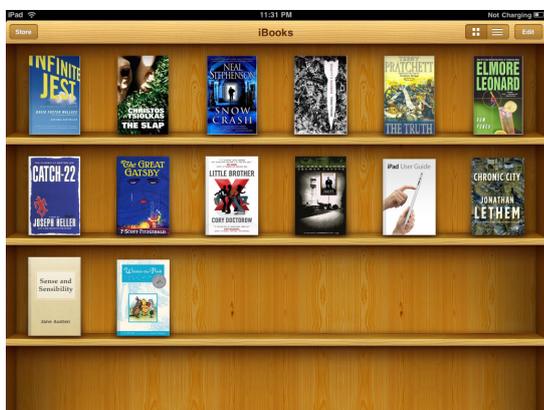


Abbildung 2: iBooks Bücherregal von Apple

Deshalb musste eine neue Designrichtung her: „Flat Design“ heißt der neue Trend und ist mehr als nur vorübergehend.

Seit 2013 zeigt sich der Trend zum „flachen Design“ in vielen Gestaltungen. Das Flat Design fand zum ersten mal Anwendung in der Gestaltung von Smartphone-Applikationen. Den großen Durchbruch machte das Flat Design mit dem Launch des Betriebssystems Windows 8 und Windows Phone 8 von Microsoft im sogenannten Metro-Design. Der Begriff „Metro“ wurde von Microsoft als Codename für diese neue Designrichtung eingeführt. Im Laufe der Zeit wurde es jedoch in „Microsoft Modern UI“ umbenannt. Die neue Oberfläche von Windows war vor allem auf die Bedienung mit den Fingern ausgelegt und setzte aus diesem Grunde auf eine sehr einfache visuelle Gestaltung.

Durch diese Entwicklung hatte sich der neue Trend des Metro-Designs in einer recht kurzen Zeit zu einer großen Entwicklung im Bereich des User Interface Designs entwickelt.

Folgende Prinzipien werden beim Flat Design nach den Grundsätzen „Die Form folgt der Funktion“ und „Weniger ist mehr“ verfolgt [3]:

- Keine 3D-Elemente
- Verzicht auf Schatten, Verläufe oder Lichteffekte
- Hervorstechende Typografie

- Wechselnde und viele Farben
- Klare Linien, geometrische Formen
- Sehr einfach designte Icons
- Aufbau im Kachel-/Rastersystem

Auch die Firma Apple Inc. folgte diesem Trend nach kurzer Zeit mit dem Launch des Smartphone Betriebssystems iOS 7. Die wichtigsten Neuerungen im Redesign der Oberfläche waren: Alle Oberflächen-Objekte, die an natürlichen Gegenstände wie Leder, Holz, Metal etc. erinnern, waren verschwunden. Ebenso wurde auf 3D Effekte bei Eingabefeldern und Rahmen verzichtet.



Abbildung 3: Windows 8 im Flat Design

Flat Designs vs. Google Material Design

Die visuelle Sprache des Flat Designs ist sofort zu erkennen: große Inhaltsflächen und aussagekräftige Bilder, deutlich erkennbarer Rastersystem, mehr Weißflächen und serifenlose Schriftarten. Durch den Einsatz von Flat Design wirkt die grafische Oberfläche einer Software aufgeräumter und übersichtlicher. Beim Flat Design steht ganz klar der Inhalt einer Anwendung im Vordergrund, daher sollte man möglichst auf unnötige Elemente verzichten.

Jedoch hat auch diese Designrichtung eine Schattenseite: Bei einem Usability-Test der Agentur Usability.de hat sich herausgestellt, dass bei der Smartphone-Oberfläche von Windows 8 die Testpersonen große Schwierigkeiten beim Erkennen von klickbaren Elementen hatten. Daraus lässt sich erschließen, dass eine zu stark abstrakte Gestaltung auch zu Fehlinterpretationen führen kann [4].

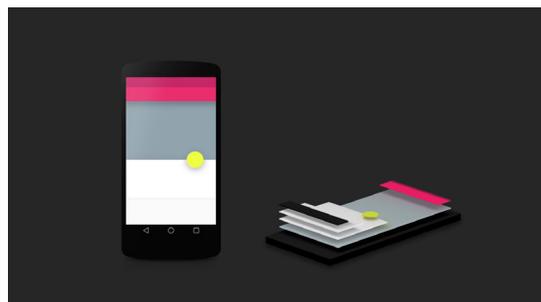


Abbildung 4: Konzept des Material Designs von Google

Ein möglicher Lösungsansatz für dieses Problem könnte das "Material Design" von Google sein. Anders beim Flat Design bei dem es in erster Linie um die Reduzierung von Details geht, beschäftigt sich das Material Design schwerpunktmäßig mit der Benutzerfreundlichkeit und die Einhaltung physikalischer Gesetze. Anders als im Flat Design wird hier mit Licht und Schatten gearbeitet um Zusammen-

hänge und klickere Elemente als auch solche zu visualisieren.

Letztendlich lässt sich erschließen, dass das Material Design von Google womöglich eine positive Weiterentwicklung von Flat Design in Zukunft sein kann, um eine noch bessere User Experience und Usability zu erzeugen.

[1] <http://www.itwissen.info/definition/lexikon/Benutzeroberflaeche-UI-user-interface.html>

[2] <https://de.wikipedia.org/wiki/Skeuomorphismus>

[3] <http://t3n.de/news/flat-design-webdesign-trend-438941/>

[4] <http://heftarchiv.internetworld.de/2013/Ausgabe-13-2013/Flach-wie-ein-Brett>

Bildquellen:

- Abbildung 1: <http://www.dtelepathy.com/blog/inspiration/50-skeuomorphic-designs>
- Abbildung 2: <http://webdevs.co/skeuomorphism-vs-flat-design/>
- Abbildung 3: <http://early-adopter.com/flat-vs-skeuomorphic/>
- Abbildung 4: <http://venturebeat.com/2014/06/27/top-designers-react-to-googles-new-material-design-language/>

Automatisierte Sicherheitsanalyse von Datenverkehr mit Intrusion Detection in proprietären Protokollen

Marcel Zeller*, Martin Zieher

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2016/2017

Sicherheit in der Informationstechnik hat längst einen hohen Stellenwert. Daten- und Systemzugriff, Integrität und Verfügbarkeit sind die Werte, die geschützt werden müssen. Zwar werden immer wieder große Angriffe bekannt, die Sicherheitslücken ausnutzen und so hohen Schaden anrichten, doch die Lücken werden meist schnell behoben. Daher sind Denial-of-Service (DoS)-Angriffe gängig, die sich vermehrt, statt gegen die Systeme an sich, gegen ungeschützte DNS-Server richten, die plötzlich DoS-Angriffen in noch nie dagewesener Größe ausgesetzt sind. Doch auch für den unerlaubten Daten- und Systemzugriff gibt es längst neue Vorgehensweisen. Offizielle Schnittstellen werden dazu ausgenutzt, vermeintlich belanglose Daten zu gewinnen. Korrekt aggregiert bieten diese Daten neue Angriffsvektoren, die auch das Nutzen einer normalen Login-Maske umfassen können und so zum eigentlichen Ziel des Angriffs führen.

Sicherheitsprobleme

Eine häufig anzutreffende Vorgehensweise bei der Softwareentwicklung begünstigt mögliche Angriffe. Diese zielt auf Funktionalität ab, während Nicht-Funktionalität nur selten relevant ist. Beim Thema Sicherheit handelt es sich jedoch meist um Einschränkungen der Funktionalität für bestimmte Nutzer in bestimmten Nutzungsszenarios. Des Weiteren wird die Bedeutung von Daten häufig unterschätzt. Tatsächlich hat „vom Prinzip her [...] jedes versandte Paket eine Bedeutung, die nicht zu vernachlässigen ist“ [1]. So sind bei einer riesigen Anzahl von Online-Diensten unterschiedliche Fehlermeldungen beim Login für eine E-Mail-Adresse, der kein Account zugeordnet ist, und einem ungültigen Passwort gängig. Angreifer können so Einträge in gestohlenen Passwortlisten von Drittanbietern effizient herausfiltern, für die es im angegriffenen Dienst gar keinen Account gibt und in Folge die durch die bekannten Passwörter von anderen Diensten ohnehin schon schnelle Brute-Force-Angriffe um ein Vielfaches beschleunigen.

Gegenmaßnahmen sind hierbei mit Sicherheitssystemen, wie sie gegen Sicherheitslücken genutzt werden, nicht umzusetzen. Die

Schwierigkeit liegt in der Unterscheidung zwischen Zugriffen von normalen Nutzern und solchen von Angreifern. Ein Ansatz kann die DoS-Abwehr sein, doch bewegt sich die Zahl der Anfragen dieser Angreifer durch deren Vorgehensweise „Ausprobieren“ zwar auf einem deutlich erhöhten Niveau gegenüber einem normalen Nutzer, allerdings bei weitem nicht auf dem Niveau eines DoS-Angriffs. Damit würden Anti-DoS-Systeme zu viele „false-positives“ liefern.

Intrusion Detection System

Eine erheblich aussichtsreichere Lösung stellt ein Intrusion Detection System (IDS) dar. Hierbei werden verschiedene Kennzahlen und Muster herangezogen oder berechnet und mit Grenzwerten und Normalverhalten verglichen. Zwei grundlegend unterschiedliche Herangehensweisen sind möglich: Ein **Host-basiertes System**, das Events des Betriebssystems und eigener Software nutzt sowie ein **Netzwerk-basiertes System**, das den Netzwerktraffic überwacht [2].

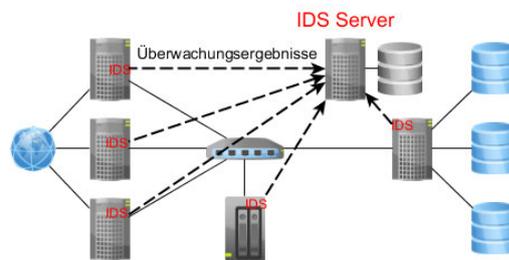


Abbildung 1: Host-basiertes IDS: lokale Überwachung, Ergebnisse werden gesammelt

In der vorliegenden Abschlussarbeit wurde aufgrund der komplexen und vor allem skalierenden Netzwerkinfrastruktur ein Netzwerk-basiertes IDS umgesetzt, das sich leichter der sich ändernden Umgebung anpassen kann, wengleich die Logik des Systems an bereits bestehende Server angegliedert wurde und nicht, wie sonst bei netzwerk-basierten Systemen üblich, separat läuft. Das System soll dabei hinter einer vorgelagerten Serverumge-

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

bung laufen. **Methoden zur Überwachung** Zur Überwachung gibt es verschiedene Methoden, die in die folgenden Bereiche eingeteilt werden können [3]:

- **Veränderungsanalyse:** Periodisches Prüfen auf Änderungen, wobei sowohl definiertes als auch dynamisch aktualisierendes Normalverhalten herangezogen werden kann.
- **Aufrufsüberwachung:** Bestimmte Aufrufe oder Kombinationen aus Aufrufen werden als verdächtig kategorisiert, gegen die dann verglichen wird.
- **Angriffssignaturen:** Muster aus vergangenen Angriffen werden gespeichert und mit aktuellen Werten auf Ähnlichkeiten verglichen.
- **Auslastungsanalyse:** Die Auslastung eines Systems oder eines Netzwerkes wird auf Grenzwerte oder Normalverhalten hin überprüft.
- **Nutzungsprofile:** Profile einzelner Nutzer oder Gruppen werden erstellt, wodurch Änderungen des Nutzungsverhalten erkannt werden können.

Je nach Anwendungsfall müssen daraus die richtigen Methoden gewählt werden. Allerdings birgt jede Methode auch spezielle Schwierigkeiten wie Verfügbarkeit von historischen Vergleichsdaten, schwierig festzulegenden Grenzwerten oder Einschränkungen durch den Datenschutz.

Datengrundlage Es ist sowohl möglich, aufgezeichnete Daten zu nutzen, als auch darauf zu verzichten. Zur Unterscheidung gibt es die Begriffe „**misuse detection**“ und „**anomaly detection**“ [4]. Die Methodik der ersten Variante sieht vor, ungewolltes Verhalten auf Basis alter Angriffe, ungewünschter Systemzustände oder ähnlichem zu definieren und so Abweichung im Live-System festzustellen. "False-positives" können gut vermieden werden, neuartige, bislang unbekannte Angriffe bleiben jedoch unerkannt. Hier setzt der zweite Ansatz an, bei dem das aktuelle Systemverhalten nach Änderungen überwacht wird. Die Erkennung beschränkt sich so nicht auf bereits

bekannte Angriffe, erhöht jedoch die Rate der „false-positives“ merklich.

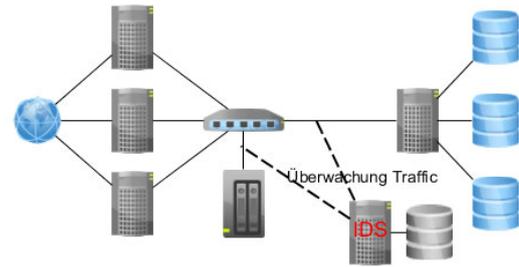


Abbildung 2: Netzwerk-basiertes IDS: Überwachung des Traffics

Eine weitere Eigenschaft eines IDS besteht darin, meist **keine automatischen Gegenmaßnahmen** auszuführen. Zwar gibt es Möglichkeiten, an das System Gegenmaßnahmen wie eine Sperrung oder Drosselung einer IP anzuschließen, doch bei einer sehr große Anzahl von Nutzern mit unterschiedlichen Anwendungsfällen, wie in diesem Projekt der Fall, müssten automatische Gegenmaßnahmen sehr flexibel ausgelegt sein. Insbesondere vor dem Hintergrund von inkorrekten Funden, stellt es sich oft als die bessere Lösung dar, Entwickler oder Administratoren die Ergebnisse des IDS auswerten und die richtigen Gegenmaßnahmen anstoßen zu lassen.

Im Projekt wurde ein IDS auf Basis der „anomaly detection“ umgesetzt, das an die zuständigen Entwickler melden soll, wenn ungewöhnliches Verhalten auftritt. Von Bedeutung war dabei, eine möglichst hohe Erkennungsrate im komplexen und dynamischen System zu erreichen, während gleichzeitig sehr wenige „false-positives“ auftreten. Eine zu hohe Zahl von solchen falschen Alarmen würde, bedingt durch die sehr hohe Zahl der Nutzer, die Entwickler überfluten und das System damit unbrauchbar machen. Tatsächlich kann eine **korrekte Konfiguration** allerdings erst im laufenden Betrieb aufgrund der erhaltenen Alarme vorgenommen werden und muss auch danach regelmäßig überprüft und gegebenenfalls angepasst werden.

[1] Ziegler, P.S. 2008. Netzwerkangriffe von innen. Köln: O'Reilly Verlag

[2] Mukherjee, B., Heberlein, L.T., Levitt, K.N. 1994. Network intrusion detection. IEEE network <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.219.4284&rep=rep1&type=pdf> [Zugriff am 9.11.2016]

[3] Plötner, J., Wendzel, S. 2005. Praxisbuch Netzwerk-Sicherheit. Bonn: Galileo Press

[4] Kemmerer, R.A., Vigna, G. 2002. Intrusion Detection: A Brief History and Overview. Computer IEEE <http://www.ismlab.usf.edu/isec/files/kemmerer2002.pdf> [Zugriff am 26.10.2016]

Bildquellen:

- Abbildungen 1-2: eigene Abbildungen

