

IT-Innovationen

Band 12
Januar 2014

Grußwort des Dekans

Liebe Leserinnen und Leser,

Der digitale Wandel ist allgegenwärtig. Maschinen in Fabriken kommunizieren miteinander, der heimische Heizkessel ist mit dem Internet verbunden, der reibungslose Betrieb der Ladesäulen von E-Autos ist selbstverständlich. In Haushaltsgeräten, Produktionsmaschinen und Fahrzeugen steckt zunehmend komplexe Software. Komfort, Energieeffizienz und Sicherheit von Autos werden maßgeblich durch Software bestimmt. Antriebs- und Assistenzsysteme kommen ohne Steuerungssoftware nicht aus. Vernetztes und automatisiertes Fahren sind ohne Informationstechnik unmöglich. In der Medizintechnik arbeiten Softwareentwickler an der Steuerung von Prothesen und suchen nach Lösungen für die Tele-Medizin. Ob Industrie, Infrastruktur, Medizintechnik, Energie, Handel oder Logistik, die Arbeitsfelder der Informationstechniker sind vielfältig und gehen quer durch alle Branchen. Dabei ist entscheidend, in die jeweils einzelnen Fragestellungen hineinzuwachsen und die dort anstehenden Aufgaben zu lösen. Das ist der rote Faden des Studiums der Informationstechnik in Esslingen. Überzeugen Sie sich selbst in den Ihnen vorliegenden IT-Innovationen von der Vielfältigkeit der Arbeitsfelder und der Aufgabenstellungen und gewinnen Sie einen Eindruck vom hohen Ausbildungsstand unserer Absolventen.



Viele Anregungen beim Lesen wünscht Ihnen Ihr

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

Peter Dück, 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

Florian Ackermann	Entwurf und Implementierung eines Kommunikationssystems mit öffentlichen, mobilen Komponenten	1
Sabine Aesch	Konzeption eines intelligenten Datenloggers zur Aufzeichnung von Messdaten auf Bussystemen im Fahrzeug und Realisierung mit einem autonomen Embedded System unter C und C#	2
Roman Asanov	Simulationsumgebung für verteilte Ladeinfrastruktur	3
Oliver Breuning	Entwurf und Implementierung einer Regelung für ein Gyroskopstabilisiertes einachsiges Lastentransportsystem einschließlich Optimierung der Regelparameter mit Matlab/Simulink und Erweiterung der hardwarenahen Software des eingebetteten Echtzeitsystems auf Basis eines ARM-Cortex-Prozessors und FreeRTOS	5
Kathrin Bromberg	Manöverprädiktion im urbanen Verkehr Datenanalyse und Auswertung mit MATLAB	7
Philipp Dentsas	3D-Modellierung eines Stadtteils mit 3ds Max	8
Alexander Diel	Interaktive Visualisierung eines komplexen Graphen	10
Kristian Dürsch	Erstellung einer Augmented Reality App zur Visualisierung von Windkraftanlagen, für bessere Bürgerbeteiligung mit neuen Technologien.	11
Kai Dybionka	Entwicklung eines Programms zur Durchführung von Lasttests aus der Cloud	13
Daniel Eisele	Verschlüsselung mit elliptischen Kurven im Embedded Bereich	14
Philipp Erler	Decision Making in a Group of Artificial Intelligences in Different Simulated Environments	16
Dawinja Falter	Evaluation verschiedener Webtechnologien und prototypische Portierung des Konzepts zur Bildung von Oberflächenelementen	18
Markus Foitschik	Konzeption und Implementierung einer protokollunabhängigen Remote-Support-Anbindung von TeamViewer an einen Headless Server unter Linux	21
Ann-Kathrin Frey	Usability für immersives Unterrichten und Lernen: Am Beispiel Cyber-Classroom School Edition	24
Ilja Galat	Simulation von Fahrzeugsensoren zur Umgebungserfassung für Fahrerassistenzsysteme in OpenCL unter Verwendung einer GPU einschließlich Performancevergleich der GPU-Implementierung mit bestehenden Simulationen auf Basis einer CPU	26
Thomas Gottuck	Design and development of a test-framework for a non-locally deterministic workflow engine	28
Martin Heinzmann	Realisierung einer Musikabspielplattform zur verbesserten Demonstrationsfähigkeit einer Class-D Audioendstufe im Bereich der Modelleisenbahn.	30
Simon Helming	Design und Implementierung eines Objektbaums für die Abbildung des AUTOSAR Virtual Function Bus in Python	32
Lukas Janssen	Konzeption einer Systemsimulation für Aufzugssysteme sowie die beispielhafte Umsetzung ausgewählter Komponenten	34

Manuel Kurfiß	Design, Implementierung und Test eines Analysetools in Java zur Detektion von EMS-Bus-Protokoll-Verletzungen sowie eine statistische Auswertung und Visualisierung dieser Daten	36
Daniel Lombardo	Entwicklung einer Referenzimplementierung für plattformunabhängige Hybrid-Apps im Enterprise-Umfeld und Gegenüberstellung von UI-Frameworks	38
Max Mairle	Konzeption eines Embedded Systems für die Rückspeisung der Bremsenergie eines Aufzugs über einen Frequenzumrichter sowie Realisierung von Regelungsfunktionen mit Generierung des Codes aus Simulink für einen Cortex-M4F-Mikrocontroller einschließlich Aufbau eines Prototyps und Durchführung von Messungen	39
Marc Miller	Konzeption, Umsetzung und Verifikation eines Wireless Local Area Network (WLAN) Systems zur drahtlosen Übertragung von Mess- sowie Applikationsdaten zwischen Fahrzeug und Kommandostand im Motorsport	41
Marc-Michael Müller	Entwicklung und Integration eines ServiceFollowing-Systems zwischen DAB- und Webradio innerhalb eines Fahrzeugdemonstrationssystems unter Verwendung einer Servicedatenbank	43
Miguel Oesterlein	Entwicklung eines hardwareunabhängigen WLAN-Stacks für den Einsatz in einem CAN Gateway zur Aufzeichnung von Diagnose- und Statusinformationen	44
Thomas Pfeiffer	Web basierte Datenerfassung mittels RFID-Funkstrecke	46
Peter Plötze	Entwurf und Implementierung eines Protokolls zur Auslagerung von Steuergeräte-Funktionalitäten auf einen PC unter Einsatz von MATLAB/Simulink-Modellen	47
Michael Popow	Integration einer automotiven Müdigkeitserkennung auf Android	49
Lia Schepke	Ein modularer Content-Server für den Test von Connected Car Modulen mit Fokus auf Telefonie und Web	50
Alexander Schoch	Analyse charakteristischer Sensorsignale und Modellierung eines Regelkreises zur dynamischen Ansteuerung eines Aktors	52
Valon Sejdijaj	Connectivity am Beispiel des Elektrofahrzeugs eScooter LiBERTé-App-Entwicklung und Entwicklung der Kommunikation zwischen eScooter und mobilem Endgerät	53
Thies Spannkebel	Konzeption und Entwicklung eines Systems zur Belegungserkennung und Reservierung von Parkplätzen	54
Ruven Spiller	Gegenüberstellung von Client-Server-Frameworks auf der Basis von JavaEE und Node.js im Enterprise-Umfeld	55
Dominik Spindler	Integration eines Lua-Skript-Interpreters in ein Komponentenframework zur flexiblen Erweiterung des Funktionsumfangs von Embedded-Sensoren über nachladbare Skripte	57
David Stier	Portierung eines Mikrocontroller basierten, echtzeitfähigen Bus-Kommunikationsinterfaces auf Zielsystem PowerPC mit embedded FPGA	58
Thomas Stoof	Effiziente Implementierung eines modellbasierten Algorithmus zur Verarbeitung des Fahrerwunsches an einem eBike	60

Samed Sulanc	Entwurf, Aufbau und Programmierung eines mechatronischen Systems zur manuellen motorunterstützten Führung eines elektrisch angetriebenen Fahrzeugs	62
Mesih Tasci	Konzeption, Aufbau und Softwareentwicklung eines modularen Messdatenerfassungssystems zur Erprobung von Reinigungsmaschinen	64
Nick Newill Tchouante Kembe	Profiling von Software in C und C++ anhand eines realen Embedded Projektes	65
Bernd Theissler	Konzeption und Realisierung eines Test-Management-Werkzeugs für Fahrzeugtests als Windows 8-Tablet-App auf Basis von C# und XAML	67
Tobias Tropper	Nutzerorientierte Plattform für den privaten Verleih von Büchern, Musik, Spielen und Filmen mittels einer Android-Applikation	69
Ivan Vishev	Entwicklung und Integration eines Steer-by-Wire Systems für ein modulares Elektrofahrzeug	71
David Zentner	Konzeption und Entwicklung einer Monitoring-Funktion für die Zustände von OS-Tasks und für AUTOSAR-Runnables	73

Entwurf und Implementierung eines Kommunikationssystems mit öffentlichen, mobilen Komponenten

Florian Ackermann*, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Mit dem Beginn der hohen Verfügbarkeit von schnellen Internetzugängen und der schnellen Weiterentwicklung der Informationstechnologie im Allgemeinen, steigt auch die Attraktivität von Videokommunikationsplattformen. Im Speziellen entsteht zur Zeit das vielversprechende, frei nutzbare Peer-to-Peer-Protokoll WebRTC [1] welches eine hohe Verbindungsqualität bietet. Um mit dieser Technologie nun Gewinne zu erwirtschaften, wurde ein Projekt des Startup-Unternehmens „avatimes“, gestartet. Das Projekt verbindet die Kunden eines, eigens hierzu eingerichteten Onlineshops, mit Mitarbeitern von „avatimes“. Diese Mitarbeiter agieren dann, mit einem Mobilgerät ausgestattet, als Augen und Ohren des Kunden. Die Videokommunikation wird automatisiert hergestellt. Automatisiert bedeutet in diesem Fall, dass ein Hauptprogramm die erforderlichen Daten zusammen trägt, Mitarbeiter und Mobilgerät selbstständig dem Auftrag zuordnet, und zum gewünschten Startzeitpunkt die Übertragung startet.

Alle Daten werden in einer relationalen MySQL-Datenbank gespeichert. Dies ermöglicht einen einfachen und vor allem schnellen Zugriff. Das Finden eines zuständigen Mitarbeiters erfolgt nach folgendem Muster: Es wird zunächst via SMS eine Anfrage an alle verfügbaren Mitarbeiter geschickt. Eine Zusage des Mitarbeiters erfolgt ebenfalls via SMS.

Diese Zusage wird dann vom Hauptprogramm entsprechend interpretiert. Die Mobilgeräte befinden sich in Schließfächern. Um ein Mobilgerät einem bestimmten Mitarbeiter zu einer bestimmten Zeit zur Verfügung zu stellen, kann die, zu diesen Schließfächern gehörende, Verwaltungssoftware, über eine XML-Schnittstelle angesprochen und jederzeit neu programmiert werden. Die Realisierung der eigentlichen Videokommunikation erfolgt schlussendlich über das bereits erwähnte WebRTC.

Das System kann von beliebig vielen Benutzern gleichzeitig genutzt werden. Dies wird durch Multithreading realisiert. Es kommen außerdem mehrere Konzepte der Objektorientierung zum Einsatz, welche eine effiziente Programmierung erleichtern. Eine Steuerkonsole, welche via eines Sockets mit dem Hauptprogramm kommuniziert, gibt dem Systemadministrator jederzeit die Möglichkeit händisch einzugreifen.

Ziel ist es, eine Plattform zu schaffen, welche für verschiedene Anwendungsfälle zur Verfügung steht. Im Allgemeinen ermöglicht das System Menschen, an wichtigen Orten präsent zu sein, obwohl sie eigentlich durch logistische, finanzielle oder zeitliche Gründe, verhindert wären.

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

[1] <http://www.webrtc.org> 12/2013

Konzeption eines intelligenten Datenloggers zur Aufzeichnung von Messdaten auf Bussystemen im Fahrzeug und Realisierung mit einem autonomen Embedded System unter C und C#

Sabine Aesch^{*}, Joachim Goll, Manfred Dausmann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Von sicherheitsrelevanten Funktionen innerhalb des Automobils wie beispielsweise ABS oder ESP bis hin zur Kommunikation mit der Außenwelt – ein modernes Auto ist heutzutage viel mehr als einfach nur ein fahrbarer Untersatz. Je größer der Funktionsumfang, desto komplizierter und schwieriger wird die Umsetzung. Immer mehr Steuergeräte kommen hinzu, die über verschiedene Busse miteinander kommunizieren müssen (siehe Abbildung 1).



Abbildung 1: Vernetzung der Steuergeräte im Kraftfahrzeug

Das Erkennen von Fehlverhalten einzelner Steuergeräte oder des gesamten Systems muss nicht erst während der Wartung in der Werkstatt, sondern auch bereits während der Entwicklung und Fertigung sichergestellt sein. Dafür werden Diagnosewerkzeuge eingesetzt, welche mit den einzelnen Steuergeräten über einen Bus kommunizieren und den Kommunikationsverlauf aufzeichnen können [1].

Ein weit verbreitetes Diagnosewerkzeug ist samDia der Firma Samtec Automotive Software & Electronics GmbH. Über die samDia-Software lässt sich die Hardware, in diesem Fall das HSX-Interface, konfigurieren. Das HSX-Interface wiederum ist dazu in der Lage, direkt mit einem Kfz-typischen Bussystem zu kommunizieren.

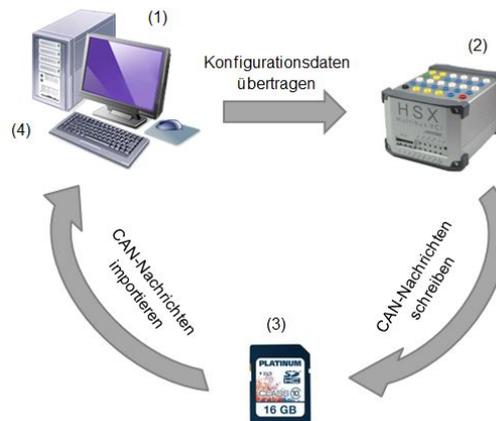


Abbildung 2: Projektaufbau

Das durchgeführte Projekt beschäftigt sich mit dem Aufzeichnen und Auswerten von CAN-Nachrichten und der Diagnosekommunikation auf einem CAN-Bus. Dabei lässt sich das Projekt in verschiedene Teilschritte, wie sie in Abbildung 2 zu sehen sind, aufteilen. Das Verhalten der Hardware wird über das Schreiben von C-Skripten innerhalb der samDia-Software gesteuert (1). Die so erzeugten Konfigurationen werden anschließend auf das HSX-Interface übertragen. Im nächsten Schritt wird die Verbindung von PC und HSX-Interface getrennt und das HSX-Interface über einen CAN-Bus an ein Fahrzeug oder einzelne Steuergeräte angeschlossen. Nun ist das HSX-Interface dazu in der Lage, CAN-Nachrichten gemäß der vorangegangenen Konfiguration aufzuzeichnen (2) und diese auf eine SD-Karte zu schreiben (3). Zur Auswertung der Daten wird die SD-Karte dem HSX-Interface entnommen und an einen PC angeschlossen. Eine eigenentwickelte Analyse-Software importiert die aufgezeichneten Daten, bereitet diese auf und stellt sie schließlich am Bildschirm dar (4).

^{*}Diese Arbeit wurde durchgeführt bei der Firma IT Designers GmbH, Esslingen

[1] Werner Zimmermann, Ralf Schmidgall: Bussysteme in der Fahrzeugtechnik, Vieweg+Teubner Verlag, 4. Auflage, 2011

Bildquellen:

- Abbildung 1: IT Designers GmbH
- Abbildung 2: Eigene Darstellung

Simulationsumgebung für verteilte Ladeinfrastruktur

Roman Asanov*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

“Gehören Elektroautos die Zukunft? Manches spricht dafür, etwa die fehlende Abgasbelastung am Auto. In Zeiten stetig steigender Benzinpreise und gleichzeitig steigender Nachfrage nach Energie, scheinen die Entwicklungen im Bereich der Elektromobilität vielversprechend zu sein. Das Prinzip der Elektromobilität ist denkbar einfach: Batterien treiben die Fahrzeuge, anstelle eines Verbrennungsmotors, an und sorgen dadurch – zumindest vordergründig – für CO₂-freie Fortbewegung.“ [1] Momentan ist die Technologie der Elektromobilität noch nicht fortgeschritten und wird deshalb weiterentwickelt. Es existiert bereits eine kleine Anzahl von Elektroautos in Deutschland. Auch BMW startete vor kurzem eine Serienproduktion eines Kleinwagens i3. Zum heutigen Stand ist es möglich Elektrofahrzeuge mit Strom aufzuladen. Da aber die Batterien noch zu wenig Kapazität besitzen, ist es momentan nicht möglich, längere Strecken zu fahren.

Der gesamte Aufladevorgang geschieht über eine Ladestation, die identisch bedienbar zu einer Tankstation ist. Mehrere Ladestationen sind in der Regel an ein zentrales System angebunden und kommunizieren mit diesem. Diese Infrastruktur bietet einen Vorteil wie z. B. Wartung, da alle Ereignisse der Ladestationen an zentrales System gemeldet werden. Auf diese Weise lässt sich feststellen, ob die Stationen aktiv bzw. inaktiv sind, Fehlermeldungen vorhanden sind und wie lange an einer Station bereits aufgeladen wurde. Ein weiterer und wesentlicher Vorteil ist, dass der Aufladevorgang zeitlich steuerbar ist. Es ist also möglich Fahrzeuge gezielt zum Laden aufzufordern, wenn z. B. die Aufladekosten niedrig sind.



Abbildung 1: Ladestation

Die Open Charge Alliance – ein globales Konsortium von privaten und öffentlichen Einrichtungen, hat dafür OCPP (Open Charge Point Protocol) entwickelt, das die Kommunikation zwischen den Ladestationen und dem zentralen System beschreibt. Die Firma GIR entwickelte dafür einen OCPP-Simulator. Dieser Simulator wurde in JavaScript geschrieben und läuft unter Node.js – ein plattformunabhängiges, sehr schnelles Framework, der die Entwicklung von serverseitigen Webanwendungen ermöglicht. [2]

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

Der Simulator kann sowohl die Ladestationen als auch das zentrale System simulieren. Es ist also möglich beispielsweise ein zentrales System und mehrere Ladestationen mit dem selben Simulator zu instanzieren. Der Kommunikationsprozess zwischen den Ladestationen und dem zentralen System geschieht über den OCPP, der beim Informationsaustausch zwischen den Clients (Ladestationen) und dem Server (zentrales System) für bestimmte Datenstrukturen zuständig ist.

Ziel dieser Abschlussarbeit „Simulationsumgebung für verteilte Ladeinfrastruktur“ ist, den Simulator um Funktionen zu erweitern, damit unter anderem Fehlerzustände von den Ladestationen zu simulieren und erfassen zu können. Zusätzlich soll der Autorisierungsvorgang implementiert werden, der

beim Starten und Stoppen der Ladevorgänge eingreift. Dies ist bewusst so gedacht, damit der Aufladevorgang nicht von einer nicht autorisierten Person unterbrochen werden kann. Da die Bedienung des Simulators aus dem Auslieferungszustand nur über die Konsole möglich ist, soll die Steuerung mit Hilfe eines Webbrowsers ermöglicht werden. Dies soll durch ein Front-End für die Clients und für den Server realisiert werden. Die Clients (Ladestationen) können dann über das Front-End über HTTP mit dem Simulator kommunizieren. Dasselbe Prinzip gilt auch auf der Serverseite (zentrales System). Hier wird jedoch nur die Übersicht über die Ereignisse der Ladestationen angezeigt. Die Simulatoren kommunizieren miteinander nur über OCPP.

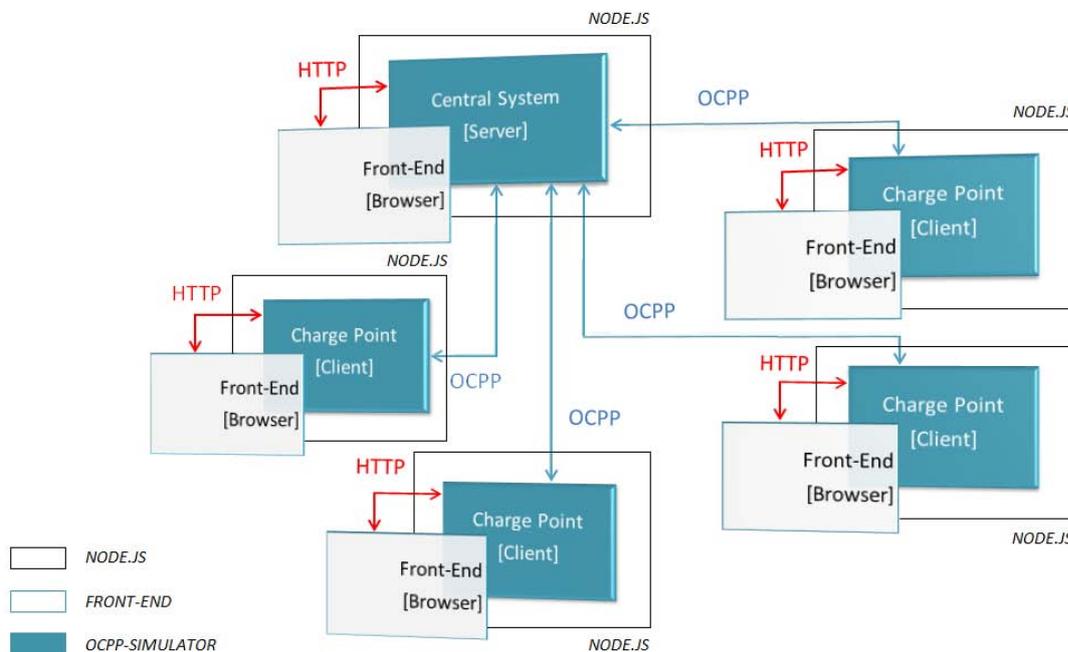


Abbildung 2: Server (zentrales System) und Clients (Ladestationen)

[1] <http://blog.insm.de/2083-autofahren-mit-strom-welche-zukunft-haben-elektroautos> (Stand: 26.11.2013)

[2] Vgl. Roden, Golo (2012): Node.js & Co., S. vii.

Bildquellen:

- Abbildung 1: <http://crome-projekt.de/12/2013>
- Abbildung 2: Eigene Darstellung

Entwurf und Implementierung einer Regelung für ein Gyroskop-stabilisiertes einachsiges Lastentransportsystem einschließlich Optimierung der Regelparameter mit Matlab/Simulink und Erweiterung der hardwarenahen Software des eingebetteten Echtzeitsystems auf Basis eines ARM-Cortex-Prozessors und FreeRTOS

Oliver Breuning*, Reinhard Keller, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Kraftassistenzsysteme gewinnen zunehmend an Bedeutung, nicht zuletzt bedingt durch die Anforderungen einer alternden Gesellschaft. Diese bieten dem Nutzer eine physische Entlastung und verhindern übermäßige körperliche Belastungen und damit Verletzungen, z.B. einen Leistenbruch, oder unterstützen bei chronischen Erkrankungen, z.B. bei einem Bandscheibenvorfall.

Im Rahmen einer Forschungsstudie der Robert Bosch GmbH wurden Unterstützungsfunktionen an einem einachsigen Lastentransportsystem nach dem Segway-Prinzip untersucht (siehe Abbildung 1). Beim Aufnehmen und Absetzen der Last sowie beim Balancieren lässt sich mit diesem System eine erhebliche Unterstützung erreichen [1].

Bei der bislang umgesetzten Funktion erfolgt die Stabilisierung durch ständige Ausgleichsbewegungen der Räder. Dies erscheint für die avisierten Nutzerfunktionen als ungünstig, weswegen Alternativen zur Stabilisierung gesucht werden, zum Beispiel mithilfe von mechanischen Gyroskopen.

Gyroskope finden in einer Vielzahl verschiedener Produkte (Flugzeug, Automobil, Mobiltelefon, etc.) als Drehratensensoren Verwendung, können aber ebenso als Aktoren fungieren.

In Satelliten werden sie beispielsweise zur aktiven Lageregelung verwendet. Die Variation des Drehimpulses des Gyroskops beeinflusst den Drehimpuls des Gesamtsystems erhalten bleibt [2]. Das Phänomen der Drehimpulserhaltung soll auch bei der Stabilisierung des einachsigen Lastentransportsystems ausgenutzt werden.



Abbildung 1: Lastensystem stabilisiert mittels Segway-Prinzip

Abbildung 2 zeigt die schematische Zeichnung der verwendeten Gyroskopbox. Die Kreiselmotoren (in Abb. 2 rot dargestellt) rotieren mit einer konstanten Geschwindigkeit. Durch Ausüben einer Drehbewegung um den Winkel φ mit der Winkelgeschwindigkeit ω wird ein Moment erzeugt, das in Abb. 2 violett dargestellt ist. Es kommen zwei gegenläufige Gyroskope zum Einsatz. Durch die Vektoraddition der Momente m_{G_r} und m_{G_l} wird das Gesamtmoment, in Abb. 2 als m_{GB} dargestellt, erzeugt.

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

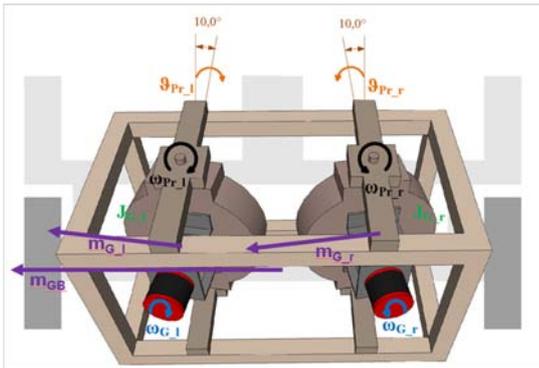


Abbildung 2: Twin-Gyro

Diese Gyroskopbox wird an der Rückseite des Lastentransportsystems befestigt (siehe Abb. 3). Das Gesamtsystem, bestehend aus Gyroskopen, Radmotoren, inertialer Messeinheit, Magnet-Dreh-Encoder zur Messung der Bewegung der Präzessionsmotoren und Dreh-Encoder für die Bewegungserkennung der Räder, wurde aufgebaut und eine Stabilisierung über Gyroskope realisiert.

Die Regelung, die Verarbeitung der Sensorwerte sowie die Ansteuerung der Aktoren werden auf Basis eines ARM-Cortex-M4 Prozessors und des Echtzeit-Betriebssystems FreeRTOS umgesetzt.

Tasks rufen die erstellten Treiber- und Reglerfunktionen zyklisch auf. Die Regelung der Gleichstrommotoren erfolgt über einen Kaskadenregler. Für die Regelung des Gesamtsystems, also für die Regelung der Balancierfunktion, kommt ein Zustandsregler 6. Ordnung zum Einsatz. Als Zustände fungieren der Nickwinkel und die Nickwinkelgeschwindigkeit des Lastentransportsystems, Winkel und Drehrate der Präzessionsmotoren sowie die Position und die Geschwindigkeit des Lastentransportsystems.

Für die Simulation des Systems und die Reglerentwürfe (Kaskadenregelung der Motoren, Zustandsregelung) kommt Matlab/Simulink zur Anwendung. Zudem diente Matlab auch während der Implementierung der einzelnen Hardware-Treiber zur Darstellung und Auswertung von Messwerten der Sensoren. Die grafische Darstellung erlaubte eine Plausibilitätsprüfung der ges-

senen Werte, z.B. auf Sprünge oder auf ein Drift der Messwerte. Mittels der nun geprüften Sensoren konnte dann die Systemidentifikation beispielsweise der einzelnen Motoren durchgeführt werden. Es konnten jetzt beispielsweise die Messwerte der Sensoren verwendet werden, um die Reaktionszeit auf einen Sprung zu ermitteln.

Das in dieser Abschlussarbeit neu umgesetzte Stabilisierungskonzept ließ sich schließlich am realen Aufbau bestätigen.

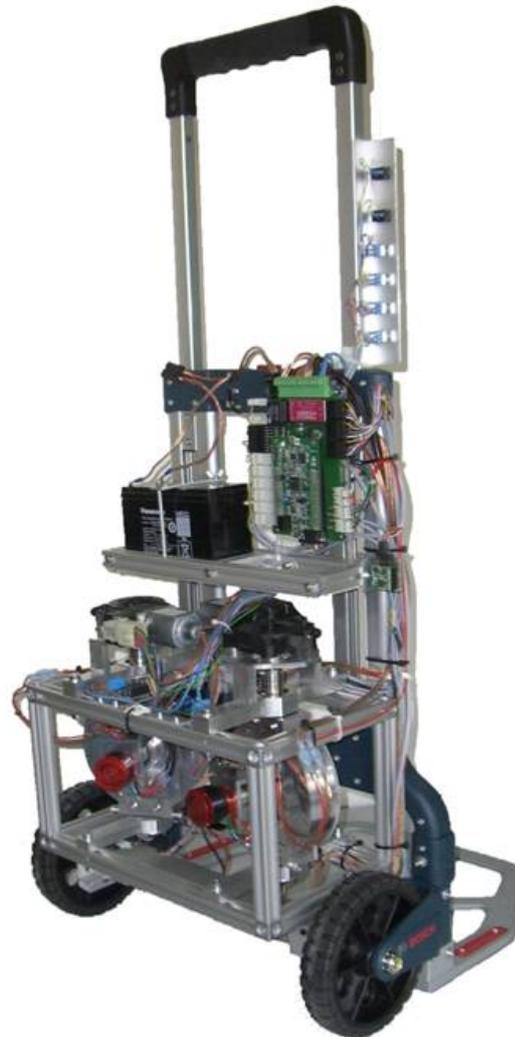


Abbildung 3: Lastensystem mit Gyroskopbox

- [1] Studie Robert Bosch GmbH
- [2] Hering, Martin, Stohrer: Physik für Ingenieure; 7. Auflage Springer Verlag (1999)

Bildquellen:

Abbildung 1-3: Bildarchiv Robert Bosch GmbH

Manöverprädiktion im urbanen Verkehr Datenanalyse und Auswertung mit MATLAB

Kathrin Bromberg*, Reiner Marchthaler, Hermann Kull

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

An einem Zebrastreifen: Ein Fahrzeug nähert sich. Der Fußgänger bleibt stehen. Der Fahrer verringert die Geschwindigkeit des Fahrzeuges. Der Passant überquert die Straße.

Diese Abfolge zeigt, dass der Fußgänger in der Lage ist, die Absicht des Fahrers anhand dessen vorangegangenen Verhaltens einzuschätzen. Hätten die Assistenzsysteme eines Fahrzeugs die Fähigkeit, das Handeln des Fahrers vorzubestimmen, so könnten diese gezielter unterstützen.

So könnte in einer Gefahrensituation schon weitaus früher entschieden werden, ob ein Ausweichmanöver oder eine Gefahrenbremsung unterstützt werden muss. Aber auch Komfortfunktionen, wie der Spurverlassenswarner, welcher derzeit bei einem beabsichtigten Spurwechsel ohne Blinksignal irrtümlich warnt, könnten verbessert werden. Der Spurverlassenswarner wüsste in diesem Fall, dass der Spurwechsel beabsichtigt war und würde nicht warnen.

Um die Assistenzsysteme dahingehend zu erweitern, dass diese auf die Handlung des Fahrers reagieren, muss zunächst die Intention des Fahrers erkannt werden. Daher ist die zentrale Frage dieser Bachelor-Thesis: „Wie kann die Intention eines Fahrers anhand der aktuell verbauten Sensorik ermittelt werden?“ Zu diesem Zweck wurde im Rahmen einer Studie eine Versuchsfahrt mit jeweils 15 Frauen und Männern à zwei Stunden durch Berlin aufgezeichnet.

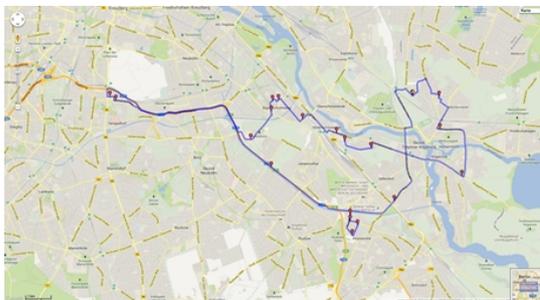


Abbildung 1: Versuchsstrecke durch Berlin

Die aufgezeichneten Daten wurden mittels MATLAB-Skripten in Fahrmanöver unterteilt. Die Parameter von den CAN-Bussen und dem FlexRay wurden auf Auffälligkeiten, die auf ein Manöver schließen lassen, untersucht.

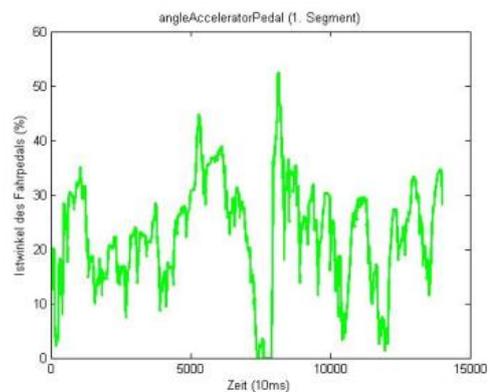


Abbildung 2: Signalbeispiel

Die durch die bereits vorhandenen Sensorik aufgezeichneten Fahrzeugdaten haben in bestimmten Situationen Muster, so dass eine Detektion eines Manövers vor ihrer Ausführung möglich ist.

Zusätzlich wurden die Blicke der Probanden mit einem Eye Tracker aufgezeichnet. Auch diese Daten wurden anhand von MATLAB-Skripten auf Verhaltensmuster untersucht.

Derzeit bietet die Erfassung der Blickrichtung jedoch noch einige Herausforderungen. So kann zum Beispiel, bei einem Schulterblick, die Pupille des von der Kamera abgewandten Auges nicht erfasst werden. Dies führt teilweise zum Abreißen des Blickvektors, aber auch zu falschen Blickvektoren. Ein möglicher Lösungsansatz ist es, den Blickvektor mit dem Kopfvektor zu plausibilisieren.

Die Studie fand im Rahmen des öffentlich geförderten Projektes UR:BAN statt.

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

[1] www.urban-online.org 12/2013

Bildquellen:

- Abbildung 1: selbst erstellt mit Google Maps
- Abbildung 2: selbst erstellt mit MATLAB

3D-Modellierung eines Stadtteils mit 3ds Max

Philippos Dentsas*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die computergestützte Umsetzung und Darstellung von dreidimensionalen Szenen wird heutzutage immer wichtiger. Das Ziel ist dabei die Visualisierung und Abbildung der Realität in einem durch Computergrafik generierten Modell. Die Modellierung von 3D-Gebäuden findet heutzutage in vielen Bereichen Anwendung: in der Architektur zur Stadtplanung, in der Filmindustrie und in Computerspielen.

Grundsätzlich fängt man bei der Modellierung einer komplexen dreidimensionalen Szene mit Grundkörpern (Primitive) an. Zu den Grundkörpern gehören z.B. Würfel, Quader, Zylinder, Kegel und Kugeln. Diese dienen als grobe Objekte und als Beginn für das weitere Modellieren und bestehen aus mehreren Unterelementen: Vertex (Plural: Vertices), Kante (engl.: edge) und Polygon.

Vertices bilden die Eckpunkte von Grundkörpern, deren Position im Koordinatensystem über einen Vektor festgelegt wird [1]. Kanten verbinden mehrere Vertices über eine Verbindungslinie miteinander. Polygone entstehen, wenn durch einen Linienzug (Polygonzug) Vertices durch Kanten miteinander verbunden werden und dabei eine Fläche umschließen [2].

Aus den Primitiven lassen sich durch Manipulation der Unterelemente komplexere Körper erstellen. So können aus einfachen Grundkörpern stufenweise vielfältige Formen entstehen.

Die verschiedenen Methoden umfassen unter anderem das Extrudieren, zusammengesetzte Objekte und Drehverfahren. Beim Extrudieren werden einzelne Polygone von der Oberfläche eines Modells herausgehoben. Zusammengesetzte Objekte sind meist komplexer und entstehen aus der Kombination mehrerer einfacher Grundobjekte. So können z.B. Kugelformen in Quaderobjekte hineingeschnitten werden (Abbildung 2).

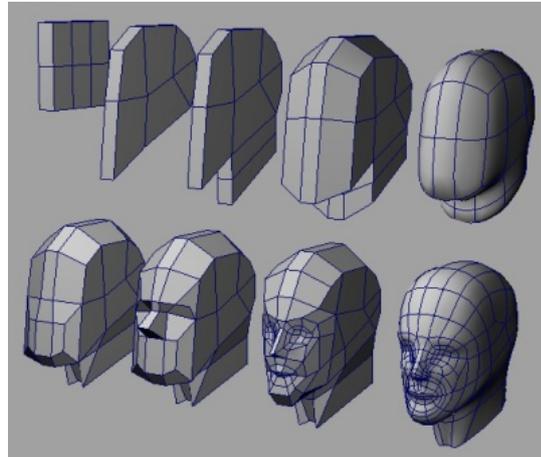


Abbildung 1: Modellierung komplexer Formen aus einfachen Grundkörpern

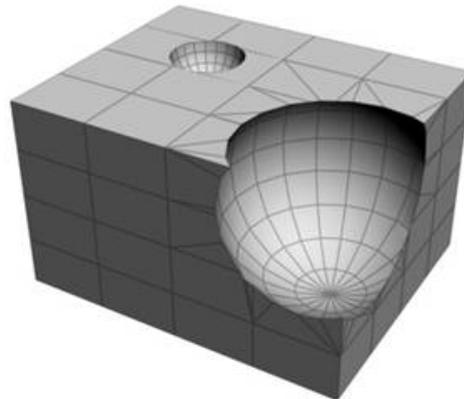


Abbildung 2: Ein zusammengesetztes Objekt, entstanden aus einer Operation mit mehreren Primitiven

Drehverfahren bieten eine sehr effiziente Möglichkeit zur Erstellung von Formen, die sonst umständlich zu modellieren wären: dabei wird eine kurvenförmige Linie um eine zentrale Achse gedreht, wobei (wie bei einer Drehmaschine) ein Objekt entsteht und geformt wird. Beispiele hierfür sind Vasen, Glocken oder Weingläser (Abbildung 3).

*Diese Arbeit wurde durchgeführt an der Fakultät Informationstechnik

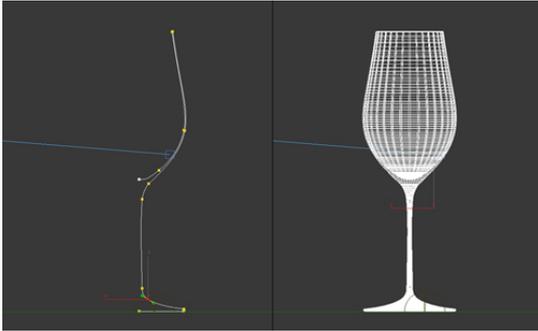


Abbildung 3: Objekte wie Weingläser entstehen mit Drehverfahren

In dieser Abschlussarbeit wurde eine 3D-Szene eines historischen Stadtteils von Esslingen erstellt. Der gewählte Stadtteil ist die Esslinger Burg. Die modellierten Teile dieser mittelalterlichen Befestigung umfassen den Dicken Turm, die Hochwacht, den Seilergang und die Burgstaffel, die mit ihrer steilen Treppe hinauf zur Hochwacht führt. Die Gebäude wurden mit 3ds Max erstellt, einem 3D-Computergrafikprogramm der Firma Autodesk, das in verschiedenen Branchen professionell eingesetzt wird für Modellierungen, Animationen, Rendering und Compositing [3].

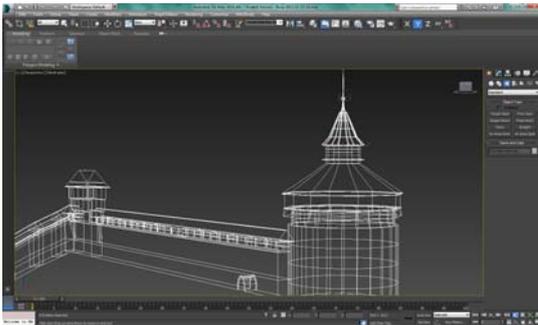


Abbildung 4: Modellierung in 3ds Max

Je nach Endanwendung ist eine unterschiedlich genaue und feine Modellierung notwendig. Für dieses Modell wurde auf eine realitätsnahe und detailreiche Form der einzelnen Teile Wert gelegt, um dem Original an Ähnlichkeit möglichst nahe zu kommen. Dazu wurden mehrere Bereiche der Burganlage von verschiedenen Blickwinkeln fotografiert. Die Fotos wurden herangezogen und ausgewertet, um wichtige Informationen zur Lage und zu den Maßen der Gebäudeteile zu bekommen.

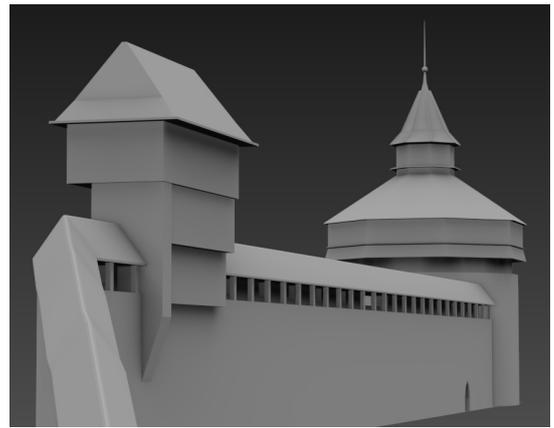


Abbildung 5: Gerenderte 3D-Szene der Esslinger Burg

In früheren Studien- und Abschlussarbeiten wurden das Hochschulgebäude in der Flandernstraße und eine Reihe von anderen Gebäuden und Plätzen der Esslinger Innenstadt bereits für ein virtuelles Esslingen modelliert. Dieser Sammlung von Stadtteilen kommt nun auch die Esslinger Burg hinzu.

- [1] <http://de.wikipedia.org/wiki/Vertex> 12/2013
 [2] <http://de.wikipedia.org/wiki/Polygon> 12/2013
 [3] http://de.wikipedia.org/wiki/3ds_Max 12/2013

Bildquellen:

- Abbildung 1: http://sillord.free.fr/wip/tutotopology/textrusion_fichiers/boxtohead.jpg 12/2013
- Abbildung 2: http://www.tutorials3d.com/imagestut/tut_free3_objcomp_clip_image047.jpg 12/2013
- Abbildung 3: http://media.tumblr.com/tumblr_lu6xplyXt01r1n4g2.jpg 12/2013
- Abbildung 4-5: Eigene Darstellung

Interaktive Visualisierung eines komplexen Graphen

Alexander Diel*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die Informationsflut in der heutigen Zeit ist kaum zu bändigen. Um diese Flut in eine Richtung zu lenken, benötigt man Hilfsmittel wie Graphen. Diese werden z.B. genutzt um Stammbäume, Bahnnetze oder Beziehungsgeflechte darzustellen.



Abbildung 1: S-Bahn Liniennetz Stuttgart

Der Plan eines Bahnnetzes (wie Abbildung 1) wird täglich betrachtet und hilft Fahrgästen den Weg nach Hause oder zur Arbeit zu finden. Viele Menschen wissen jedoch nicht, dass es sich bei der Abbildung um einen Graph handelt. Das Liniennetz wird als Graph dargestellt, um die geographischen Begebenheiten zu vereinfachen. Die Fahrgäste müssen sich keine Gedanken darüber machen wie groß die Entfernungen zwischen den Haltestellen sind. Sie planen ihre Fahrtrouten nur noch in der Anzahl der Haltestellen oder der benötigten Fahrzeit. Das Wichtigste hierbei ist, dass den Menschen genau die Information vermittelt wird, die sie benötigen. Diese Eigenschaft eines Graphen lässt sich auf beliebige Daten abbilden.

Im Rahmen dieser Arbeit wurde ein Beziehungsgeflecht komplexer Daten visualisiert. Diese Beziehungen werden bisher in Form von Listen dargestellt, d.h. um die Beziehung eines Datensatzes zwischen anderen festzustellen, müssen die Listen durchsucht und die passenden Verbindungen herausgearbeitet werden. Diese Arbeit ist zeitraubend und fehleranfällig. Mit Hilfe eines Graphen können die Informationen detailliert und übersichtlich

dargestellt werden und die Beziehungen sind auf einen Blick erkennbar.

Die Herausforderung hierbei besteht in der Behandlung sehr großer Datenmengen. Der Firma Cellent Finance Solutions AG liegen Datenmengen in einem siebenstelligen Zahlenbereich vor. Selbst mit einem Graphen stößt man hier an die Grenzen der Übersichtlichkeit.

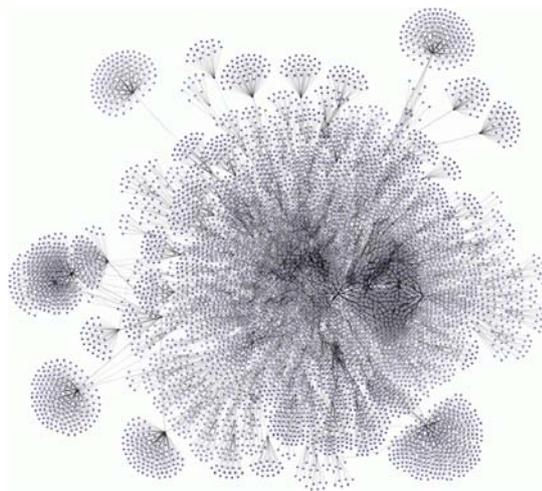


Abbildung 2: Graph mit 10 000 Knoten

Um dieses Problem zu umgehen konnte auf den Girvan–Newman–Algorithmus zurückgegriffen werden, der den Graph in Cluster unterteilt. Der Algorithmus wurde speziell dafür entwickelt um Beziehungen in einem sozialen Netzwerk zu analysieren [1]. Die Cluster konnten in Gruppenknoten gekapselt werden und verringern dadurch den Rechenaufwand bei der Darstellung des Graphen. Durch Interaktion kann sich der Benutzer auf dem Graphen bewegen und detaillierte Informationen abfragen. Über einen Gruppenknoten erhält man Zugriff auf den darin gespeicherten Abschnitt des Graphen. Wenn genauere Informationen zu den direkten Beziehungen eines Knotens benötigt werden, kann dies ebenfalls dargestellt werden.

*Diese Arbeit wurde durchgeführt bei der Firma Cellent Finance Solutions AG, Stuttgart

[1] Girvan M. and Newman M. E. J., Community structure in social and biological networks

Bildquellen:

- Abbildung 1: <http://www.s-bahn-region-stuttgart.de> (12/2013)
- Abbildung 2: <http://www.yworks.com> (12/2013)

Erstellung einer Augmented Reality App zur Visualisierung von Windkraftanlagen, für bessere Bürgerbeteiligung mit neuen Technologien.

Kristian Dürsch*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

In Folge der Änderung des Landesplanungsgesetzes und der Veröffentlichung des Winderlasses stehen die Kommunen in der Region Stuttgart vor der Herausforderung, geeignete Standorte für Windkraftanlagen festzulegen und durch eine gesetzeskonforme Planung eine verbindliche Grundlage für die Genehmigung bzw. Ablehnung von Bauanträgen zu schaffen. Bei der Entscheidung für potenzielle Standorte von Windrädern sollte zudem die Bevölkerung in einem möglichst transparenten Verfahren eingebunden werden. Dazu können u.a. auch sachlich belastbare Visualisierungslösungen wertvolle Beiträge leisten.

Bestehende Visualisierungsansätze weisen teilweise verschiedene gravierende Schwachpunkte auf. Etwa erlauben z.B. Visualisierungen auf Basis von Fotomontagen keine „Rundumsicht“ aus unterschiedlichen Perspektiven; Fotomontagen haben z.T. gravierende Qualitätsmängel aufgrund optischer Verzerrungen. Deshalb sind diese insbesondere für die Beteiligung der Bevölkerung an der Entscheidungsfindung nur sehr bedingt geeignet. Daher könnten z.B. auch dreidimensionale Visualisierungslösungen geeignet sein, die unter Einbezug von Animationselementen eine individuellere Sicht auf Windräder ermöglichen, ohne dass dafür eine komplexe Virtual Reality Umgebung benötigt wird.

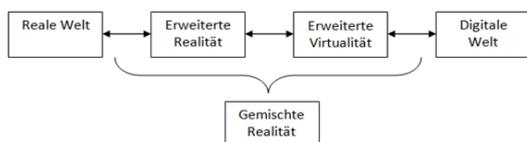


Abbildung 1: Zusammenhänge zwischen realer und digitaler Welt und deren Zwischenstadien.

Das vorgeschlagene Projekt soll nun Möglichkeiten solcher Visualisierungen mit Blick auf eine verbesserte Einschätzung der opti-

schen Auswirkungen von Windkraftprojekten möglichst transparent aufzeigen und beispielhaft anhand einiger Referenzmodelle darstellen. Die fertige App soll dazu beitragen, die Entscheidungsfindung bei der Bestimmung von Standorten von Windkraftanlagen und ihre Nachvollziehbarkeit zu unterstützen. Den Trägern der kommunalen Planung soll dafür zu transparenten Bedingungen ein verlässliches Instrument an die Hand gegeben werden, mit dem die Bevölkerung besser in die Entscheidungsfindung eingebunden werden kann. Mit Hilfe dieses Projekts werden folgende Ergebnisse angestrebt:

–Die fertige App erlaubt durch ihre vergleichsweise niedrigen technischen Voraussetzungen beim Endanwender (mobiles Android Gerät mit GPS-Empfänger und Kamera) eine einfache Nutzung und ermöglicht so eine breite Bürgerbeteiligung.

–Die angebotene Visualisierungslösung vermittelt der lokalen Bevölkerung einen auf den jeweils individuellen Blickwinkel (eigener Standort des Betrachters) bezogenen Eindruck der geplanten Baumaßnahmen und ermöglicht so eine bessere Urteilsbildung und Entscheidungsfindung.



Abbildung 2: Mockup der Windanlagen-Visualisierungs App (WAV-App) aus Sicht eines Anwenders

–Das VDC-Fellbach bietet Kommunen sowie Interessengemeinschaften für oder gegen Windkraftanlagen, das Erstellen von Visualisierungen von Windkraftanlagen als

*Diese Arbeit wurde durchgeführt bei der Firma Virtual Dimension Center, Fellbach

Dienstleistung an.

Durch die steigende Rechenleistung von mobilen Geräten hat das Thema Augmented Reality (AR) das Potenzial allgegenwärtiger zu werden und dadurch neue Einsatzmöglichkeiten zu schaffen. Dieses Potenzial wird im Hinblick auf die Verbesserung der Bürgerbeteiligung getestet. Die kombinierte Visualisierung einer virtuellen Welt mit einer realen Welt bezeichnet man als Augmented Reality. Dabei werden gespeicherte reale Daten mit berechneten virtuellen Daten gemischt. Es erfolgt quasi eine Überlagerung des Bildes der realen Welt mit dem Bild der virtuellen Welt. Der Anwender nutzt beide Datensätze als Information.[1]

Um die visuellen Auswirkungen von möglichen Windkraftanlagen darzustellen, soll in diesem Projekt eine auf Android-Geräten laufende AR-App (Eine Anwendung die mit Einsatz von erweiterter Realität arbeitet) entwickelt werden. Diese wird die Position des mobilen Gerätes des Nutzers über das Globale Positionssystem (GPS) ermitteln und in ein von der Kamera des Gerätes aufgenommenes Bild der realen Welt, das animierte Model von Windrädern einfügen (rendern). Herausforderungen dabei werden sein, das Windrad in der richtigen Größe, im Verhältnis zum Abstand darzustellen. Und die Positionierung der Windräder anhand der Attitude in der richtigen Höhe. dies wird durch Ungenauigkeiten des GPS erschwert. Hierfür werden im Zuge der Arbeit verschiedene Methoden der Höhenbestimmung getestet und implementiert.

Eine Sichtbarkeitsanalyse, welche überprüft, ob von der aktuellen Position das Windrad zu sehen ist, oder ob es durch Hügel oder Berge verdeckt ist wird ebenfalls implementiert und sorgt dafür, dass nicht sichtbare Windräder ausgeblendet werden. Hierbei wird nur die Landschaftsform betrachtet. Bebauung und Bewaldung wird hierbei voraussichtlich nicht beachtet werden können.

Um dem Nutzer eine einfache Orientierungsmöglichkeit zu gewähren wird in einer Übersichtskarte die aktuelle Position angezeigt. Auf Basis der Satellitenbilder von Google Maps kann so ein Bezug zur Umgebung gefunden werden. Über die ebenfalls in der Karte dargestellten Windradstandorte kann die Entfernung

und die Positionierung der Windräder zueinander betrachtet werden.



Abbildung 3: Die Kartenansicht in welcher der Nutzer die genaue Position von sich und den umliegenden Windkraftanlagen sehen kann.

Des Weiteren soll eine einfache Möglichkeit gegeben werden, die um zu visualisierende Windräder für die App verfügbar zu machen. Dies beinhaltet das Eingeben der Position der Windräder sowie einiger Parameter wie der Höhe, Ausrichtung und Rotordurchmesser. Hierfür könnte eine Erweiterung der WAV-App oder ein webbasiertes Generator in Frage kommen. Die Verbreitung folgt über eine in der App eingebaute Downloadfunktion, welche die lokal gespeicherten Standorte auf Aktualität überprüft und neu erstellte oder überarbeitete Versionen vom Server abrufen.

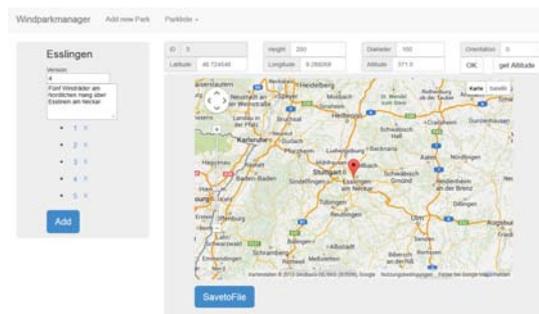


Abbildung 4: Webbasiertes Windparkmanager über welchen neue Parks angelegt und bearbeitet werden können.

[1] Virtuelle Realität, Prof. Dr. Reinhard Schmidt, Esslingen, im September 2004

Entwicklung eines Programms zur Durchführung von Lasttests aus der Cloud

Kai Dybionka*, Dominik Schoop, Kai Warendorf

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

In der heutigen Zeit lässt sich das Internet und die darin gespeicherten Daten nicht mehr wegdenken. Jeder Mensch ist online und im World Wide Web unterwegs. Viele nutzen ihr Smartphone, um auf Webseiten von Internet-Suchmaschinen nach Informationen zu suchen, oder sich in sozialen Netzwerken zu verständigen. Doch wieviele Nutzer können gleichzeitig solche Seiten benutzen, bevor diese nicht mehr erreichbar sind? Ab wann bricht ein Internet-Forum zusammen, wenn jeder User es zur gleichen Zeit aufruft?

Um solche Zahlen zu erkennen, benötigt der Betreiber einer Webseite Stresstests. Bei dieser Art der Prüfung werden Internetbenutzer auf einer Seite durch Prozesse auf einem Rechner ersetzt. Die Prozesse simulieren den menschlichen Anwender durch verschiedene Testszenarien und Analysieren das Verhalten der untersuchten Seite. Die Internetseite wird einem sogenannten Lasttest ausgesetzt. Durch das Ergebnis dieser Belastungsprobe wird die Maximallast auf eine Seite deutlich. Diese Last gibt dem Betreiber dieser Webseite die Möglichkeit die Maximalbenutzer für seine Seite zu erkennen und sich darauf einzustellen.

Diese Abschlussarbeit beschäftigt sich mit einem solchen Lasttest. Im Speziellen wird darauf Wert gelegt, dass sich dieser Test automatisch aufbaut und er ohne weitere Hilfe selbstständig abläuft. Dabei wird davon ausgegangen, dass man verschiedene Hosts besitzt, die aber außer dem Betriebssystem keine spezielle Software vorinstalliert haben müssen. Das entwickelnde Programm soll sich selbst einrichten, um einen Lasttest auf eine gewünschte Webseite zu starten. Außerdem soll es einfach und intuitiv bedienbar sein, und

die Test müssen sich wiederholen und modifizieren lassen.

Um eine variable Anzahl an Testbenutzern zu generieren, werden die Systeme aus einer Cloud heraus gestartet, da es in dieser IT-Infrastruktur möglich ist, sich dynamisch Rechenkapazität zu mieten. Ein weiterer Vorteil der Cloud sind die verteilten Rechner, das heißt, der Test wird nicht von einem Host ausgeführt, sondern von vielen. Anhand dieser Eigenschaft kann die Belastung auf eine Homepage variabel und an jede Seite angepasst werden.

Die Herausforderung der vorliegenden Bachelorarbeit ist, ein solches Programm zu entwerfen und zu realisieren. Als zugrundeliegende Software wurde *tsung*, ein „open-source multi-protocol distributed load testing tool“ [1], gewählt. Mit dieser Software ist es möglich Tausende virtuelle Benutzer gleichzeitig zu simulieren. Außerdem ist es, durch die große Anzahl der unterstützten Protokolle, für verschiedene Szenarien ausgeprägt. Da *tsung* ein Kommandozeilen-Tool ist, wurde zur einfacheren Bedienung eine PHP-Umgebung programmiert, durch die *tsung* bedient werden kann, und im Anschluss die Ergebnisse eines durchgeführten Tests ausgibt. Als Cloud-Dienst wurde die Amazon Elastic Compute Cloud (EC2) gewählt. Mit dieser Cloud ist es möglich, innerhalb weniger Minuten Kapazitäten auf den Hosts zu erweitern oder zu verringern, und man ist in der Lage, mehrere Instanzen gleichzeitig in Betrieb zu nehmen. Zusätzlich kann man sich einen festgelegten IP-Bereich selbst wählen und sich somit verschiedene Hosts aus verschiedenen Kontinenten simulieren. [2]

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

[1] <http://tsung.erlang-projects.org> 12/2013
[2] <http://aws.amazon.com/de/ec2> 12/2013

Verschlüsselung mit elliptischen Kurven im Embedded Bereich

Daniel Eisele*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Diverse Datenschutzskandale der letzten Jahre sowie die jüngsten Enthüllungen von Edward Snowden über die Überwachungsaktivitäten der NSA haben Themen wie Datenschutz, Schutz der Privatsphäre, den eigenen Umgang mit persönlichen Daten, die Vertraulichkeit und Geheimhaltung von privaten, aber auch von geschäftlichen Daten und Betriebsgeheimnissen in den Mittelpunkt der Medienberichterstattung sowie zum Gegenstand von Diskussionen in Familien, Betrieben, an Hochschulen und zu vielen weiteren Gelegenheiten gemacht. Dabei geht es auch immer wieder um Fragen, wie Nachrichten und Daten bei der Übertragung und Speicherung vor unautorisierten Zugriffen, Manipulationen oder Diebstählen geschützt werden können. Hierbei kommt die Kryptologie zum Einsatz, welches die Wissenschaft ist, die sich mit der Ver- und Entschlüsselung von Nachrichten beschäftigt. Dabei hat die Menschheit schon früh begonnen, vertrauenswürdige Nachrichten geheim zu halten. Das erste bekannte Kryptosystem namens „Atbash“ wurde 600 J.v.C. von den Juden eingeführt. Heutzutage findet die Kryptologie ihren größten Einsatz in der Computertechnik. Ein neues Gebiet der Kryptologie ist die Verschlüsselung und Entschlüsselung mit elliptischen Kurven (eng. Elliptic Curve Cryptography: ECC). Die erste Verschlüsselungstechnik mit ECC wurde 1985 von Victor S. Miller und Neal Koblitz eingeführt.

Diese Arbeit hat die Analyse und Darstellung von ECC und deren spezifische Anwendungsmöglichkeit in Embedded Systemen zum Gegenstand. ECC gehört zu den asymmetrischen Algorithmen. Die Sicherheit von ECC beruht auf dem Problem des berechnen eines diskreten Logarithmus. Da das diskrete Logarithmus Problem in elliptischen Körpern schwieriger zu lösen ist, als in anderen zyklischen Gruppen, kann bei ECC für das gleiche Sicherheitsniveau eine kleinere Bit-Zahl als Schlüssel verwendet werden. Dieses Ersparnis an Bits und damit die Einsparung an Speicherplatz und Rechenleistung ist für den Embedded Bereich von großer Bedeutung, wodurch ECC trotz komplizierter Implementierung ein geeignetes Verfahren darstellt und

daher häufig eingesetzt wird.

Eine elliptische Kurve für ECC hat die Form $E : y^2 \equiv x^3 + ax + b \pmod{p}$. Außerdem muss die Bedingung $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ erfüllt sein. Zudem wird ein Punkt \mathcal{O} im Unendlichen eingeführt, der das neutrale Element ist. Durch diese Voraussetzungen lässt sich ein Punkt auf der elliptischen Kurve durch einen anderen Punkt oder durch zwei andere Punkte auf der elliptischen Kurve berechnen. In Abbildung 1 wird eine elliptische Kurve über den reellen Zahlen mit einer Punktaddition $P+Q$ veranschaulicht.

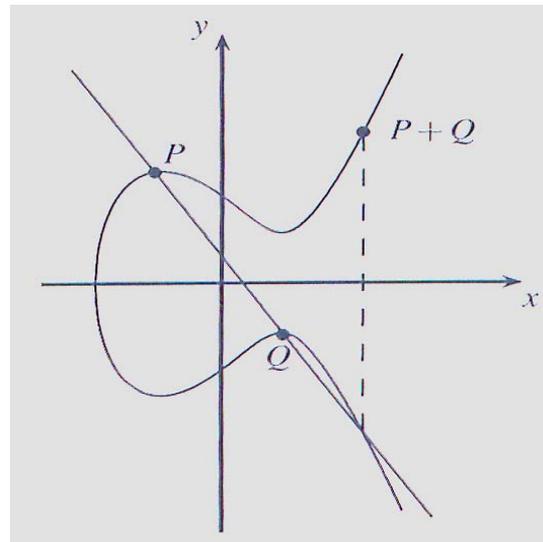


Abbildung 1: Elliptische Kurve über den reellen Zahlen

Man erkennt, dass die Gerade durch die Punkte P und Q die elliptische Kurve in dem dritten Punkt schneidet, der gespiegelt an der x -Achse $P+Q$ ergibt. Bei einem Punkt $P=Q$ würde man die Tangente durch P betrachten. Diese Tangente würde die Kurve ebenfalls in einem dritten Punkt schneiden, so dass gilt $P+P=2P$. Wenn diese Punktaddition öfters durchgeführt wird, erhält man ein Punkt $\sum_{j=1}^i P_j = i \cdot P$. Die Anzahl der Punktadditionen i muss geheim gehalten werden, denn sie ist der private Schlüssel. Die Sicherheit beruht nun darauf, dass man für die Berechnung von i einen diskreten Logarithmus lösen muss, auch wenn alle anderen Parameter bekannt sind.

*Diese Arbeit wurde durchgeführt an der Fakultät Informationstechnik

Da das Lösen eines solchen diskreten Logarithmus rechnen- und zeitaufwändig ist, hat man durch i einen privaten Schlüssel generiert.

Ein Einsatzgebiet von elliptischen Kurven ist, der sichere Austausch von symmetrischen Schlüsseln. Dafür lässt sich der Diffie-Hellman Schlüsselaustausch mit ECC (ECDH) realisieren. Hierzu wählt man in einer ersten Phase eine elliptische Kurve, nach den obigen Bestimmungen, und einen Startpunkt Q , mit dem jeder weitere Punkt auf der elliptischen Kurve berechnet werden kann. Es gibt Standardisierungen für elliptische Kurven mit Startpunkte die verwendet werden sollten, da diese elliptischen Kurven einen sicheren Einsatz in der Kryptologie gewährleisten. In der zweiten Phase wird das eigentliche Protokoll durchlaufen, siehe dazu Abbildung 2. Als erstes wählt Alice ihren privaten Schlüssel k_{prA} . Bob macht dasselbe und wählt seinen privaten Schlüssel k_{prB} . Die privaten Schlüssel k_{prA} und k_{prB} sind Zufallszahlen und müssen aus der Menge $\{2, 3, \dots, \#E - 1\}$ sein, wobei $\#E$ die Anzahl der Punkte auf der elliptischen Kurve ist. Im zweiten Schritt wird der öffentliche Schlüssel von Alice und von Bob berechnet. Dafür addieren Alice und Bob k_{prA} bzw. k_{prB} Punkte auf den Startpunkt Q . Alice landet bei dem Punkt K_{pubA} und Bob landet bei dem Punkt K_{pubB} . K_{pubA} bzw. K_{pubB} ist der öffentliche Schlüssel von Alice bzw. Bob. Als nächstes sendet Alice ihren öffentlichen Schlüssel an Bob und Bob sendet seinen öffentlichen Schlüssel an Alice. Nun findet erneut eine Punktaddition auf der elliptischen Kurve statt. Diesmal addiert Alice k_{prA} Punkte auf Bobs öffentlichen Schlüssel (K_{pubB}) und landet bei dem Punkt K_{AB} . Bob landet bei dem gleichen Punkt K_{AB} wenn er k_{prB} Punkte

auf Alice öffentlichen Schlüssel (K_{pubA}) addiert. Dadurch ist der Punkt K_{AB} der symmetrische Schlüssel den nur Alice und Bob kennen [1].

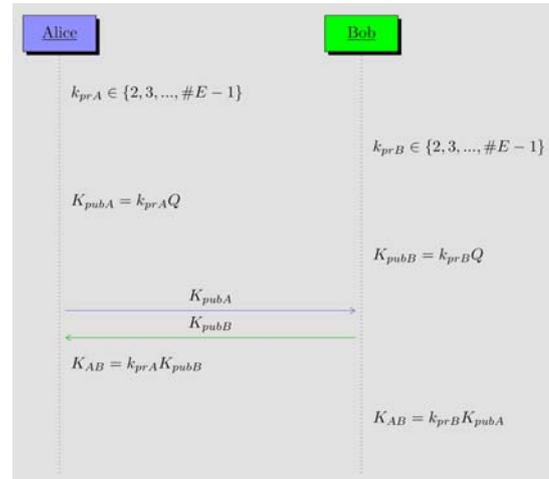


Abbildung 2: Sequenzdiagramm ECDH

Der Schlüsselaustausch sowie die digitalen Signatur, die sich ebenfalls mit ECC realisieren lässt, könnte in der Vehicle-to-Vehicle (V2V) und Vehicle-to-Infrastructure (V2I) Kommunikation eingesetzt werden. Hierbei handelt es sich um eine Kommunikation von einem Fahrzeug mit einem anderen Fahrzeug (V2V) bzw. von einem Fahrzeug mit einer Schnittstelle z.B. einer Ampel (V2I). Dabei muss bei V2V- und bei V2I- Kommunikationen die Abhörsicherheit und die Authentifizierung der Kommunikationspartner gewährleistet sein. ECC stellt auf Grund seiner Eigenschaften insbesondere der, des geringen Bedarf an Rechenleistung und Speicherkapazität ein Ideales Verfahren für diese und ähnliche Einsatzgebiete da [2].

- [1] Christof Paar, Jan Pelzl: Understanding Cryptography: Springer Verlag 1.Auflage 2010: ISBN 978-3-642-04100-6
- [2] Marko Wolf: Security Engineering for Vehicular IT Systems: Vieweg+Teubner Verlag 1.Auflage 2009: ISBN 978-3-8348-0795-3

Bildquellen:

- Abbildung 1: [1]
- Abbildung 2: Selbst erstelltes Bild

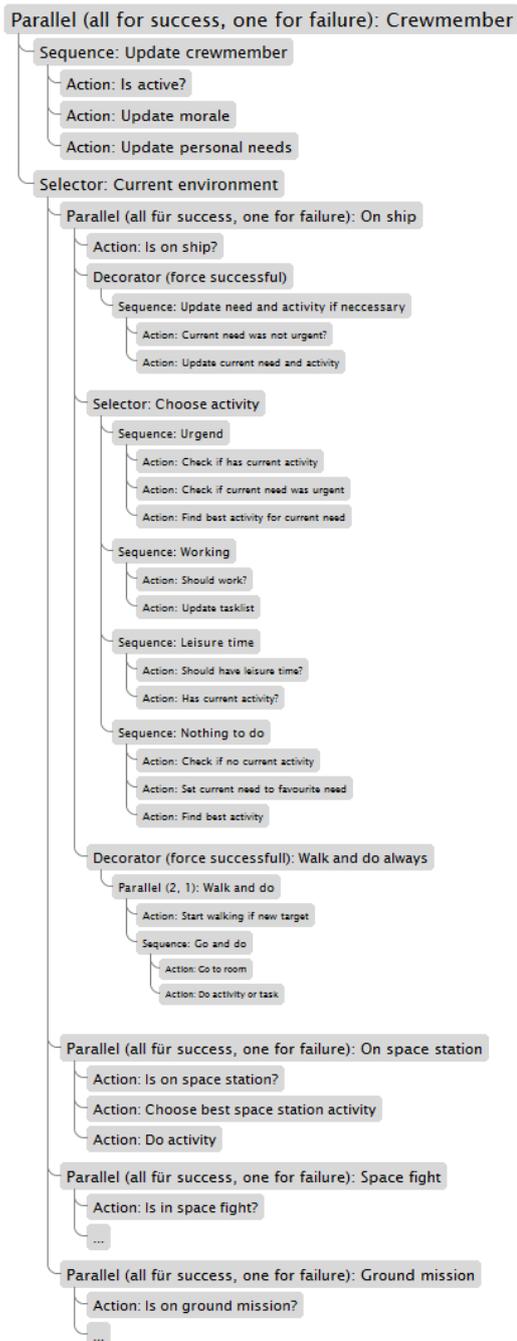


Figure 2: Part of the behavior tree for crew members in Cosmonautica

There is a limited number of node types available for behavior trees. The most basic of them are Action, Sequence, Selector, Parallel, and Decorator. Only the action nodes modify the program's object. Action nodes cannot have any child nodes. All other nodes determine whether actions are carried out or not. Sequence nodes evaluate their child nodes until one returns something else than successful. Selectors do the opposite: They evaluate their child nodes until one returns something else than failure. Both sequence and selector nodes remember at which child the evaluation stopped. The next time they are evaluating they start with the remembered child node. Parallel nodes evaluate their child nodes until a pre-defined number of failures or successes is reached. Decorator nodes have a single child node. They modify how often this child node is evaluated or they modify its return value.

Now, behavior trees are used in "Cosmonautica" to control the behavior of crewmembers in the player's space ship. They determine e.g. in which situations the crew members work or when a need has to be satisfied. Utility functions are used to compare all possible actions to decide which work task or activity needs to be done.

This combination of behavior trees and utility functions has proven to be an easy and powerful way to control the behavior of artificial intelligences. Due to its qualities, it can replace most finite state machines in object-oriented programs.

Key words: Artificial intelligence, Behavior, Behavior tree, Utility function, Finite state machine, Fuzzy logic, Goal-oriented behavior

- [1] Chamandard, A. J. (2007, 9 6). Understanding Behavior Trees. <http://aigamedev.com/open/article/bt-overview>
- [2] Millington, I., & Funge, J. (2009). Artificial Intelligence for Games, 2nd Edition. Boca Raton: CRC Press

Figure sources:

- Figure 1: http://files.aigamedev.com/insiders/BehaviorTrees_Slides.ppt (Slide 41)
- Figure 2: Chasing Carrots KG

Evaluation verschiedener Webtechnologien und prototypische Portierung des Konzepts zur Bildung von Oberflächenelementen

Dawinja Falter*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Wenn man in letzter Zeit den Markt für PCs, Tablets und Smartphones beobachtet hat, wird ein starker Trend hin zu mobilen Geräten sichtbar [1]. Daneben wird der Ausbau des schnellen, mobilen Internets immer weiter vorangetrieben [2]. Somit ist es wenig verwunderlich, dass Kunden vermehrt Webvisualisierungen fordern, sodass sie die Möglichkeit haben ihre Industrieanlagen nicht nur direkt an dem HMI-Display zu überwachen bzw. steuern, sondern eben dies von überall aus tun können. An diesem Punkt setzt die Thesis an, um einen Prototyp einer HMI-Webvisualisierung (WebVisu) zu erstellen. Dabei ist der Fokus auf der Frage, mit welcher Technologie SÜTRON Oberflächenelemente bzw. kurz DSO für Decorated Screen Objects am Besten dargestellt werden können, ein DSO kann z. B. eine Schaltfläche darstellen. Es spielt in diesem speziellen Fall nicht nur die

Technologie selbst eine Rolle. Nach Möglichkeit sollte das bestehende DSO-Konzept zur Erstellung der DSOs genutzt werden, sodass das Programm TSvisIT auf den HMI-Displays und die WebVisu im Browser auf dieselbe Art und Weise funktionieren. Somit orientiert sich der Prototyp an der bereits bestehenden TSvisIT DSO-Implementierung in C# und GDI, der Prototyp wird in JavaScript implementiert werden.

Im ersten Schritt wurde dafür nach verfügbaren Technologien recherchiert. Folgende können für das Anzeigen von Oberflächenelementen genutzt werden:

- Scalable Vector Graphics (SVG)
- HTML5 Canvas-Element
 - 2D-Kontext (2D-Canvas)
 - WebGL-Kontext (WebGL)
- Silverlight



Abbildung 1: Testbild des 2D-Canvas Prototyps

* Diese Arbeit wurde durchgeführt bei der Firma SÜTRON electronic GmbH, Filderstadt



Abbildung 2: Testbild des SVG Prototyps

Die Erstellung der Prototypen beschränkt sich neben der grafischen Darstellung der DSOs auf die Interaktion mit dem Benutzer, sprich Maus bzw. Touch-Eingaben. Somit sollte es mit dem fertigen Prototyp möglich sein ein Bild anzuzeigen, und über eine Schaltfläche einen Bildwechsel anzustoßen. Das primäre Ziel der WebVisu ist, dass sie auf möglichst vielen Geräten und Plattformen lauffähig ist. Darunter fallen PCs, Smartphones, Tablets sowie die neueren HMI-Displays selbst. Nicht zu vergessen ist dabei der Fortschritt der Implementierung der Technologien in den marktführenden Browsern. Ein weiteres Kriterium ist die Zukunftssicherheit der Technologie. Während der Untersuchung auf diese Auswahlkriterien stellte sich heraus, dass die Unterstützung von Silverlight auf mobilen Geräten nicht gegeben ist. Auch bei WebGL ist die Unterstützung noch nicht auf allen Geräten über die Grafikkarte oder den Browser gegeben. Daneben muss für die Anzeige von Text in WebGL ein Umweg über den 2D-Kontext eines HTML5 Canvas-Elements gegangen werden, was bei vielen Textelementen in einer WebVisu die Performance stark beeinträchtigen kann. Daher wurden diese beiden Technologien ab diesen Zeitpunkten nicht weiter beachtet.

Da ein Teil des DSO-Konzepts das Decorator-Pattern darstellt, können mit nur wenig Implementierungsaufwand viele DSOs dargestellt werden. Für den Prototyp wurden somit die grundlegenden Dekoratoren implementiert, wie z. B. ein einfarbig gefülltes Rechteck oder ein Text. Sodass eine Prognose abgegeben werden konnte, ob die Technologie und das DSO-Konzept vereint werden können oder eventuell Probleme auftreten.

Schließlich wurden die beiden Prototypen auf verschiedener Hardware gegenüberge-

stellt, um die Performance der Technologien zu testen. Dabei wurde einerseits untersucht, wie viele DSOs angezeigt werden können, ohne sichtbare Verzögerungen zu erkennen z. B. bei einem Bildwechsel. Andererseits die grafische Anzeige der DSOs selbst, sowie die Interaktion. Um vergleichbare Werte zu haben, wurde auf den Plattformen nach Möglichkeit mit den Browsern Chrome, Firefox, Safari, Internet Explorer sowie deren mobilen Versionen gearbeitet. Was die grafische Anzeige der DSOs mit 2D-Canvas betrifft, wurden keine großen Unterschiede sichtbar, lediglich kleinere Anzeigefehler traten auf. Beispielsweise konnte die kursive Schrift mit dem Firefox-Browser nicht korrekt dargestellt werden. Die Anzeige von SVG mit dem Internet Explorer ist in beiden Versionen (Mobile & Desktop) nicht wie gewünscht, da es mit TSvisIT möglich ist Elemente über den Bildrand hinaus zu schieben, wobei dann nur der Teil innerhalb des Bildes sichtbar ist, hier war ebenfalls der Teil außerhalb des Bildes sichtbar. Ansonsten wurden auch hier keine weiteren größeren Unterschiede festgestellt.

Im weiteren Verlauf der Tests konnten weitere Unterschiede, neben der grafischen Anzeige der DSOs selbst, beobachtet werden. Die zwei Technologien SVG und 2D-Canvas unterscheiden sich einerseits im Bereich der Skalierbarkeit, andererseits bei der Interaktion mit dem Benutzer. Das Bild mit SVG ist eine Vektorgrafik, was verlustfreie Skalierbarkeit voraussetzt. Wohingegen das Bild mit 2D-Canvas eine Rastergrafik darstellt, somit ist ein Qualitätsverlust bei Vergrößerung des Bildes unvermeidlich. Bei Benutzereingaben über die Maus bzw. Touch kommt der Unterschied daher, dass SVG direkt im Document Object Model (DOM) repräsentiert ist. Somit können die Benutzereingaben direkt am DOM-Element

behandelt werden. Bei 2D-Canvas hat man nur die Möglichkeit zu erkennen, dass eine Benutzereingabe auf dem Canvas-Element erfolgt ist, der DSO selbst muss über die Position des Mauszeigers herausgefunden werden. Dies kann allerdings zum Vorteil werden, wenn sich zwei DSOs überlagern und der untere DSO angesprochen werden soll. Bei SVG bekommt automatisch das oberste DOM-Element den Mausklick. Daneben ist ein weiterer Vorteil von 2D-Canvas gegenüber SVG, dass es größtenteils hardwarebeschleunigt ist.

Im Verlauf der Entwicklung der Prototypen und der Performance-Tests wurde deutlich, dass die Technologie 2D-Canvas einerseits näher an der jetzigen Implementierung des DSO-Konzepts in TSvisIT ist, aber auch wesentlich Performanter. Performanter bedeutet, dass mehr DSOs ohne sichtbare Verzögerung angezeigt werden können. Dies kommt zum einen durch die verbreitete Hardwarebeschleunigung des 2D-Canvas zustande, zum anderen durch dessen fehlende Repräsentation im DOM, welches hier bei SVG in Verbindung mit dem DSO-Konzept das

Nadelöhr darstellt. Durch das DSO-Konzept wird bei SVG derzeit für jeden Dekorator ein separates DOM-Element erstellt, dies lässt die Anzahl von DOM-Elementen rapide ansteigen. Das bedeutet: je mehr DSOs pro Bild angezeigt werden müssen, desto geringer ist die DSO-Anzahl, ab der eine Verzögerung in der Bildanzeige sichtbar ist. Auch bei 2D-Canvas ist logischerweise ein Anstieg der Anzeigezeit sichtbar, jedoch nicht so gravierend wie bei SVG.

Der einzige hier wirkliche Vorteil von SVG gegenüber 2D-Canvas liegt in der verlustfreien Skalierbarkeit. Beide Technologien können dieselben grafischen Elemente darstellen und unterscheiden sich dabei nur geringfügig in der Implementierung. Die Anzeige dieser Elemente in den Browsern ist ebenfalls ohne allzu große Unterschiede (Internet Explorer ausgenommen). Je öfter allerdings das Bild neu gezeichnet werden muss, und je mehr DSOs dabei angezeigt werden müssen, desto deutlicher zeigt sich, dass in dem Fall 2D-Canvas die richtige Technologie für diese WebVisu ist.

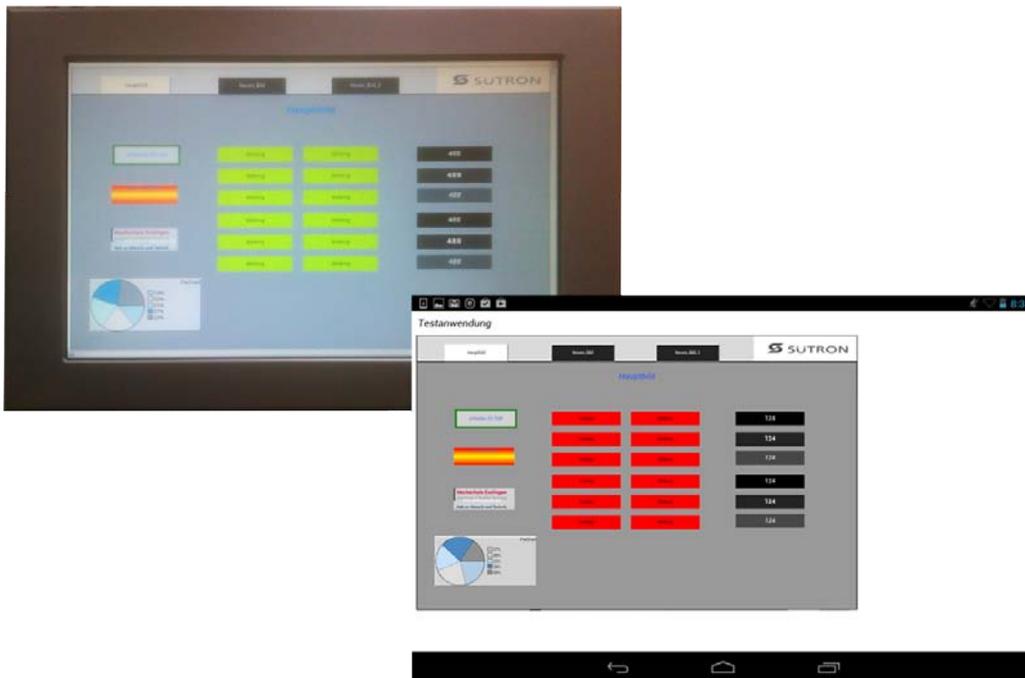


Abbildung 3: WebVisu auf einem Süttron HMI-Display und einem Nexus 7

- [1] Heise online. Marktforscher: Tablet-Markt wächst rasant, PC-Markt bricht weiter ein. <http://heise.de/-1982814>. 12/2013
- [2] Heise online. Beschleunigte Mobilfunknetze: Eine Bestandsaufnahme. <http://heise.de/-1950832>. 12/2013

Bildquellen: Eigene Bilder

Konzeption und Implementierung einer protokollunabhängigen Remote-Support-Anbindung von TeamViewer an einen Headless Server unter Linux

Markus Foitschik*, Peter Väterlein, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die Fernwartungssoftware der Firma TeamViewer GmbH wird stetig um neue Funktionen erweitert und an den sich ständig wandelnden Markt angepasst. Da sich unter Linux im Bereich der grafischen Systeme Änderungen und vor allem Neuerungen anbahnen, soll TeamViewer nun mit einer Architektur versehen werden, die von darunter liegenden graphischen Systemen (inklusive Ein-Ausgabe) unabhängig ist. So soll TeamViewer auch in Zukunft die neuen grafischen Systeme unterstützen.

Als erstes stellt sich die Frage, warum überhaupt eine Alternative zu X11 benötigt wird. X11 ist mittlerweile in die Jahre gekommen und trägt viele Altlasten. Zwar läuft es auf nahezu allen alten und neuen Systemen, aber unter anderem macht es X11 zu schaffen, dass für viele Systeme Sonderfälle bedacht werden müssen. Die meisten Probleme finden sich jedoch im Protokoll von X11, so ergeben sich hier viele sicherheitskritische Fehler, sog. „Race Conditions“, Probleme mit dem Input Stack und Weiteres. Eine Übersicht über einige der Probleme im X11 Protokoll gibt das Dokument „Why X Is Not Our Ideal Window System“[1].

Es wurde schon oftmals versucht, eine Alternative zu X11 zu schaffen bzw. X11 zu verbessern. Oftmals sah es so aus, als würde keine Alternative zu X11 von der Linux Welt akzeptiert werden. Allerdings kristallisiert sich in den vergangenen Monaten heraus, dass nun zwei Alternativen zu X11 im Kommen sind und bereits erwartet werden, sodass die Umstellung von X11 zu einer dieser Alternativen von vielen Distributionen bereits bestätigt wurde.

Diese zwei Alternativen dürften unumstritten Wayland und Canonicals MIR sein.

Beide Systeme zielen darauf ab X11 zu ersetzen und die Probleme, die X11 aufweist, nicht zu wiederholen. Gleichzeitig wird die Kompatibilität zu X11 mittels Layerapplikationen wie XMir oder xwaylandapp sichergestellt.

Wayland setzt sich zum Ziel, eine möglichst perfekte Benutzererfahrung zu liefern. Wayland versucht dies zu erreichen, indem der Server nun noch als Compositor, also als Software, die die einzelnen – bereits gerenderten – Fenster zu einem Gesamtbild zusammenfügt, fungiert. Das Gesamtkonzept, sowie einem detaillierten Vergleich zu X11 kann der Wayland Homepage entnommen werden[3].

Canonicals MIR ist die andere nennenswerte Alternative zu X11 und Wayland. MIR zielt vor allem darauf ab, eine Architektur zu haben, welche es erlaubt MIR (und die MIR Clients) auf möglichst vielen Systemen lauffähig zu halten. So plant Canonical den Einsatz von MIR nicht nur in Ubuntu, sondern auch in dem Ubuntu Phone, einem eigens für Ubuntu entwickelten Smartphone. MIR selbst besteht aus mehreren austauschbaren Komponenten. Die komplette Architektur und Spezifikation ist der MirSpec Seite zu entnehmen[2].

Beide Systeme bringen trotz ihrer Vorteile viel Arbeit für die Entwickler der Clientanwendungen mit sich. Bisher wurden die meisten Applikationen auf Basis der Xlib, bzw. XCB erstellt, welche die Kommunikation mit dem X Server übernehmen. Diese Applikationen müssen angepasst werden für Wayland bzw. MIR, da das Protokoll grundlegend anders ist. Auch Toolkits wie GTK+, Qt, WxWidgets, etc müssen portiert werden, damit diese auf Wayland lauffähig sind (insbesondere, da diese nun selbst für das Zeichnen zuständig sind).

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

Aufgrund dieses bevorstehenden Wandels soll die Software TeamViewer die Möglichkeit haben, den Desktop von neuen Systemen abgreifen zu können. Hierzu gibt es zwar eine bestehende Architektur, die es ermöglicht einen Desktop mit mehreren Methoden abzugreifen, allerdings ist diese für jeweils ein grafisches System zugeschnitten. Diese funktioniert aktuell nur mit genau einem grafischen System und nicht mit vielen verschiedenen Systemen. Die nächste sich ergebende Problematik ist die, dass das Programm gegen die Clientbibliotheken des Displayserver gelinkt werden muss. Wird TeamViewer nun auf einem System gestartet, welches Wayland statt X11 hat, würde die Xlib trotzdem verlangt werden, da dynamisch dagegen gelinkt wurde und der Loader diese Bibliothek nachziehen will. Für die Architektur wurden drei Möglichkeiten evaluiert.

Die erste Möglichkeit ist, mehrere TeamViewer Versionen anzubieten, die jeweils über einen anderen Core verfügen. Die Nachteile hieraus sind leicht ersichtlich. TeamViewer zielt darauf ab, von allen Usern möglichst einfach verstanden zu werden. Muss der User, der sich ggf. mit seinem System nicht auskennt, auf der Homepage entscheiden, welche Version er benötigt, könnte er damit überfordert sein.

Die zweite Möglichkeit ist überhaupt nicht gegen Bibliotheken zu linken, sondern alles über das sogenannte Late Binding nachzuladen. Hierbei müssen alle Funktionen, die aus der Bibliothek benötigt werden, mithilfe von Funktionspointern nachgebildet, die Bibliothek zur Laufzeit nachgeladen, jede einzelne Funktion aufgelöst und als Zeiger gespeichert werden. Dies bringt den Vorteil, dass TeamViewer gänzlich unabhängig von installierten Bibliotheken wäre und damit auf jedem System zumindest starten würde.

Die dritte Methodik liegt darin, einen unabhängigen Desktopcore zu schreiben, der noch keine Implementierung der Architektur kennt,

sondern nur seine Interfaces. Die Implementierungen der Interfaces werden hierbei von Plugins geliefert, die, ähnlich dem Late Binding, zur Laufzeit nachgeladen werden. Hierbei muss zwar ebenso ein Funktionspointer gespeichert werden, allerdings nur einer, der eine Funktion repräsentiert, die wiederum eine Factory zurückliefert. Diese Factory kann dazu verwendet werden, die einzelnen Implementierungen zu erzeugen und an den Desktopcore weiterzureichen. Hier ergibt sich der Vorteil, dass diese Plugins leicht nachgeliefert werden könnten.

Nach Evaluierung der Ansätze wurde schließlich der dritte Ansatz umgesetzt. Es wurden Interfaces für alle abhängigen Teile erstellt, wie Tastatur, Maus und Abgreifen und die bestehende Implementierung von X11 in eine eigene Bibliothek ausgelagert. Eine Illustration des Ansatzes ist Abbildung 1 zu entnehmen. Auf diesem sieht man den generellen Ablauf: TeamViewer lädt das Plugin (in diesem Fall libGrabX11) dynamisch nach und fordert die Factory an. Mithilfe der erhaltenen Factory können nun die Implementierungen der einzelnen Systeme angefordert werden.

Aufbauend auf dieser Architektur wurde nun als Proof-of-Concept die Implementierung von X11 so erweitert, dass bei einem fehlenden X11 System eine Xvfb Instanz gestartet wird. Xvfb ist ein virtueller X11 Server, welcher keinerlei Grafikhardware benötigt, da das komplette Zeichnen im Hauptspeicher des Systems stattfindet. Auf diese Weise kann eine komplette Desktopumgebung z.B. auf Servern genutzt werden.

Der zweite Proof-of-Concept ist das Abgreifen des Framebuffers. Hierbei wird die Textkonsole abgegriffen, womit quasi ein „Zero-Configuration“ SSH-Server gestartet wird, der ohne das für SSH übliche Portforwarding auch hinter Routern funktionsfähig ist. Hierbei sind diverse Teile der Architektur nicht implementiert, wie zum Beispiel das Abgreifen der Maus.

Für die Zukunft ergibt sich, aufbauend auf dieser Architektur, die Möglichkeit, alternative Displaysysteme zu unterstützen. MIR und Wayland wurden nur deshalb noch nicht implementiert, da diese sich noch im Frühstadium der Entwicklung befinden, noch nicht ihr gesamtes Potential ausschöpfen und deswei-

teren wirtschaftlich noch nicht von Interesse sind. Außerdem muss zuerst auf ein voll ausgereiftes System gewartet und verschiedene Möglichkeiten evaluiert werden, das Display abzugreifen.

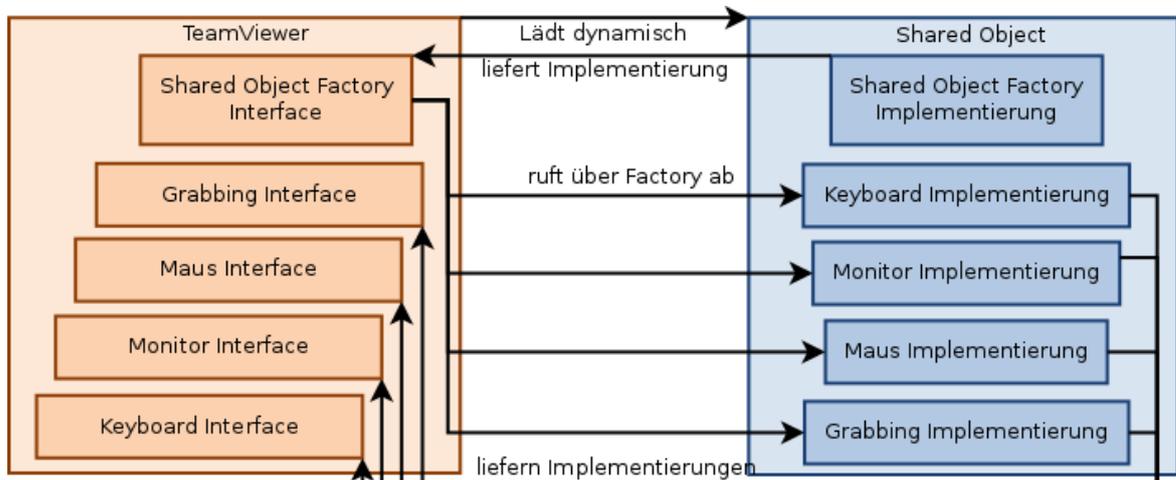


Abbildung 1: Implementierte Architektur

- [1] Hania Gajewska, Mark S. Manass und Joel McCormack, Why X Is Not Our Ideal Window System
- [2] MirSpec, online: <https://wiki.ubuntu.com/Mir/Spec> (zuletzt abgerufen: 28.11.2013)
- [3] Wayland Architecture, online: <http://wayland.freedesktop.org/architecture.html> (zuletzt abgerufen: 28.11.2013)

Bildquellen: eigene Darstellung

Usability für immersives Unterrichten und Lernen: Am Beispiel Cyber-Classroom School Edition

Ann-Kathrin Frey*, Astrid Beck, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Aufgrund von schlechtem Design und nicht durchdachten Bedienkonzepten, ist nicht jede Software einwandfrei bedienbar und oftmals viel zu komplex zu erlernen. Die Nutzer sollten jedoch in der Lage sein Software effektiv und effizient zu nutzen, damit sie durch ihre Anwendungssoftware bei ihrer täglichen Arbeit nicht behindert werden [1]. Aus diesem Grund muss ein Produkt für die Nutzer so gestaltet werden, dass es intuitiv zu bedienen ist und eine gute Gebrauchstauglichkeit und Nutzerzufriedenheit garantiert wird.

Seit dem enormen Fortschritt der 3D-Stereo Technik spielt die Usability nicht mehr nur für 2D-Benutzungsflächen, sondern auch für 3D-Benutzungsflächen eine wichtige Rolle. Die 3D-Stereo Technik hat in den letzten Jahren rapide zugenommen und wird in vielen verschiedenen Themengebieten eingesetzt. Erstreicht wenn potenzielle Nutzer die 3D-Benutzungsflächen

bedienen sollen, ist eine gute Gebrauchstauglichkeit sehr wichtig. In diesem Fall geht es nicht um 3D-Stereo Filme im Kino oder um 3D-Animationen im Internet, sondern um die direkte Nutzung und Interaktion mit einer 3D-Benutzungsfläche.

Ein neues Einsatzgebiet der 3D-Stereo Technik ist der Bildungsbereich. Visuelles 3D-Stereo Lernen wird fortwährend zum Standard an Schulen. Eine 3D-Stereo Lehr- und Lernumgebung muss also so gestaltet sein, dass alle Nutzergruppen – in diesem Fall die Lehrer und Schüler – sie gerne einsetzen und mit ihr effektiv und effizient lernen können. Hierfür müssen sämtliche Bedienkonzepte mit verschiedenen Nutzern geprüft und auf deren Bedürfnisse hin gestaltet werden. Alle Altersklassen der Schüler und Lehrer sollten die 3D-Stereo Software nutzen können und sie vor allem auch gerne einsetzen.

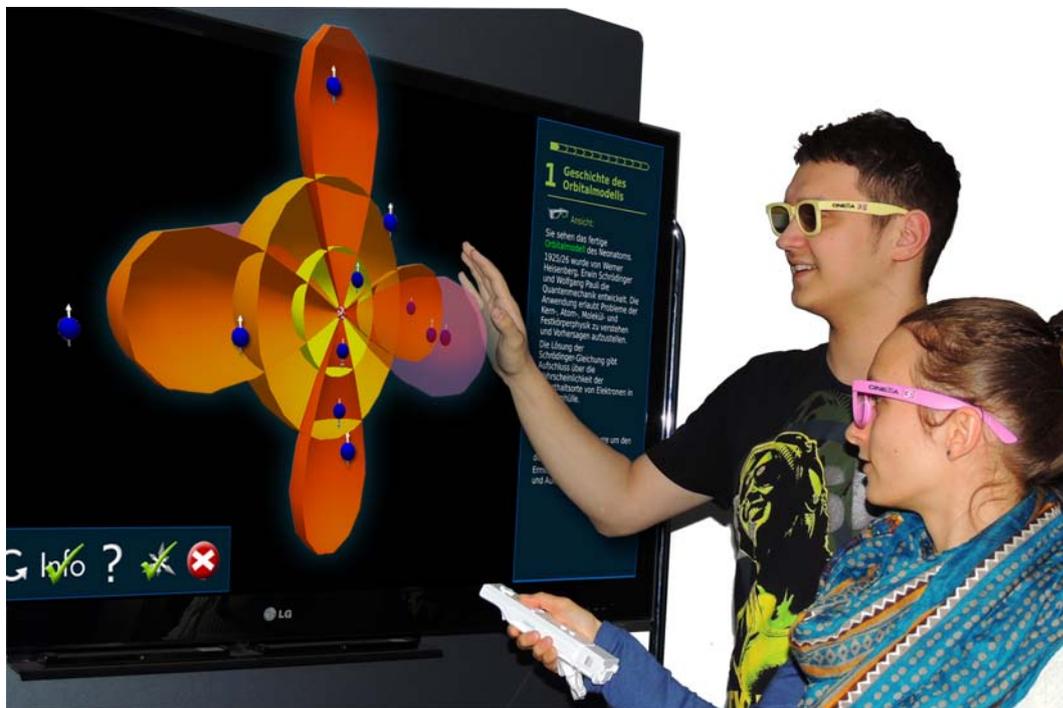


Abbildung 1: Orbitalmodell des Cyber-Classrooms

*Diese Arbeit wurde durchgeführt bei der Firma VISENSO GmbH, Vaihingen

In dieser Bachelorarbeit werden verschiedene Usability Methoden erarbeitet, um eine virtuelle Lernumgebung benutzerfreundlicher und intuitiver zu gestalten. Der Cyber-Classroom ist eine 3D-Stereo Lehr- und Lernumgebung für eine innovative Wissensvermittlung. Mit Hilfe dieser Software, welche hauptsächlich auf einem 3D-Stereo Fernseher läuft und mit Hilfe einer Nintendo Wii Fernbedienung gesteuert wird, können komplexe und räumliche Themen, wie zum Beispiel der Aufbau und die Funktionalität des Orbitalmodells, dem Nutzer besser veranschaulicht werden (siehe Abbildung 1).

Es wird nicht nur ein Konzept der benutzeradaptiven Benutzungsoberfläche, sondern auch ein neues Interaktionskonzept erstellt. Die Interaktion des Cyber-Classrooms soll für den Nutzer zukünftig auch über Smartphones bedienbar sein. Um eine Benutzungsoberfläche benutzerfreundlich zu gestalten, muss der Nutzer in den Entwicklungsprozess mit eingebunden werden. Es werden somit, vor Beginn der Entwurfsphase, einige Interviews und Diskussionen mit potentiellen Nutzern durchgeführt. Zur Verdeutlichung, welche Anwender den Cyber-Classroom zukünftig nutzen, werden die entsprechenden Nutzergruppen identifiziert und User Stories verfasst. Diese User Stories dienen insbesondere dazu, ein konkretes Szenario zu beschreiben, wie

und von wem der Cyber-Classroom zukünftig genutzt wird. Die Erstellung eines Storyboards anhand einer Nutzgruppe ist auch Bestandteil dieser Arbeit, um die einzelnen Szenarios visuell darzustellen. Anhand der Ergebnisse der Nutzungskontextanalyse und den definierten System- und Usability Anforderungen werden neue Konzepte entworfen und schon bereits existierende Konzepte überarbeitet. Hierfür werden existierende User Interface Guidelines und Multitouch Guidelines hinzugezogen, um schon existierende Elemente, wie zum Beispiel Buttons oder Checkboxes, sowie etablierte Gesten, zu verwenden.

Für eine spätere Weiterentwicklung der einzelnen Konzepte des Cyber-Classrooms, wird eine UI Spezifikation erstellt. Weiterhin werden die einzelnen User Interface Komponenten der Software, sowie das Interaktionskonzept mit den zugehörigen Funktionen dokumentiert. Hinsichtlich der angefertigten Konzepte und der User Interface Spezifikation wird dann ein Prototyp des ausgearbeiteten Interaktionskonzeptes erstellt und das Designkonzept des Cyber-Classrooms angepasst, sodass eine abschließende Konzeptevaluation durchgeführt werden kann. Die einzelnen Usability Tests, die im Zuge dieser Arbeit erstellt werden, dienen zur Bestätigung der einzelnen Konzepte und um kleinere Usability Anpassungen vornehmen zu können.

Simulation von Fahrzeugsensoren zur Umgebungserfassung für Fahrerassistenzsysteme in OpenCL unter Verwendung einer GPU einschließlich Performancevergleich der GPU-Implementierung mit bestehenden Simulationen auf Basis einer CPU

Ilja Galat*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Der funktionale Test eingebetteter Software ist ein zentraler Bestandteil bei der Entwicklung von Fahrerassistenzsystemen im Automobil. Fahrerassistenzsysteme (FAS) unterstützen den Fahrer in alltäglichen Fahrsituationen. Mit Hilfe von Kameras, Radarsensoren und weiteren Sensoren wird dabei die Fahrzeugumgebung abgetastet und die gewonnene Information an den Fahrer weitergeleitet. In Abbildung 1 ist eine alltägliche Situation abgebildet. Fahrerassistenzsysteme sollen die Sicherheit und den Fahrkomfort für die Fahrzeuginsassen erhöhen. So kann zum Beispiel ein Spurhalte-Assistent das Fahrzeug während der Fahrt davon abhalten, bei schlechtem Wetter von der Fahrspur abzuweichen. Forschungsinstitute und Automobilhersteller sind ständig auf der Suche nach neuen innovativen Technologien für Fahrerassistenzsysteme. Durch die schnelle und hohe Verbreitung der FAS sind sie in der Automobilindustrie zunehmend zur unverzichtbaren

Fahrzeugausstattung geworden.

Auf Grund ihrer hohen Bedeutung und ihrer oftmals sicherheitskritischen Eigenschaften müssen solche Systeme auf korrektes Verhalten akribisch überprüft werden. Basierend auf den Anforderungen an die Fahrerassistenzsysteme werden Testfälle in Form von Fahrmanövern abgeleitet und beispielsweise an einem Prüfstand durchgeführt. Dabei kommen verschiedene HiL- (Hardware in the Loop-) oder SiL- (Software in the Loop-) Prüfstände zum Einsatz. In den Fahrmanövern werden das Verhalten von mehreren Fahrzeugen und verschiedene Eigenschaften der Fahrbahn und Umgebung definiert. Bei der zunehmenden Simulationskomplexität nimmt der Berechnungsaufwand der Simulation stark zu, was zu Echtzeitverletzungen während einer HiL-Simulation führen kann. Solche Echtzeitprobleme können das Testergebnis beeinflussen und zu Mehraufwänden im Testprozess bis hin zu einem unerkannten Fehler führen.

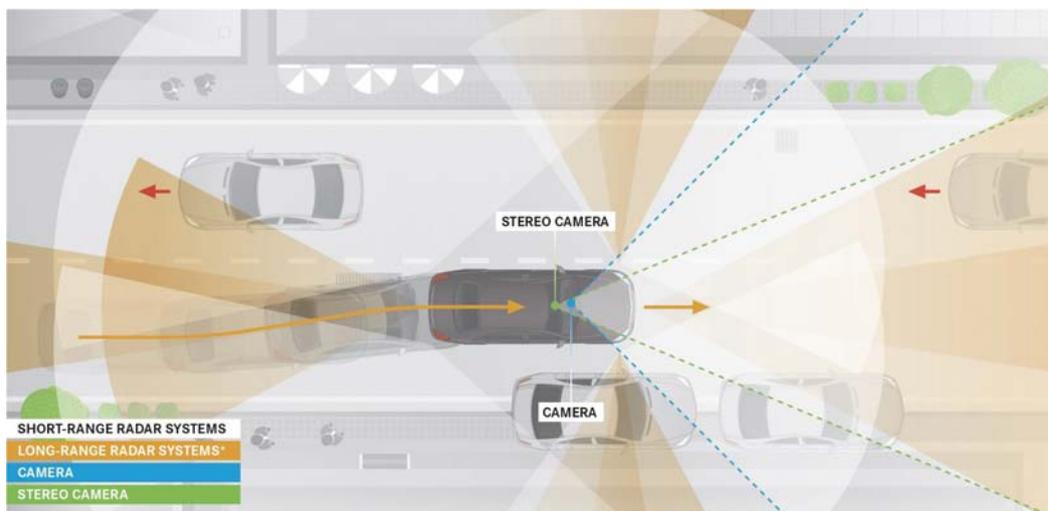


Abbildung 1: Mercedes-Benz S 500 Intelligent Drive

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

Im Rahmen dieser Arbeit soll ein möglichst realitätsnahes Sensormodell für ein Simulationsprogramm implementiert werden. Dabei soll die Berechnung von Simulationsdaten nicht, wie im industriellen Umfeld heute üblich, auf einer CPU, sondern auf einer Grafikkarte stattfinden. Da die Simulation der im Fahrzeug verbauten Sensoren immer aufwändiger wird, soll die Rechenleistung der Grafikkarten zusätzlich zur CPU genutzt werden, um letztere zu entlasten. Im Gegensatz zur CPU bietet eine Grafikkarte die Möglichkeit, die GPU-Recheneinheiten parallel ausführen zu können, was bei optimaler Nutzung zu einer enormen Steigerung der Rechenleistung führt. Das Prinzip von GPU-Recheneinheiten geht auf das SIMD-Verfahren (Single Instruction Multiple Data) zurück, bei dem Befehle gleichzeitig auf mehrere Datensätze parallel angewendet werden. Inzwischen gibt es verschiedene Programmiersprachen, die eine parallele Programmierung ermöglichen. Im Rahmen dieser Arbeit wird OpenCL verwendet. OpenCL (Open Computing Language) ist ein offener Standard zur Parallelprogrammierung auf heterogenen Systemen. Das Sensormodell soll

zunächst auf sechs bis acht Radarsensoren, die an einem Simulationsfahrzeug angebracht sind, eingeschränkt werden. Jeder Sensor soll mit jedem Simulationsschritt die relevanten Informationen über die detektierten Objekte an die Simulationsumgebung liefern. Hierzu zählen zum Beispiel die Objekt-Koordinaten oder die kürzeste Distanz zwischen dem simulierten Fahrzeug und einem Hindernis. Da die Grafikkarte über den PCIe-Bus mit eher geringer Transfer rate angeschlossen ist, bildet der PCIe-Bus bei hohem Datenvolumen den Flaschenhals. Aus diesem Grund war im zweiten Teil der vorliegenden Arbeit eine Performance-Analyse mit Vergleich einer allgemeinen Implementierung und einer OpenCL-Implementierung zu erstellen. Dabei wurde nach Möglichkeiten zur Erhöhung des Datentransfers zwischen Host (CPU) und Device (Grafikkarte) gesucht. Abbildung 2 zeigt das Ausführungsmodell von OpenCL. Die Optimierung der Rechenzeit unter Verwendung einer GPU und OpenCL war letztendlich der Schwerpunkt der vorliegenden Arbeit.

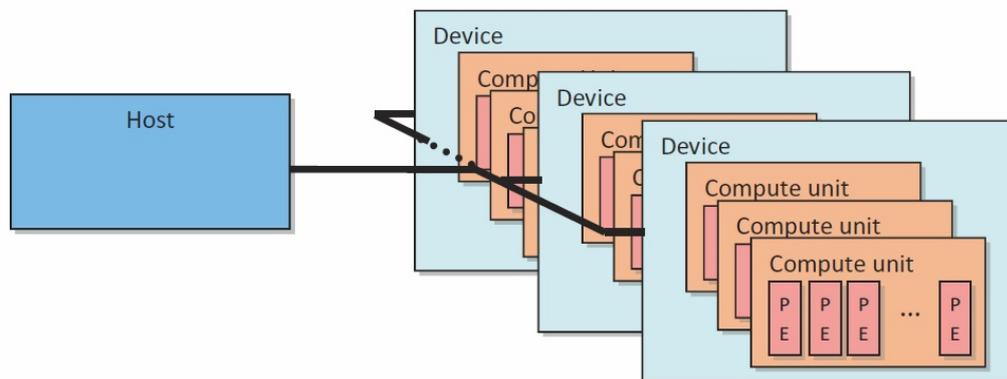


Abbildung 2: Das Ausführungsmodell von OpenCL

Bildquellen:

- Abbildung 1: Daimler AG, <http://technicity.daimler.com/mercedes-benz-s-klasse-intelligent-drive> 12/2013
- Abbildung 2: Benedict R. Caster, Heterogeneous Computing with OpenCL, 2013

Design and development of a test-framework for a non-locally deterministic workflow engine

Thomas Gottuck*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Workflow is everywhere. From the processing and tracking of an entire order to the execution of a single production step by a machine, anything can be achieved by utilising workflows. The workflow engine subject to this bachelor thesis is a highly complex system developed by Lukaseder Hard- und Softwareentwicklung which manages distinctive realms, each populated with users which in turn have particular privileges, allowing them to manage for example objects, other users, files or workflows itself. A workflow is a series of states with transitions leading from any state to any other. They allow the workflow to progress from state to state, when the conditions of the transition are matched. Actions are executed as the workflow enters or leaves states. To run a workflow, it is allocated to an object and over the course of its progress may change values in the freely customizable structure of the object, start further workflows or communicate with modules connected to the workflow-engine. Depending on the scenario of use, the workflow might run autonomously or be dependent on user input. Typically, a user will make use of front end client software which connects to the back end via the connector. The connector handles the connection to the back end and provides API functionality. The back end consists of a server, handling all requests and a SQL-database where all data is stored.

Yet, as most human engineers are not flawless, so is also – with increasing complexity – their work. The quality of software can be increased in many ways, one of them being testing. Software tests have the ambition to reveal flaws and rate the performance of software. Possible testing scenarios range from reviewing concepts and code, to testing small parts independently, to testing software as a whole. It must be noted however, that putting effort into testing can prove the existence of bugs in code, but not the absence of such[1]. Test scenarios may be countless and straining, but “trifles make perfection, and perfection is not trifle”[2].

Therefore, the object of this bachelor thesis

is to thoroughly test the workflow-engine to ensure a high standard of quality. This is achieved by simulating client activity in a framework substituting the front end, thus using the connectors API functions, and verifying the desired processing. Avoidance of faulty verification is maintained by additional checking of the data stored in the database directly. Furthermore, to ensure comparable results, initial conditions as identical as possible are to be established before any test is conducted.



Figure 1: Scanning a ticket is one of many ways of possible input to the workflow engine

When the test-framework is started, it reads an XML file containing configuration data like the IP address and login data of the database and the back end server, and tests. Tests consist of one or more instructions, which may be conditions or actions. An action performs anything that is possible by using the API, while a condition might additionally connect to the database. Each test possesses prerequisites for its execution, for instance a time period or a condition. An example would be a test, that is started only if 500 ms have passed since the launch of the framework and a condition, checking the existence of a user, has succeeded. It might then, for instance, create an object with the aid of that user. The test-framework will continue until all tests are completed or there are no more tests left of which the prerequisites can be matched at present or future. Finally, the results of the

*This work has been carried out at the company Lukaseder Hard- und Softwareentwicklung, Stuttgart

tests are stored in a CSV file for analysis. If configured that way, which is only possible if the test is run on a local back end server launched by the test-framework, the database is completely erased and reset to an initial state prior to the test, containing only one realm with a single, administrative user. This reset function is determined to provide the framework with a deterministic behaviour, meaning that conducting the same tests will lead to the same results. In a common application, the system may behave non-deterministic because of factors hidden from the user, like the CPU utilization of the system the back end server is running on, the clients' network capacity or the fragmentation of the database.

```

Testframework
Reset = True
Backend IP = 127.0.0.1
Backend port = 22
Backend user = Administrator
Backend pass = WorkFrame
Database IP =
Database user =
Database pass =

Reading test node
"trigger on privilege not found":
Triggercondition:
Label = check privilege
Username = Administrator
Database = null
Password = WorkFrame
Realm = null
TriggerInfo:
Repeat = 3
Action:
Label = create privilege
Username = Administrator
Realm = null
Password = WorkFrame

Reading test node
"trigger on privilege found":
Triggercondition:
Label = check privilege
Username = Administrator
Database = null
Password = WorkFrame
Realm = null
TriggerInfo:
Repeat = 2
Action:
Label = remove privilege
Username = Administrator
Realm = null
Password = WorkFrame

Reading test node
"wait 2000ms, create workflow":
TimeInfo:
Start = 2000
End = -1
Action:
Label = wf1 create
Username = Administrator
Realm = null
Password = WorkFrame
Condition:
Label = wf1 check
Username = Administrator
Database = null
Password = WorkFrame
Realm = null
Levels = 0-0-0

Reading test node
"wait 2500ms, try something that's not allowed":
TimeInfo:
Start = 2500
End = -1
Action:
Label = wf1 create
Username = Administrator
Realm = null
Password = WorkFrame
Condition:
Label = wf1 check
Username = Administrator
Database = null
Password = WorkFrame
Realm = null
Levels = 0-0-0

Deleting databases:
null
null DROPPED
null CREATED
null REGISTERED
Standardadmin created
[18:44:35.529] PostLoad null
[18:44:35.538] Load Users
[18:44:35.544] Load Users took 00:00:00.0146948
[18:44:35.544] Load ActiveData
[18:44:35.546] Load ActiveData took 00:00:00.0013832
[18:44:35.546] Load linkedRealms
[18:44:35.549] Load linkedRealms took 00:00:00.0031612
[18:44:35.549] Load Workflows
[18:44:35.550] Load Workflows took 00:00:00.0013699
[18:44:35.550] Load ids
[18:44:35.551] Load ids took 00:00:00.0047672
[18:44:35.551] Load RealmPrivs
[18:44:35.558] Load RealmPrivs took 00:00:00.0047672
[18:44:35.558] PostLoad null took 00:00:00.0298762
RDW

[176.83] ms] Running test "trigger on privilege not found"
[606.0855] ms] Running test "trigger on privilege found"
[662.0934] ms] Running test "trigger on privilege not found"
[697.0984] ms] Running test "trigger on privilege found"
[707.1825] ms] Running test "trigger on privilege not found"
[2000.2816] ms] Running test "wait 2000ms, create workflow"
[2500.3519] ms] Running test "wait 2500ms, try something that's not allowed"

Finished test routine at 2514.3537 ms

OK "trigger on privilege not found"
OK "trigger on privilege found"
OK "trigger on privilege not found"
OK "trigger on privilege found"
OK "trigger on privilege not found"
OK "wait 2000ms, create workflow"
Failed "wait 2500ms, try something that's not allowed"
- Action "wf1 create" failed
- The named privilege does not exist

Finished. [Press any key]

```

Figure 2: Exemplary output of the test-framework

- [1] Steve McConnell. Code Complete, Second Edition. Microsoft Press, Redmond, Washington 98052-6399. ISBN 978-3-86063-593-3
- [2] Michelangelo

Figure sources:
Figure 1-2: Thomas Gottuck

Realisierung einer Musikabspielplattform zur verbesserten Demonstrationsfähigkeit einer Class-D Audioendstufe im Bereich der Modelleisenbahn.

Martin Heinzmann*, Karlheinz Höfer, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Auch im Bereich der Modelleisenbahn nutzt man ein immer breiter werdendes Spektrum der digitalen Signalverarbeitung. Deshalb werden bei der Märklin Engineering GmbH ständig neue Technologien erforscht und Machbarkeitsstudien durchgeführt, worunter auch diese Bachelorarbeit fällt. [1]

Um ein gutes Klangerlebnis zu bekommen bedarf es keinem besonders hoch abgetastetem Audiosignal. Ab einem gewissen Punkt wird durch höhere Abtastraten keine hörbare Verbesserung der Klangqualität erreicht. Wenn also an der Art der Aufnahme eines Signals keine Verbesserung mehr erreicht werden kann, muss man sich Gedanken um die Verarbeitung und Ausgabe machen.

Um die Wiedergabequalität eines Audiosignals zu verbessern, kann der Frequenzgang

des Lautsprechers geglättet werden. Physikalische Optimierungen an den Lautsprechern oder der Entwurf analoger Filter sind zwar möglich, jedoch steht der damit verbundene Aufwand in keinem Verhältnis zu dem zu erwartenden Ergebnis. Was in der Filterentwicklung im Analogbereich noch aufwendig und unflexibel durch Hardware erreicht wurde, lässt sich nun mit einer digitalen Audioendstufe, in Verbindung mit digitaler Signalverarbeitung, elegant und vielfältig umsetzen.

Der in Abbildung 1 dargestellte blaue Kurvenverlauf zeigt den unveränderten Frequenzgang eines Lautsprechers. Die grünen Linien grenzen den gewünschten Bereich für den optimierten Frequenzgang ein. Der rote Kurvenverlauf zeigt den durch die Audioendstufe optimierten Frequenzgang des selben Lautsprechers.

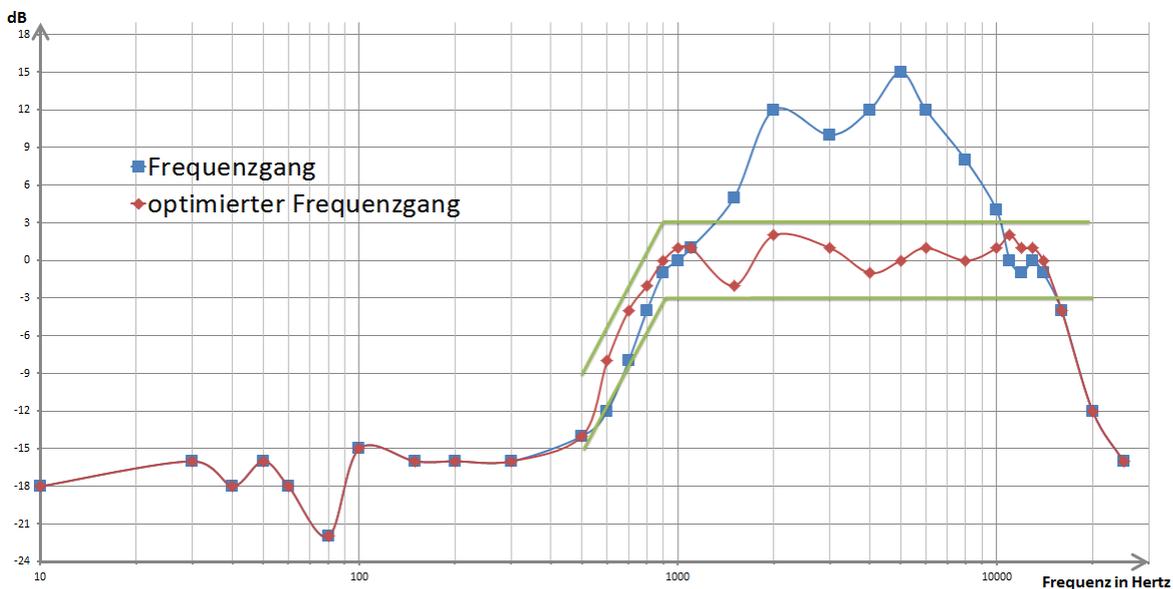


Abbildung 1: Frequenzgang

Im Rahmen dieser Bachelorarbeit wurde die Realisierung einer Musikabspielplattform zur verbesserten Demonstrationsfähigkeit einer Class-D Audioendstufe im Bereich der Modelleisenbahn evaluiert und implementiert. Durch

das Abspielen von Musik, lässt sich die Verbesserung der Klangqualität durch die digitale Audioendstufe hervorragend demonstrieren.

*Diese Arbeit wurde durchgeführt bei der Firma Märklin Engineering GmbH, Göppingen

Die für die Musikabspielplattform verwendeten Hardwarekomponenten sind:

- eine microSD-Karte – für den Zugriff auf die Audiodaten
- ein Microcontroller – zum Auslesen, Verarbeiten und Übertragung der Audiodaten von der microSD an die Audioendstufe, mit folgenden Spezifikationen
 - max 50 MHz Taktfrequenz
 - 32-Bit-Architektur
 - 64 Kb RAM, 512 Kb ROM
- eine Class-D Audioendstufe – zur digitalen Aufbereitung der Audiodaten
- ein Lautsprecher – zur Wiedergabe der angepassten Audiodaten

Der Microcontroller und die microSD kommunizieren über die SPI-Schnittstelle, da diese am besten dafür geeignet ist. [2] Die Audioendstufe wird über die I2C-Schnittstelle durch den Microcontroller konfiguriert. Die Audiodaten überträgt der Microcontroller per I2S-Protokoll, welches über den SCI-Bus implementiert ist, an die Audioendstufe.

Um den Frequenzgang des eingesetzten Lautsprechers zu optimieren, muss dieser im unveränderten Zustand ausgemessen werden. Anhand der Messung können die benötigten Equalizer-Einstellungen ermittelt und an die Audioendstufe übertragen werden. Mit Hilfe ihrer fünf Equalizer, kann die Audioendstufe nun einen optimierten Frequenzgang erzeugen. Da die Wiedergabe Mono erfolgt, werden bei Stereo-Signalen beide Kanäle von der Audioendstufe addiert und das Ergebnis halbiert. Somit entstehen keine Verluste des Signals.

Um die Musik steuern zu können, wurde die Kommunikation zwischen der Central Station 2 – ein von Märklin entwickeltes Steuergerät für Modelleisenbahnen – und der Musikabspielplattform implementiert. Dadurch lässt sie sich ohne viel Aufwand in jegliches Modell integrieren.

Einer enormen Verbesserung der akustischen Wahrnehmung während des Spielbetriebes steht nun nichts mehr im Wege.

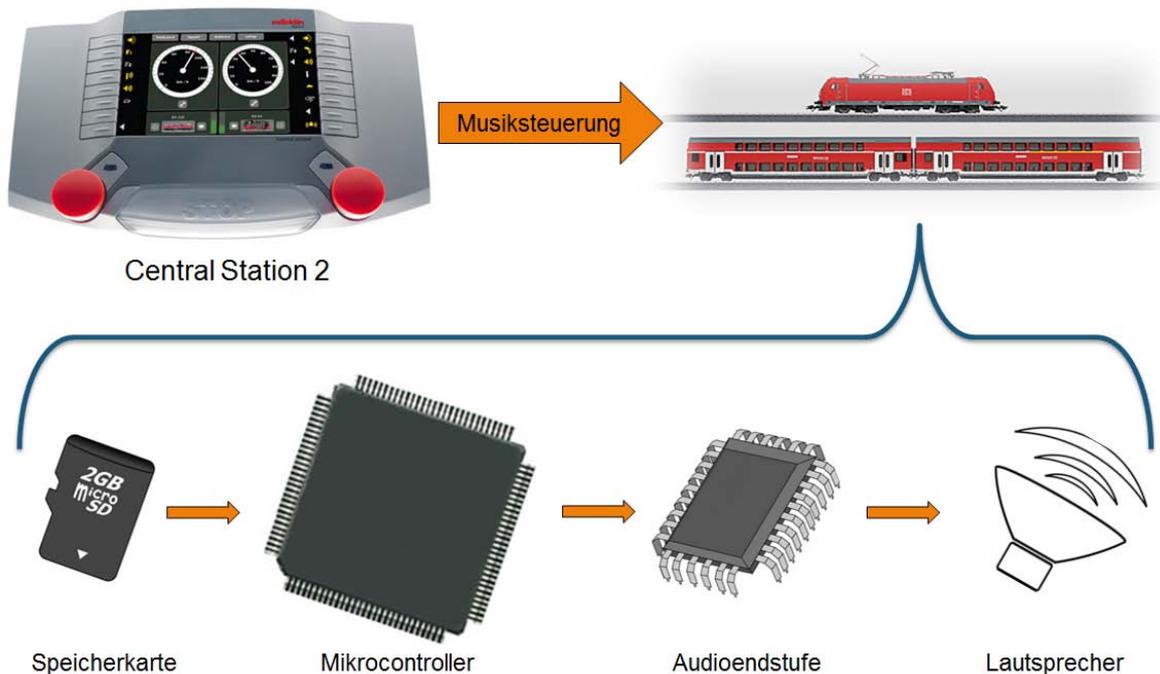


Abbildung 2: Verwendete Hardwarekomponenten

[1] <http://www.maerklin-engineering.com>
12/2013

[2] http://www.mikrocontroller.net/articles/MMC-_und_SD-Karten 12/2013

Design und Implementierung eines Objektbaums für die Abbildung des AUTOSAR Virtual Function Bus in Python

Simon Helming*, Reiner Marchthaler, Hermann Kull

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Ziel von AUTOSAR ist es den Austausch von Software auf Steuergeräten zu erleichtern. AUTOSAR wurde von einigen großen Automobilhersteller sowie Automobilzulieferern im Jahr 2002 gegründet. Das Ziel von AUTOSAR ist der Entwurf von nachhaltigen Produkten sowie die Minimierung der Kosten und Ressourcen. AUTOSAR beschreibt Standards für Software im Auto und wird in mehreren Schichten unterteilt.

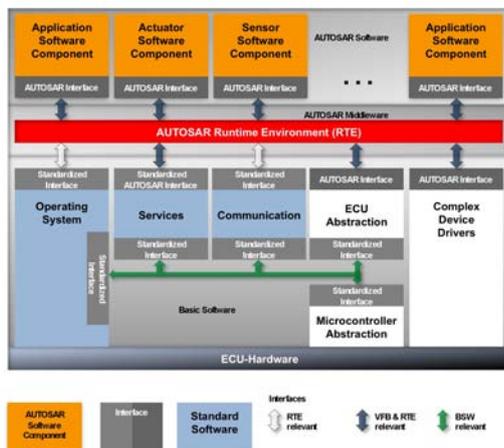


Abbildung 1: AUTOSAR Architektur

Die Kommunikation zwischen den einzelnen „Software-Components“ wird über den „Virtual Functional Bus (VFB)“ ermöglicht. Der „Virtual Functional Bus“ ermöglicht eine Kommunikation zwischen einzelnen „Software-Components“ sowie zwischen einzelnen „Electronic Control Units (ECU)“. Durch die Kommunikation über den „VFB“ sind „ECUs“ unabhängig und können ausgetauscht werden, ohne dass die Software neu geschrieben werden muss. Der Datenaustausch erfolgt über das ARXML-Format, welches auf dem XML-Format basiert. Hier werden die einzelnen Komponenten definiert sowie die Verbindung

zu anderen Komponenten festgelegt. [2]

Die Aufgabe dieser Bachelorarbeit war es nun dieses ARXML-Format in der Programmiersprache „Python“ abzubilden. Es sollte möglich sein eine ARXML-Datei einzulesen und diese soll selbständig in Python abgebildet werden um sie später weiter verarbeiten zu können.

Für das Lösen der Aufgabe wurden zwei Konzepte entwickelt.

Für das erste Konzept hat man den Aufbau des ARXML-Formats näher betrachtet. Dieser Aufbau sollte in Python direkt umgesetzt werden. Der Vorteil den man hieraus ziehen kann ist, dass man ein gegebenes Konzept hat, was eine bestimmte Reife aufweisen kann und somit robust ist. Die Verständlichkeit basiert auf dem ARXML-Format und es ist nicht notwendig sich in ein weiteres Konzept einarbeiten zu müssen. Die Nachteile sind, dass mögliche Verbesserungen der Struktur des Aufbaus oder der Flexibilität wegfallen.

Das zweite Konzept greift hauptsächlich die Nachteile des ersten Konzeptes auf. Man analysiert den Aufbau der ARXML-Datei und versucht ein besseres oder gar effizienteres Konzept zu erhalten.

Versucht man ein eigenes Konzept zu entwickeln, hat man die Chance aus den Fehlern anderer Konzepte zu lernen. Es bietet Gelegenheit seine eigenen Ideen und Vorstellungen von dem richtigen Aufbau zu verwirklichen. Das Konzept kann in seiner Struktur robuster sein, weniger anfällig auf Fehler, die Effizienz kann gesteigert werden, das Erzeugen oder das Lesen wird vereinfacht. Versucht man ein eigenes Konzept zu entwickeln, beansprucht dies jedoch Zeit. Man entschied sich für das erste Konzept. Da bei dieser Aufgabe eine neue Entwicklung eines eigenen Konzept nicht notwendig ist und das Konzept sehr gut ist hat man auf das zweite Konzept verzichtet.

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

Bei der Abbildung des VFB in Python wurden verschiedene Klassen erzeugt, die jedes Element (Component, Composition, Connector, Port) separat verarbeitet. Die eigentliche Verarbeitung und Generierung fand in der Klasse „VFB“ statt. Hier wurde die gesamte Verarbeitung gestartet: vom Einlesen bis zum Abbilden der ARXML-Datei. Durch die Separierung der einzelnen Elemente kann man bei Bedarf einzelne Klassen bearbeiten, ohne dass das gesamte System beeinträchtigt wird.

Das Programm soll nicht nur die ARXML-Datei in Python abbilden, sondern es sollte ebenfalls möglich sein, Verbindungen einer gewählten Komponente zwischen anderen Komponenten anzuzeigen und auszugeben. Diese Aufgabe wurde durch einen Suchalgorithmus gelöst werden, welcher eine rekursive Schleife benutzte um jeden einzelnen Pfad von außen nach innen des Komponentenbaumes abzulaufen.

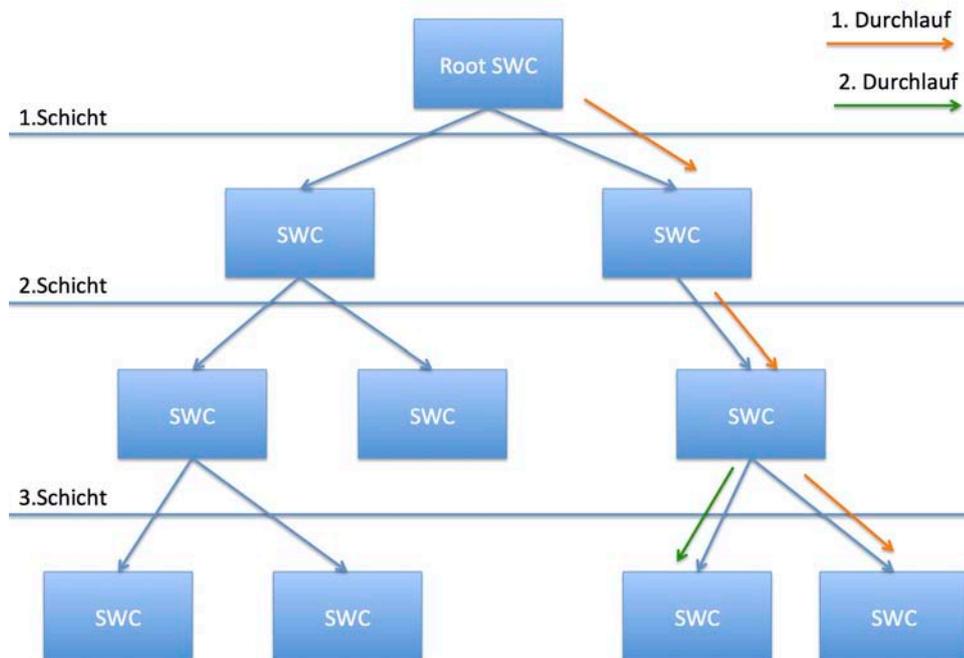


Abbildung 2: Autosar Durchlauf

Abbildung 2 zeigt einen solchen Vorgang. Hier wird zuerst der äußere Pfad durchlaufen. Beim zweiten Durchlauf beginnt er jedoch nicht ganz oben, da die SWC in Schicht 2 weitere SWC's besitzt und diese ebenfalls überprüft. Der Algorithmus läuft solange den Pfad entlang, bis er entweder die gesuchte Komponente gefunden hat oder am Ende ist. Hat der Algorithmus die gesuchte Komponente gefunden, wird die Start-SWC sowie der verbundenene Port in eine Liste eingetragen und am Ende zurückgegeben.

Der Aufbau der Abbildung in Python ähnelt dem Aufbau der ARXML-Datei. Jedes Element im VFB wird von einer jeweiligen Klasse erzeugt. Dadurch ist es möglich Verarbeitungsschritte ohne starke Beeinträchtigung des Gesamtsystems vorzunehmen. Die Ausgabe der Verbindungen einer SWC erfolgt durch eine rekursive Schleife, die das System von der Wurzel her abläuft, bis alle Pfade überprüft wurden.

[1] <http://autosar.org/index.php?p=1&up=1&uup=2&uuup=0>
12/2013

[2] <http://efinio.com/presse/AUTOSAR.pdf>
12/2013

Bildquellen:

- Abbildung 1: http://www.autosar.org/gfx/AUTOSAR_ECU_softw_arch_b.jpg 12/2013
- Abbildung 2: eigene Zeichnung

Konzeption und einer Systemsimulation für Aufzugssysteme sowie die beispielhafte Umsetzung ausgewählter Komponenten

Lukas Janssen*, Hermann Kull, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Große Gebäudekomplexe wie Banken, Krankenhäuser oder Einkaufszentren sind ohne Aufzüge nicht mehr denkbar. Neben dem Durchsatz sind Zuverlässigkeit, Verfügbarkeit und Fahrkomfort wichtige Merkmale. Zunehmend ist auch das Umweltbewusstsein und das damit verbundene Interesse an Energieeffizienz und Nachhaltigkeit. Um diese Faktoren zu erfüllen, bedarf es einer intelligenten Steuerung. Die Entwickler solcher Steuerungen benötigen unterstützende Tools, um die zukunftsorientierten Innovationen in neuen Steuerungsgenerationen zu implementieren.

Am Ende jeder Entwicklung steht der Systemtest. Er muss gewährleisten, dass die Steuerung mit allen beteiligten Komponenten, also dem System, fehlerfrei interagieren kann. Jeder Betriebszustand und jede Aufgabe der Steuerung muss einwandfrei funktionieren, wenn die Steuerung im Feld eingesetzt werden soll. Um Kosten, Aufwand und Platz zu sparen, wird die Steuerung erst an einer Simulation erprobt. Der bisher verwendete Simulator entspricht, in mancher Hinsicht, einer idealen Welt. Er bietet z.B. keine Möglichkeit den bei der Fahrt auftretenden Schlupf zu simulieren. Ebenfalls bietet er keine Option, Testfälle automatisiert zu durchlaufen und ausreichend nachvollziehbar zu protokollieren. Um diese und weitere Optionen zukünftig nutzen zu können, soll im Rahmen dieser Arbeit eine neue Simulation konzipiert werden.

Für die Konzeption einer solchen Simulation sind mehrere, teilweise iterative, Schritte notwendig. Da keine alten Lösungen berücksichtigt werden sollen, kann eine beliebige Methodik gewählt werden.

Der gewählte Lösungsansatz orientiert sich an dem Problembereich. Im Vordergrund der Problemdefinition stehen die Anforderungen, welche mit den Anwendungsfunktionen des Systems übereinstimmen. Hierfür ist die aus der Literatur bekannte objektorientierte Systemanalyse erforderlich, da das zu simulierende System aus vielen verteilten Komponenten besteht. Um diese in Software nachbilden zu können, müssen sie abstrakt

dargestellt werden. Das entspricht genau



Abbildung 1: Kabine im Schacht

dem Gedanken der objektorientierten Programmierung [1].

Die Herangehensweisen der objektorientierten Systemanalyse unterscheiden sich je nach Literatur (vgl. [1] und [2]), da verschiedene Problemstellungen verschiedene Techniken erfordern. Im Grunde lässt sich das Vorgehen wie folgt gliedern:

- Anwendungsfälle und Anforderungen sammeln
- Anforderungen priorisieren
- Kontextdiagramm erstellen
- Anforderungen ergänzen
- Use-Case-Diagramme erstellen
- Klassen der konzeptionellen Sicht (Fachklassen) ermitteln
- Ablaufmodelle entwickeln

*Diese Arbeit wurde durchgeführt bei der Firma ThyssenKrupp Elevator Innovation GmbH, Neuhausen

- Architektur erstellen und Schnittstellen beschreiben
- Klassendiagramm erstellen

Die einzelnen Schritte können mehrfach durchlaufen und verfeinert werden.

Anfangs sind Anwendungsfälle und Anforderungen zu formulieren. Für ein Projekt dieser Größe und in diesem Zeitrahmen, kann eine einzelne Person keine vollständige Requirementspezifikation durchführen. Mithilfe selbst erstellter Anwendungsfälle konnten aber die Hauptanforderungen gesammelt werden. Die vielen Nebenanforderungen an ein vollständiges Produkt, können durch den zeitlich begrenzten Rahmen nur teilweise berücksichtigt werden.

Die gefundenen Anwendungsfälle werden in den Anforderungen festgehalten. Da ein Anwendungsfall meistens mehrere Vorgänge beinhaltet, werden die Anforderungen um diese Vorgänge ingenieurwissenschaftlich erweitert. Die technischen Anforderungen werden also ebenso betrachtet. Mit diesem Schritt entstehen die tatsächlichen Anforderungen an die Software. Sie sollten widerspruchsfrei und vollständig sein, denn nur was gefordert wird, kann später auch implementiert werden.

tiert werden.

Eine modulare Architektur soll die Wartbarkeit und Austauschbarkeit von Komponenten unterstützen. Die Geschäftsprozesse und Grenzen des zu simulierenden Systems werden festgelegt und genau analysiert. Hierbei hilft das Kontextdiagramm, welches schon schematisch die Datenflüsse darstellt. Für das Kontextdiagramm werden Komponenten und Kommunikationsbeziehungen erfasst und anschließend in Entity-Objekte überführt.

Die Änderbarkeit von Komponenten ist wegen stetiger Entwicklung und großer Variantenvielfalt von großer Bedeutung und wird durch präzise Definition der Schnittstellen zwischen den Objekten erreicht. Unter Berücksichtigung der Anwendungsfälle und mithilfe von Kontroll-Objekte bilden die Entity- und Interface-Objekte eine Verarbeitungsschicht. Eine höhere Ebene kümmert sich um die Benutzerinteraktion, die Kommunikation mit der Steuerung und ggf. um automatisierte Testfälle.

Die Schwierigkeit bei solch eine Konzeption liegt darin, alle Zusammenhänge und Beziehungen zu erfassen, diese zu priorisieren und in ein integriertes Ganzes zu überführen.

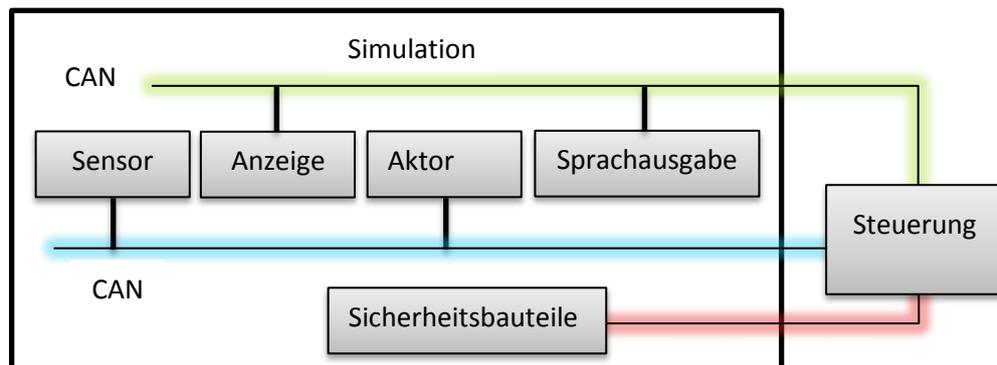


Abbildung 2: Beispielhafte Simulationsgrenze

- [1] Bernd Oestereich, Analyse und Design mit der UML 2.5, Objektorientierte Softwareentwicklung, Olenbourg Verlag; {10. Auflage}
- [2] Joachim Goll, Methoden und Architekturen der Softwaretechnik, 2011, Vieweg + Teubner Verlag | Springer Fachmedien Wiesbaden GmbH; {1. Auflage}

Bildquellen:

- Abbildung 1: Eigene Grafik
- Abbildung 2: www.thyssenkrupp-aufzuege.de 12/2013

Design, Implementierung und Test eines Analysetools in Java zur Detektion von EMS-Bus-Protokoll-Verletzungen sowie eine statistische Auswertung und Visualisierung dieser Daten

Manuel Kurfiß*, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Heutzutage werden Bussysteme in der Industrie und auch in Privathaushalten immer wichtiger. Ob man nun Fertigungsstraßen, Büroinfrastrukturen oder Heimnetze aufbaut, die Verbindung der unterschiedlichen Komponenten erfolgt über ein oder mehrere Busse. Selbst auf kleinen Komponenten wie z.B. auf Mikrokontrollern ist die Verbindung vom Speicher zum Prozessor über einen Bus realisiert. Jedes Bussystem hat Spezifikationen, welche die Grundlage der Kommunikation darstellen. Diese Spezifikationen beschreiben die Zugriffsreihenfolge der Komponenten und den zeitlichen Ablauf der Kommunikation.

In dieser Arbeit wird der EMS-Bus untersucht (siehe Abbildung 1). Der EMS-Bus ist ein proprietärer Bus der Firma Bosch. Es handelt sich um ein Zweidraht System. Um die Funktionalität des EMS-Busses überprüfen zu können, wird ein Analyse-Tool benötigt, welches in

eine bestehende JAVA API integriert wird, die Nachrichten erstellen, senden und empfangen kann. Der EMS-Analyzer soll die Anforderungen des Busprotokolls in Echtzeit auf deren Einhaltung überprüfen. Im Anschluss sollen die aufgetretenen Protokollverletzungen gespeichert, ausgewertet und ausgegeben werden. Um die Vergleichbarkeit zu steigern, sollen Randbedingungen wie die Dauer des Tests und die Gesamtanzahl der Nachrichten, ebenfalls erfasst werden. Der gesamte Quellcode wird in Java geschrieben. Um den Code testen zu können werden JUnit-Tests erstellt. JUnit ist ein Framework zum Testen von Java-Programmen [1]. Diese Tests werden im gleichen Projekt wie der zu testende Code implementiert. Um bei späteren Änderungen die korrekte Funktion gewährleisten zu können, müssen alle Tests erfolgreich sein.

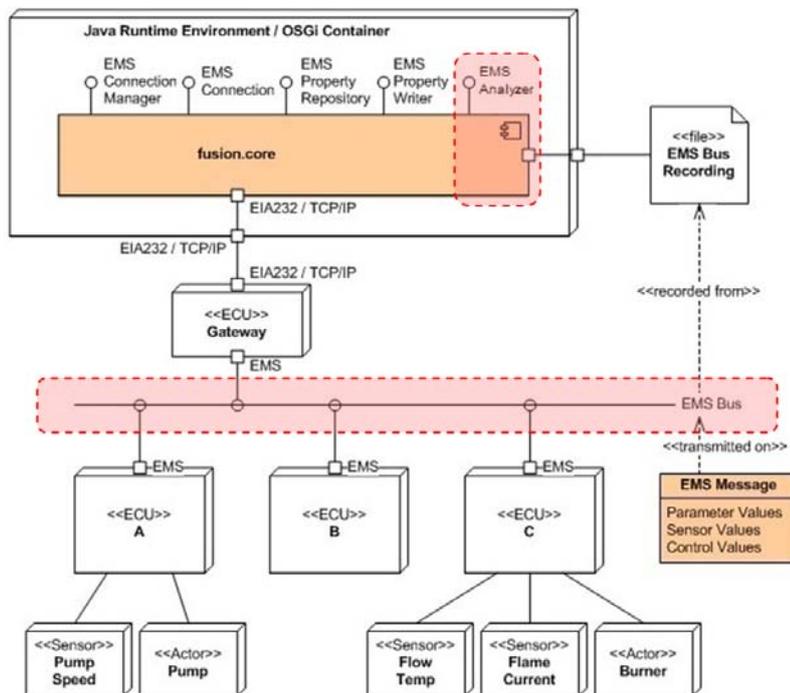


Abbildung 1: EMS Application Layer Schema

*Diese Arbeit wurde durchgeführt bei der Firma Bosch Thermotechnik GmbH, Wernau

Ein Hauptthema ist das Erstellen eines Software-Entwurfsmusters des Analyzers. Hierfür wird UML 2 als Grundlage verwendet, denn UML 2 dient zur Modellierung, Dokumentation, Spezifizierung, und Visualisierung komplexer Softwaresysteme [2]. Im ersten Schritt wird ein Strukturdiagramm benötigt. Hierfür wird das Klassendiagramm des Analyzers erstellt (siehe Abbildung 2). In diesem Diagramm wird die Klassenarchitektur dargestellt. In dieser Darstellung wird veranschaulicht, welche Klasse wie von der anderen abhängt. Die Klasse „EMSAnalysisResult“ stellt hier allerdings einen Sonderfall dar, sie wird von mehreren anderen Klassen verwendet, siehe blaue Pfeile, somit ist eine eindeutige Zuordnung nicht möglich. Des Weiteren werden Verhaltensdiagramme benötigt. Diese werden durch Sequenzdiagramme realisiert, welche den Informationsaustausch der verschiedenen Komponenten zueinander, mit deren zeitlichen und logischen Ablaufbedingungen darstellen [2]. In Abbildung 3 ist der Ablauf der 2 möglichen Fälle des „MaxBusIdleTimeChecker“ dargestellt. 1. Fall: „no detection“: Die von HandleEvent übergebene Busnachricht wird mit der „analyze()“ Methode an den „MaxBusIdleTimeChecker“ übergeben und analysiert. Da die „MaxBusIdleTime“ nicht überschritten wurde wird null zurück gegeben. Der 2. Fall: „detection“: Wird die Zeit überschritten, wird ein neues Objekt von „EMSAnalysisResult“ erstellt und mit Daten gefüllt. Anschließend wird das Objekt in „EMSAnalysisResultLogger“ gespeichert.

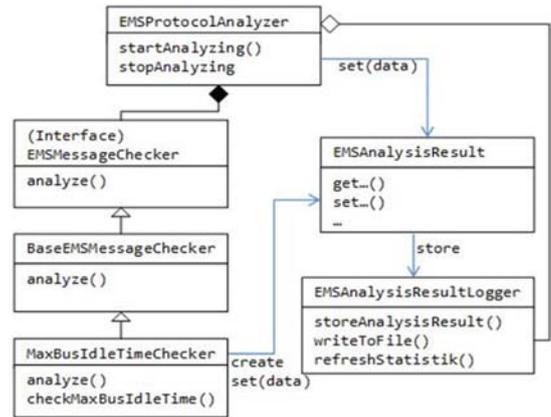


Abbildung 2: Klassendiagramm mit den wichtigsten Methoden

Nach dem Softwareentwurf wird das Modell in Java implementiert und mit JUnit Tests, die ebenfalls erstellt werden, getestet. Zunächst wird nur eine Protokollverletzung implementiert, damit das Projekt noch überschaubar ist und der Grundaufbau getestet werden kann. Läuft das Programm stabil, werden die anderen zu detektierenden Protokollverletzungen hinzugefügt, die Funktionalität getestet und deren JUnit-Tests erstellt. Werden alle Protokollverletzungen zuverlässig erkannt, werden auf der Grundlage der erkannten Verletzungen Statistiken angelegt. Von Interesse sind unter anderem welche Über-/Unterschreitungen es gab, wie oft und in welchen Abständen diese aufgetreten sind. Zur leichteren Verständlichkeit für den Anwender werden die statistischen Werte aufbereitet und in grafischen Diagrammen dargestellt.

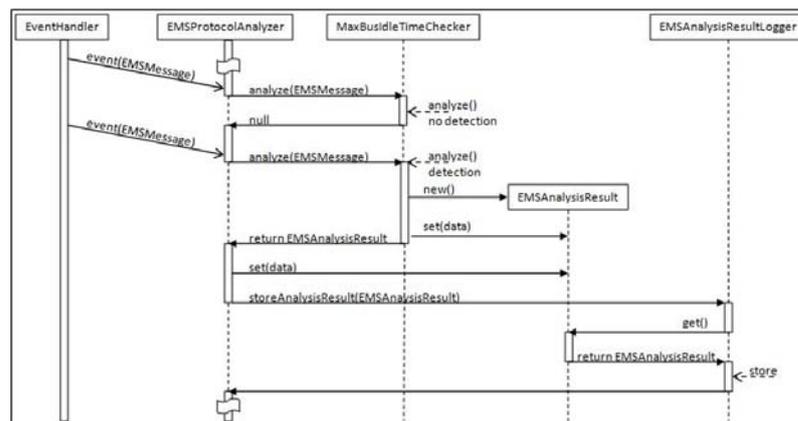


Abbildung 3: Sequenzdiagramm MaxBusIdleTime

- [1] <http://de.wikipedia.org/wiki/JUnit> (12.11.13)
 [2] UML 2 glasklar Hanser Verlag 2005, (S. 12, S. 407) ISBN 3-446-22952-3

Bildquellen:

- Abbildung 1: aus dem Bosch Intranet und ist von mir erweitert worden
- Abbildung 2-3: wurde selbst erstellt

Entwicklung einer Referenzimplementierung für plattformunabhängige Hybrid-Apps im Enterprise-Umfeld

Daniel Lombardo*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die Zukunft macht mobil und der Mobilsektor wächst und gedeiht, und das wie noch nie zuvor. Einer Studie der Marktforschungsgesellschaft IDC zufolge wurden im dritten Quartal des Jahres 2013 258,4 Millionen Smartphones verkauft, über 20 Millionen mehr als im Quartal zuvor und somit ein neuer Rekordwert [1]. Nicht nur Privatanwender, sondern auch Unternehmen möchten einen Nutzen aus dieser aufblühenden Mobilität ziehen, aus diesem Grund wurde das Konzept namens „Bring your own device“ (BYOD) geboren. BYOD sieht vor, dass Mitarbeiter ihre eigenen mobilen Geräte nicht nur privat sondern auch für die Arbeit im Unternehmen benutzen können. Einen wichtigen Aspekt, der in Verbindung mit dem BYOD-Konzept erwähnt werden muss, ist die Plattformunabhängigkeit von Apps. Plattformen gibt es mit iOS, Android, Windows Phone und BlackBerry zur Genüge. Die Entwicklung dieser nativen Apps erfordert meist viel Zeit und Geld, da sie für jede Plattform neu entwickelt werden müssen.

Das JavaScript-Framework Apache Cordova, besser bekannt als PhoneGap, bietet die Möglichkeit hybride Apps mit Technologien wie HTML5, CSS3 und JavaScript zu entwickeln. Auch plattformspezifische Funktionen wie zum Beispiel das Ansprechen von Hardwarekomponenten wie Kamera und Kompass kann durch die Cordova API in die App implementiert

werden. Des Weiteren verpackt Cordova die App in einen nativen Container, das bedeutet, die App kann wie eine übliche native App auf nahezu allen Plattformen installiert und über einen Appstore vertrieben werden.

HTML5 eröffnet viele neue Möglichkeiten und hat eine Vielzahl von JavaScript-Frameworks im Schlepptau, die die Entwicklung mobiler HTML5-Apps, wie die hybride Smart.I-App, erleichtern. Es stellt sich nun die Frage: Können die hybriden Apps bereits den Konkurrenzkampf mit den nativen Apps aufnehmen oder sind ihre Schwachstellen hinsichtlich Sicherheit und Performance zu groß?

Auch die Abteilung CI/AFU3 der Robert Bosch GmbH beschäftigt sich seit einiger Zeit mit dieser Frage. Die von ihr entwickelte Smart.I-App zur Einsicht von firmeninternen Schulungsunterlagen kann ganz im Sinne des BYOD-Konzepts auf privaten Geräten verwendet werden. Die App wurde bereits als native Version sowohl für die mobile Plattform iOS als auch Android realisiert und soll nun auch als hybride App umgesetzt werden, um als Referenzimplementierung für weitere hybride Apps zu dienen. Durch das einmalige Entwickeln einer plattformunabhängigen Hybrid-App setzt sich das Unternehmen das wirtschaftliche Ziel einer schnelleren Entwicklung bei geringeren Kosten.



Abbildung 1: Der Wunsch des Unternehmens: Plattformunabhängigkeit.

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

[1] <http://www.idc.com/getdoc.jsp?containerId=prUS2441801312/2013>

Bildquellen: Eigene Darstellung – Logos gehören den jeweiligen Inhabern

Konzeption eines Embedded Systems für die Rückspeisung der Bremsenergie eines Aufzugs über einen Frequenzumrichter sowie Realisierung von Regelungsfunktionen mit Generierung des Codes aus Simulink für einen Cortex-M4F-Mikrocontroller einschließlich Aufbau eines Prototyps und Durchführung von Messungen

Max Mairle*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die Antriebe moderner Aufzüge werden mit einer dreiphasigen Wechselspannung betrieben, welche von einem – dem Motor vorgeschalteten – Frequenzumrichter in Frequenz und Amplitude moduliert wird. Hierzu wird die eingangsseitige Wechselspannung zunächst gleichgerichtet und anschließend über einen Wechselrichter wieder zu drei einzelnen Phasen moduliert. Zwischen Gleich- und Wechselrichter befindet sich der sogenannte Zwischenkreis, welcher, ähnlich einem großen Kondensator, als Energiepuffer dient (siehe Abb. 1).

Soll ein fahrender Aufzug nun abgebremst werden, geschieht dies nicht über eine mechanische Bremse, sondern ebenfalls über Motor und Frequenzumrichter. Die kinetische Energie des Aufzugssystems wird über den Motor, welcher in diesem Fall als Generator betrieben wird, in elektrische Energie umgewandelt, was folglich die Spannung des Zwischenkreises erhöht. Um diesen vor der Zerstörung durch

Überspannung zu bewahren, muss die überschüssige Energie abgegeben werden. Klassischerweise geschieht dies über einen Abtakt-Widerstand, der die Energie in nicht weiter nutzbare Wärme umwandelt. Selbstverständlich ist dies, gerade bei größeren Aufzugsanlagen, weder ökonomisch noch ökologisch sinnvoll. Eine mögliche Alternative ist die Rückspeisung der überschüssigen Energie über einen weiteren Wechselrichter in das Stromnetz.

Dieses Szenario dient als Ausgangspunkt für die vorliegende Bachelorarbeit, in deren Rahmen ein eingebettetes System auf Basis einer modernen Prozessorarchitektur konzipiert wurde. Der hierbei entstandene Prototyp übernimmt die regelungstechnische Verarbeitung der für die Rückspeisung notwendigen Signale sowie die pulsweitenmodulierte Ansteuerung der drei Halbbrücken des netzseitigen Wechselrichters.

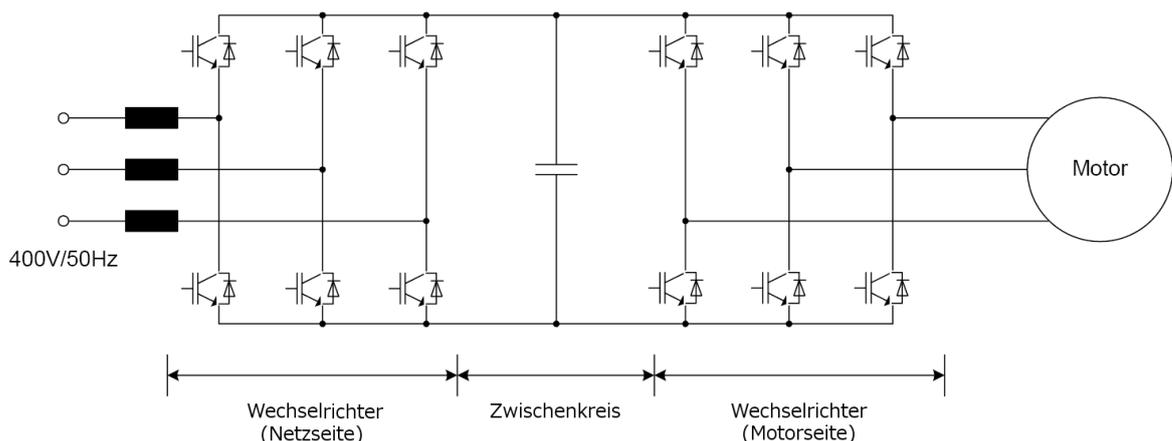


Abbildung 1: Schema rückspeisefähiger Frequenzumrichter

Als hardwareseitige Basis hierzu dient die Peripherie einer bereits vorhandenen Rückspeiseeinheit, welche nach einigen Ände-

runge direkt mit dem neuen Mikrocontroller verbunden werden konnte.

*Diese Arbeit wurde durchgeführt bei der Firma ThyssenKrupp Elevator Innovation GmbH, Neuhausen

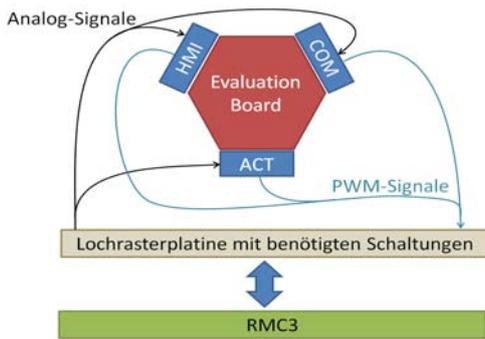


Abbildung 2: Aufbau Prototyp

Die wichtigsten Messgrößen des verwendeten Regelalgorithmus sind die Spannungen und Ströme der drei Phasen des Stromnetzes sowie die Spannung des Zwischenkreises, da die Regelung des äußersten Reglers auf einen Sollwert eben dieser Zwischenkreisspannung erfolgt. Diesem unterlagert ist ein Stromregler, der wiederum die Space Vector Modulation überlagert. Diese gibt die nötigen sechs Werte für die pulswidenmodulierte Ansteuerung der IGBT-Treiber aus, welche ihrerseits die Halbleiterschalter der Halbbrücken des Wechselrichters öffnen und schließen. Wichtig für die korrekte Regelung ist hierbei die Kenntnis des Netzwinkels. Dieser wird mit einer Phase-Locked-Loop aus den d/q-transformierten Spannungen der Phasen des Netzes berechnet und bezeichnet die aktuelle Position des Systems innerhalb einer Periode des Stromnetzes. Hieraus sind die aktuellen Werte und Verhältnisse der drei Phasen zueinander bestimmbar, was erst das zeitlich korrekte Schalten der Halbbrücken und somit letztendlich die Rückspeisung ermöglicht.

Für die Realisierung der Aufgabenstellung musste zunächst ein geeigneter Mikro-

controller ermittelt werden, der alle nötigen Voraussetzungen erfüllt. Zu diesen zählen neben technischen Anforderungen – wie diversen Schnittstellen und Hardwaremerkmalen – auch Rahmenbedingungen – wie der Preis, Garantien zur Langzeitverfügbarkeit der Komponente oder die Unterstützung des spezifischen Mikrocontrollers durch Entwicklungswerkzeuge. Auf Seiten der technischen Eigenschaften war – im Rahmen dieser Bachelorarbeit – vor allem die Anzahl und Qualität der Analog-Digital-Wandler sowie der – für die Erzeugung der PWM-Signale verantwortliche – Timer wichtig.

Nachdem mit dem XMC4500 von Infineon ein geeigneter Mikrocontroller gefunden werden konnte, wurden die für seine Programmierung zur Verfügung stehenden Entwicklungsmethoden miteinander verglichen und die in diesem Projekt am besten passende für die Implementierung ausgewählt. Neben der klassischen Mikrocontroller-Programmierung in C wurden als Werkzeuge Simulink von MathWorks und Infineons eigene, Eclipse basierte, Entwicklungsumgebung DAVE 3 untersucht, welche einen modellbasierten Ansatz zur Hardware-Konfiguration verfolgt.

Die Wahl fiel schlussendlich auf eine Kombination aus handgeschriebenem C-Code zur Konfiguration und direkter Interaktion mit der Hardware und Simulink. Letzteres Tool eignet sich optimal für die Modellierung der regelungstechnischen Aspekte der Software und ermöglicht einen ausführlichen, hardwareunabhängigen Test der Komponenten mit Hilfe von Simulationen. Abschließend wurde der für die Zielhardware benötigte Code generiert und in den manuell erstellten C-Rahmen des Programmes eingefügt.

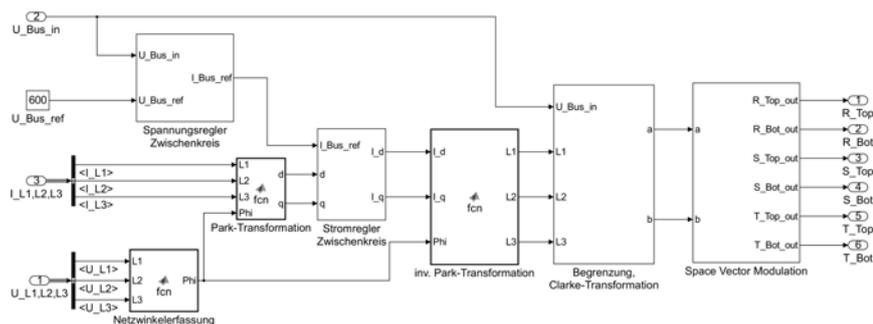


Abbildung 3: Blockschaltbild der Regelung

Bildquellen:

- Abbildung 1: Nach Benjamin Brandes. Simulation, Aufbau und Inbetriebnahme eines Hardware-in-the-Loop Systems zum Test von Frequenzumrichtern für Aufzugsanlagen, Bachelorarbeit, Hochschule Karlsruhe, Technik und Wirtschaft, 2010.
- Abbildung 2 und 3: Eigene Darstellung

Konzeption, Umsetzung und Verifikation eines Wireless Local Area Network (WLAN) Systems zur drahtlosen Übertragung von Mess- sowie Applikationsdaten zwischen Fahrzeug und Kommandostand im Motorsport

Marc Miller*, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die Betriebsabläufe auf der Rennstrecke und in der Box während eines Rennens im Motorsport sind entscheidend über Sieg oder Niederlage eines Teams. Neben der Formel 1 sind vor allem Langstreckenrennen wie das 24 Stunden Rennen von Le Mans eine der größten Herausforderungen im Rennsport. Dabei versuchen die Fahrer sich stets an der physikalischen Grenze mit dem Fahrzeug zu bewegen. Die Mechaniker und Ingenieure sind darauf getrimmt alle Abläufe so schnell und reibungslos wie möglich durchzuführen. Ein bisheriger zeitlicher Engpass während des Boxenstopps war das Auslesen von Sensordaten sowie das Konfigurieren von Fahrzeugparametern. Dazu musste bei stehendem Fahrzeug eine kabelgebundene Kommunikationsverbindung (siehe Abbildung 1 und 2)

hergestellt oder ein physikalisches Speichermedium gewechselt werden. Dies kann dazu führen, dass die konventionellen Abläufe beim Boxenstopp behindert oder eventuell sogar zeitlich verzögert werden.

Eine Verbesserung dieses Vorgangs wurde mithilfe der etablierten WLAN Technologie (IEEE 802.11) durchgeführt, um bereits während der Fahrt in der Boxengasse eine Datenverbindung zum Fahrzeug aufbauen zu können. Diese Datenverbindung kann dann bis zur Weiterfahrt des Fahrzeugs nach dem Boxenstopp bestehen. Um dies zu realisieren wurde eine Motorsport taugliche Hardware zur drahtlosen Kommunikation zwischen Fahrzeug und PC in der Box (siehe Abbildung 3) ausgewählt und ausgiebig getestet.



Abbildung 1: Eingriff während eines Boxenstopps auf der Beifahrerseite

Neben informationstechnischen Tests auf der Rennstrecke wurden verschiedene Umweltsimulationen zur Verifikation der Rennsporttauglichkeit durchgeführt. Diese Simulationen umfassten Vibrationstests auf

dem Rütteltisch, Temperaturprüfungen im Klimaschrank sowie elektronische Stresstests, wie sie in der Steuergeräteentwicklung der Automobilbranche wiederzufinden sind. Über alle Prüfungen hinweg wurde eine kontinuier-

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

liche Datenkommunikation auf der LAN und WLAN Schnittstelle der Hardware aufgebaut und aufgezeichnet. Diese Aufzeichnungen stellten sicher auch nach den Testläufen mögliche Fehlfunktionen rekonstruieren zu können. Durch gezielte Modifikationen der Leiterplatte konnten alle Tests erfolgreich und ohne Beeinträchtigung der Hardware durchgeführt werden.

Ein erstaunlicher Erfolg ergab sich bei den Reichweitentests der drahtlosen Kommunikationsverbindung beim Training eines Langstreckenrennens (VLN) am Nürburgring. Hierbei konnte mit der ausgewählten Hardware bereits bei einer Entfernung von 250 Metern des Fahrzeugs in der Boxengasse per WLAN kommuniziert werden. Selbst bei der Vorbeifahrt des Rennwagens mit Tempo 230 Km/h auf der Zielgeraden konnte eine Datenverbindung für mehrere Sekunden aufgebaut werden. Dieser Test zeigte wie performant ein WLAN System unter Verwendung von geeigneter Hardware und der richtigen Konfiguration sein kann.

Ein weiterer Teil dieser Arbeit war die Gestaltung und Umsetzung eines benutzerfreundlichen Konfigurationsassistenten in Software (siehe Abbildung 4). Diese Anwendungssoftware ermöglicht dem Kunden die Konfigurationsdaten der WLAN Hardware über eine grafische Benutzeroberfläche vom Gerät zu laden, abzuändern und erneut aufzuspielen. Bei dieser Lösung wurde auf eine Secure Shell (SSH) Kommunikationsverbindung zur Hardware zurückgegriffen, über welche die Konfigurationsdaten übertragen als auch Systemkommandos ausgeführt werden können.

Da die Möglichkeit des drahtlosen Applizierens auch sicherheitskritische Aspekte mit sich bringt, wurden Konzepte erarbeitet um Einschränkungen bei der Datenkommunikation vornehmen zu können. Ein Ansatz bestand darin, mit Hilfe einer Firewall im Systemverbund ausschließlich Protokolle zum Messdatentransfer zuzulassen.

Durch die Kombination der Motorsport tauglichen WLAN Hardware und der entwickelten Anwendungssoftware kann dieses

Gesamtkonzept in Zukunft in das Produktportfolio von Bosch Motorsport aufgenommen werden [1].

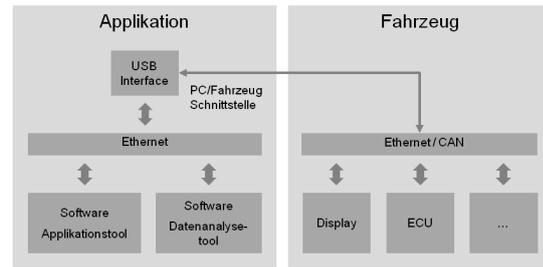


Abbildung 2: Bisherige drahtgebundene Systemkonfiguration

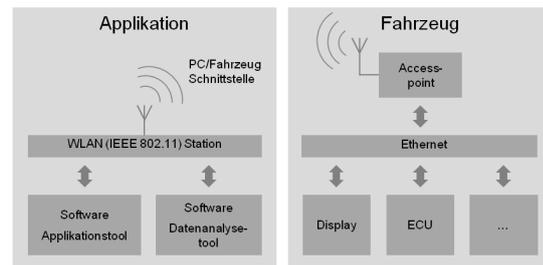


Abbildung 3: Zukünftige drahtlose Systemkonfiguration

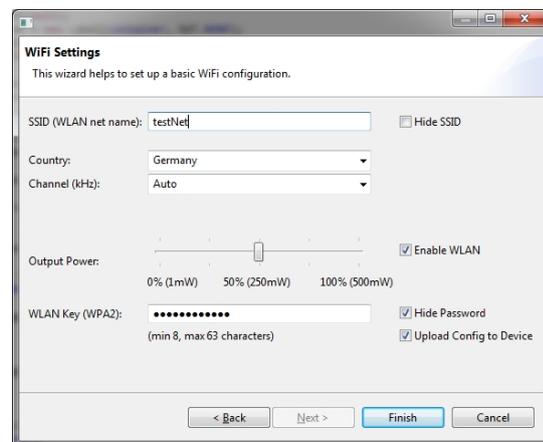


Abbildung 4: Software Einrichtungsassistent für die WLAN Hardware

[1] Bosch Engineering GmbH

Bildquellen:

Abbildung 1-4: Eigene Abbildung

Entwicklung und Integration eines ServiceFollowing-Systems zwischen DAB- und Webradio innerhalb eines Fahrzeugdemonstrationssystems unter Verwendung einer Servicedatenbank

Marc-Michael Müller*, Reinhard Schmidt, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Fahrzeug-Connectivität ist mittlerweile ein wichtiger Punkt in der Automobilentwicklung. Heutige Ansprüche an technische Geräte werden immer wichtiger und rücken mehr und mehr ins Zentrum des modernen Lebens. Logische Konsequenz daraus ist das moderne Fahrzeuge diesem Beispiel folgen und in Zukunft ebenfalls eine mobile Breitbandverbindung zur Erweiterung der medialen Verfügbarkeit im Automobil implementieren werden. Die Idee hinter dem zu entwickelnden System war die Verfügbarkeit eines Radiosenders auch über die Empfangsgrenzen von DAB hinaus per Internet/Webradio zu erweitern. So könnten zum Beispiel im Ausland auch lokale Sender aus der Heimat angehört werden und kurzfristiges wegbrechen des DAB-Signals, über schnelles Umschalten zu Webradio und wieder zurück, überbrückt werden. Auch Radiosender die ihren Service bislang nur im Internet bereitstellen halten so Einzug ins Fahrzeug.

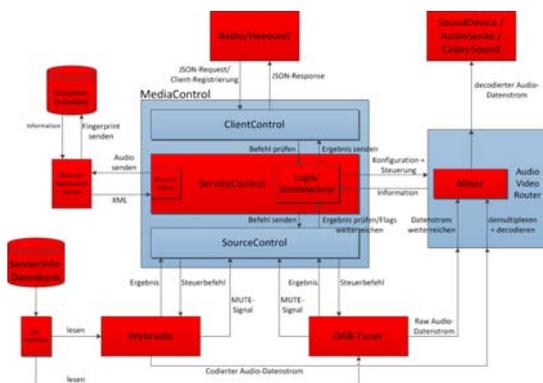


Abbildung 1: Aufbau des Systems

Wichtige Informationen zu einem Radiosender wie Name, Service Id, Logo, Frequenz, und Streaming Url werden hierbei von einer lokalen, im Fahrzeug mitgeführten Servicedatenbank der zentralen Steuerlogik zur Auswertung und Verarbeitung bereitgestellt. So lassen sich zusätzliche Informationen an

der Headunit anzeigen. An das System angeschlossene Clients (Headunit, Smartphones, Tablets) kommunizieren hierbei über JSON-Requests und Responses mit einem multithreaded TCP-Server der enthaltene Steuerkommandos an die entsprechenden Controlereinheiten weiterleitet und das Ergebnis der Aktion zurücküberträgt. Die Umschaltlogik zwischen DAB und Webstream lässt sich so usergesteuert oder/und nach einer bereitgestellten automatischen Steuerlogik realisieren. Hierbei werden von den Signalquellen Empfangsflags bereitgestellt die der ServiceController auswertet und folglich ein Umschalten antrigert oder nicht. Eine Herausforderung hierbei war die rechtzeitige Bereitstellung von geeigneten Signalketten. Während bei DAB Nutzdaten in raw PCM-Samples geliefert werden, muss bei Webstreams noch Demultiplexer und Decoder dazwischengeschaltet werden. Die verschiedenen im Internet vorkommenden Codierungen (Mp3, Ogg, ..) stellten ebenfalls eine Schwierigkeit dar.

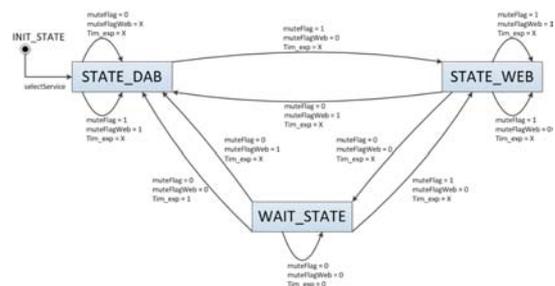


Abbildung 2: StateMachine der Umschaltlogik

Nötige Module wurden mit dem C++-Framework Qt in der Version 5.1 entwickelt und innerhalb eines Fahrzeugdemonstrationssystem zu Präsentations- und Testzwecken integriert. Wichtig war hier Rücksicht auf Erweiterbarkeit, Wartbarkeit und Stabilität zu nehmen.

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

[1] <http://de.wikipedia.org/wiki/Internetradio> 12/2013
 [2] http://de.wikipedia.org/wiki/Digital_Audio_Broadcasting 12/2013

Entwicklung eines hardwareunabhängigen WLAN-Stacks für den Einsatz in einem CAN Gateway zur Aufzeichnung von Diagnose- und Statusinformationen

Miguel Oesterlein*, Werner Zimmermann, Nikolaus Kappen

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

In der heutigen Zeit wird die Vernetzung von Automobilsteuergeräten durch unterschiedliche Bussysteme sichergestellt, beispielsweise CAN und LIN. Für den Austausch von Nachrichten aus getrennten Subsystemen ist der Einsatz von Gateways unabdingbar. Sie übertragen Daten zwischen verschiedenen Protokollen und ermöglichen

somit die Kommunikation in einem Mehrbus-system. Diese Bachelorarbeit beschreibt die Funktionsweise eines Gateways, das CAN-Botschaften bidirektional über ein WLAN-Funkmodul sendet und verarbeitet. Abbildung 1 beschreibt solch einen Aufbau.

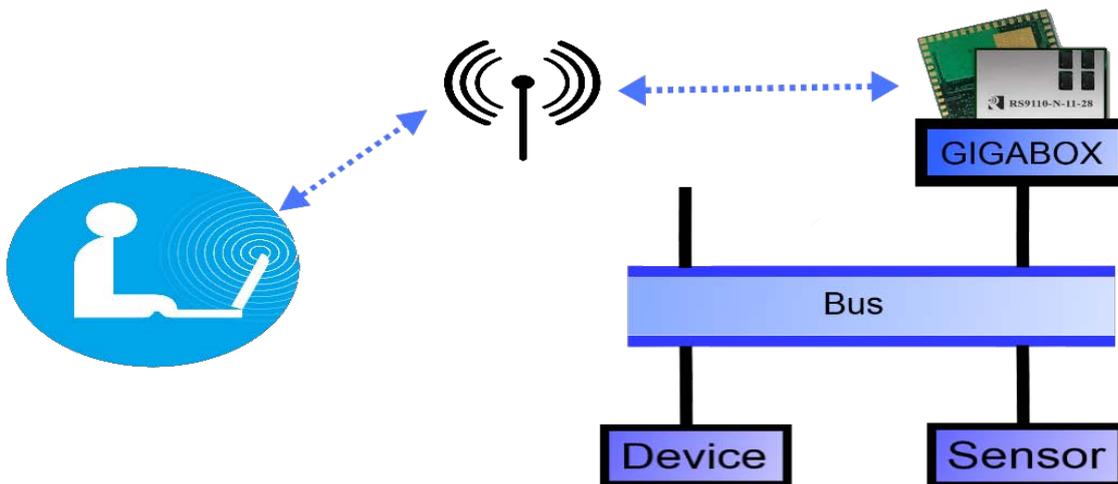


Abbildung 1: Konzeptueller Aufbau

Der Trend zu Funksystemen in der Automobilindustrie wächst stetig. Anwendungen wie Car2X setzen Funktechnologien zum Aufbau eines Fahrzeug-Ad-Hoc-Netzwerks ein und ermöglichen somit die Kommunikation zwischen bewegten Objekten. Mit WLAN können Fahrzeuge aufgrund der hohen Reichweite und der selbst aufbauenden Netzstrukturen wichtige Informationen untereinander austauschen. Ein oft beschriebenes Szenario ist die Erkennung eines Unfalls oder eines Staus durch ein vorausfahrendes Fahrzeug. Hier wird die aktuelle Situation über ein sensorbasiertes Steuergerät erkannt und die Daten über einen Bus an andere Teilnehmer des Systems weitergeleitet. Die Informationen

werden von einem Gateway empfangen und über das Funkmodul an andere in der Nähe befindlichen Fahrzeuge weitergeleitet. Jedes Fahrzeug, das mit dem Netzwerk verbunden ist, kann somit auf das Ereignis entsprechend reagieren und andere Teilnehmer des Straßenverkehrs informieren [1].

Das Ziel dieser Bachelorarbeit ist der Aufbau eines WLAN-Gateways, welches Daten von unterschiedlichen Bussystemen aufnimmt und mit technisch bedingten Latenzen über das WLAN Protokoll IEEE 802.11 an ein Endgerät (PC) weiterleitet. Am Endgerät soll eine Software zum Einsatz kommen, über die der Anwender das zu analysierende Bussystem

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

auswählt und durch Vorgabe eines Filters, spezifische Daten protokollieren kann.

Die Firma GIGATRONIK ist als Entwicklungspartner für unterschiedliche OEMs und Zulieferer tätig. Ein eigens entwickeltes prototypisches Steuergerät wird zur Konzeptentwicklung und zur Diagnose beim Kunden genutzt. Die GIGABOX ist solch ein Steuergerät und findet durch die Vielzahl von Standard-Schnittstellen vermehrt den Einsatz in der Diagnose von Produkten oder Fahrzeugen. Durch eine Skript-Sprache, die speziell für die GIGABOX entwickelt wurde, wird dem Kunden eine flexible und schnelle Lösung zur Realisierung von Applikationsanforderungen geboten. Für die Realisierung der Bachelorarbeit wird eine serienreife GIGABOX gate mit einem WLAN Modul erweitert, um so den Datenaustausch zwischen einem System und einem Endgerät zu ermöglichen. Zu den Aufgaben der Bachelorarbeit gehören:

- Aufbau eines Hardware unabhängiger Software WLAN Stacks
- GIGABOX Skript-Sprache mit neuen Funktionen erweitern
- Tests an einem Test- und Livesystem
- Implementierung einer PC Software zur Konfiguration der Hardware



Abbildung 2: GIGABOX gate FR und GIGABOX gate S

[1] <http://www.ingenieur.de/Themen/Messen/Assistenzsysteme-Car2Car-im-Fokus>
http://de.wikipedia.org/wiki/Car2Car_Communication
 (Stand: 19.11.2013)

Bildquellen:

- Abbildung 1: Eigene Darstellung
- Abbildung 2: Gigatronik

Web basierte Datenerfassung mittels RFID-Funkstrecke

Thomas Pfeiffer*, Reiner Marchthaler, Werner Zimmermann

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Maschinen werden in Zukunft immer intelligenter. Einhergehend mit dieser Entwicklung werden Daten z.B. über Laufzeiten und Fehler in digitaler Form in der Elektronik dieser Maschinen erfasst. Das Ziel der Abschlussarbeit ist ein drahtloses Auslesen dieser Daten mittels RFID im Bereich von Elektrowerkzeugen.

Die Anwendung der entwickelten Technik ist sehr vielseitig. Vorteile ergeben sich insbesondere für eine schnelle und exaktere Fehleranalyse in Entwicklung und Kundenservice. Zudem besitzt jede Maschine eine statische Seriennummer. Updates sind einfach und kostengünstig. Der Hersteller bekommt zudem Informationen über die Nutzung seiner Maschinen.

Die Arbeit besteht aus der Entwicklung und Implementierung folgender Anwendungen:

- WEB Anwendung zur Interaktion mit dem Benutzer und Analyse, Auswertung, Datenhaltung der ausgelesenen Daten. Eine weitere Funktion der WEB-Applikation ist die Kommunikation zur lokalen Interfacesoftware.
- Lokale Interfacesoftware, welche als Verbindungsstrecke zwischen dem RFID Reader und der Web Applikation fungiert. Hierfür wurden verschiedene RFID Schreib-/Lesegeräte erprobt. Die Kommunikation der Verbindung zum RFID Transceiver wurde mit USB realisiert.

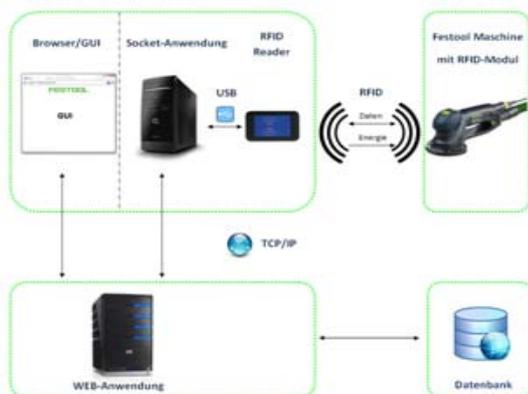


Abbildung 1: Aufbau der Anwendung

RFID ist die Abkürzung für „Radio Frequency Identification“. Es handelt sich hierbei um ein kontaktloses Identifikations- und Datenübertragungssystem, welches mit Funk arbeitet. Der Datenaustausch erfolgt durch magnetische oder elektrische Felder. Die Kommunikation bei RFID-Systemen findet zwischen dem Transponder in Abbildung 2 und einem Lese-/Schreibgerät statt. Es gibt verschiedene Arten von RFID, welche sich ausschließlich durch das genutzte Frequenzband, Speicherkapazität und Reichweite unterscheiden.

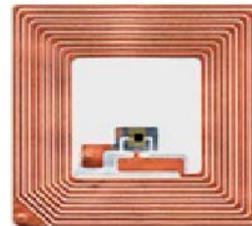


Abbildung 2: RFID-Tag mit Antenne

Die vielseitige Einsatzmöglichkeit von RFID hat mehrere Vorteile/Eigenschaften. Bei der Herstellung der Transponder erhält jeder RFID-Chip eine statische und einmalige Seriennummer. Somit kann der Transponder eindeutig zugeordnet und rückverfolgt werden.

In der Regel besteht der Transponder aus einem Mikrochip mit integriertem Speicher und einer Antenne zum Datenaustausch. Der Speicher umfasst einen EEPROM-Baustein für variable Daten, sowie einen ROM-Baustein für statische Daten.

*Diese Arbeit wurde durchgeführt bei der Firma Festool Group GmbH & Co. KG, Wendlingen a.N.

[1] wikipedia.org/wiki/RFID 12/2013

Bildquellen:

- Abbildung 1: Pfeiffer Thomas
- Abbildung 2: iwiki.de/wiki/index.php/Radio_Frequency_Identification 12/2013

Entwurf und Implementierung eines Protokolls zur Auslagerung von Steuergeräte-Funktionalitäten auf einen PC unter Einsatz von MATLAB/Simulink-Modellen

Peter Plötze*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Im Rahmen der vorliegenden Arbeit wurde eine Schnittstelle zu einem Steuergerät entwickelt, die es ermöglicht, auf dessen Funktionen über MATLAB/Simulink zuzugreifen. Ein solches Szenario findet zum Beispiel bei der Vorserienentwicklung von Komponenten für die Automobilindustrie Anwendung. Ein neu entwickeltes System wird vor der praktischen Umsetzung, wenn möglich, in einer Simulationsumgebung getestet, um erste Fehler so kostengünstig wie möglich beheben zu können. Danach wird eine prototypische Umsetzung realisiert, bevor der Schritt zum endgültigen Produkt gemacht werden kann [1]. Für beide Schritte benötigt man universell einsetzbare Prototyping-Steuergeräte und Softwarepakete. Eine in der Automobilindustrie häufig verwendete und vielseitig einsetzbare Software ist MATLAB/Simulink, mit deren Hilfe sich technische Systeme modellieren und

simulieren lassen. Sehr komfortabel ist es, wenn beim Übergang von der Simulation zum Prototypen die Erzeugnisse der Simulationsphase wiederverwendet werden können.

Diese Wiederverwendbarkeit ist ein Anwendungsfall für die Schnittstelle, die im Rahmen der Arbeit in Form eines MATLAB/Simulink-Blocks umgesetzt wurde. Ein in MATLAB/Simulink modelliertes System aus der Simulationsphase kann über die entwickelte Schnittstelle mit dem Steuergerät verbunden werden und über das ebenfalls im Rahmen der Arbeit entwickelte Protokoll die digitalen und analoge Ein- und Ausgänge auslesen und steuern. Als Steuergerät kommt die derzeit noch in Entwicklung befindliche „Geräteplattform“ von P+Z Engineering GmbH zum Einsatz. Dieses Steuergerät wird unter anderem bei der Entwicklung von Prototypen in der Automobilbranche eingesetzt.



Abbildung 1: Geräteplattform

*Diese Arbeit wurde durchgeführt bei der Firma P+Z Engineering GmbH, Stuttgart

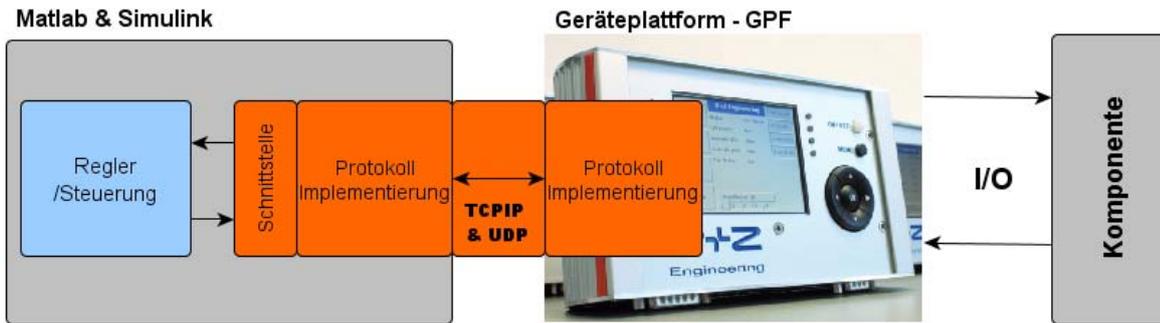


Abbildung 2: Systemumgebung

An das Protokoll wurde die Anforderung gestellt, dass es Applikationen mit harten Echtzeitanforderungen abwickeln kann und unabhängig von tieferliegenden Transportprotokollen, wie zum Beispiel CAN, UDP, TCP/IP oder RS232, definiert sein muss, um maximale Flexibilität zu gewährleisten. Das Protokoll ist an etablierten Diagnoseprotokollen aus der Automobilindustrie wie zum Beispiel UDS und XCP angelehnt und ausreichend performant ausgelegt, um in einem realistischen Szenario bestehen zu können. Das Protokoll arbeitet nach dem Client-Server-Prinzip. Dabei nimmt das Steuergerät die Rolle des Servers ein. Der Client hat neben dem lesenden und schreibenden Zugriff auf Daten des Steuergeräts auch die Möglichkeit, Funktionen auf dem Steuergerät auszuführen. Im Anschluss an die Protokollspezifikation wurde das Protokoll auf der Client- und der Server-Seite implementiert. Dafür musste für jede Seite eine auf das System abgestimmte Softwarearchitektur entwickelt werden, die die in der jeweiligen Entwicklungsumgebung vorhandenen Sprachmittel optimal ausnutzt und die gegebenen Anforderungen bezüglich Modularität und Erweiterbarkeit erfüllt. Die prototypische Umsetzung verwendet als

Transportprotokolle TCP/IP und UDP. Auf der Steuergerätsseite wurde in C implementiert und mit dem Echtzeitbetriebssystem MQX und dem FNET-Embedded-TCP/IP-Stack zur Netzwerkanbindung gearbeitet. Auf der MATLAB-Seite wurde ein objektorientierter Ansatz gewählt; die MATLAB-Instrument-Toolbox stellt die Netzwerkschicht zur Verfügung. Die größte Herausforderung auf der MATLAB-Seite bestand darin, die Echtzeitanforderungen zu erfüllen, da MATLAB ursprünglich nicht für einen solchen Anwendungsfall realisiert wurde und im Gegensatz zu MQX keine Garantien zu zeitlichen Abläufen während der Laufzeit geben kann. Es mussten daher Strukturen entwickelt werden, die eine deterministische Abarbeitung der Aufgaben des Clients ermöglichen, und es musste auf eine ereignisorientierte Auslegung des Systems zurückgegriffen werden, da sonst die vorgegebenen Latenzzeiten nicht erreicht werden konnten. Messungen zu Latenz, Datendurchsatz, Korrektheit und Vollständigkeit des Kommunikationsablaufs und eine abschließende Bewertung der Ergebnisse dienen als Entscheidungshilfe, inwiefern dieses Konzept für ein zukünftiges Produkt anwendbar ist.

[1] <http://www.automotiveit.eu/it-schlüsselrolle-fur-schnellere-entwicklungszyklen/entwicklung/id-002992-12/2013>

Bildquellen:

- Abbildung 1: P+Z Engineering
- Abbildung 2: eigene Abbildung

Integration einer automotiven Müdigkeitserkennung auf Android

Michael Popow*, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Assistenzsysteme im Automobilbereich werden immer häufiger eingesetzt, um im Straßenverkehr Unfälle zu vermeiden oder deren Auswirkungen zu reduzieren. Etablierte Systeme wie ESP und Bremsassistenten haben in der Vergangenheit bereits gute Erfolge hierbei gezeigt. Während manche Systeme unterstützend auf die Fahrdynamik wirken, setzen andere Systeme den aktuellen Zustand des Fahrers in den Fokus. Beispiel hierfür sind Assistenzsysteme zur Erkennung von Müdigkeit, insbesondere die frühzeitige Detektion des gefährlichen Sekundenschlafs.

Auf dem Gebiet der Müdigkeitserkennung werden zurzeit verschiedene Ansätze verfolgt [1]. Während zum Beispiel die direkte Beobachtung des Fahrers durch eine Kamera versucht, Müdigkeit aus dem Blinzeln des Fahrers abzuleiten, setzen andere Methoden auf die Analyse des Lenkverhaltens und errechnen daraus den aktuellen Grad der Aufmerksamkeit. Dazu werden Lenkwinkel und Lenkgeschwindigkeit über definierte Zeiträume beobachtet und in Korrelation gesetzt. Um die Müdigkeitsbewertung des Fahrers noch zu verbessern, können zusätzliche Messgrößen wie Fahrzeuggeschwindigkeit und Einfluss durch Seitenwind sowie die weiteren Bedienaktivitäten des Fahrers hinzugezogen werden. Insgesamt wird bei dieser Methode versucht, das allgemeine Fahrverhalten des Fahrers abzubilden, um besser entscheiden zu können, ob plötzliche Lenkreaktionen auf Müdigkeit oder reguläre Spurwechsel zurückzuführen sind.

Die Anwendung auf zum Beispiel einem Smart-Phone zeigt die Möglichkeit, dass der Einsatz von Fahrerassistenzsystemen sich in Zukunft nicht nur auf interne Fahrzeugelektronik beschränken muss, sondern dass man durch geeignete Technologien deren Wirkungskreis erweitern kann.

Die Firma Elektrobit Automotive hat ihren Drowsiness-Detection (engl. Müdigkeitserkennung) Algorithmus bis jetzt nur auf iOS

Geräte integriert, darum soll mit dieser Arbeit die Integration auf Android Systeme realisiert werden. Ziel der Arbeit ist es, CAN-Botschaften aus dem Fahrzeug via WiFi/UDP zu empfangen, eine Runtime-Umgebung für den Müdigkeitserkennungsalgorithmus in Android zu schaffen, sowie die entsprechenden Modifikationen am Algorithmus zu wählen, damit dieser in der neuen Umgebung fehlerfrei läuft. Eine besondere Herausforderung hierbei ist es, die empfangenen CAN-Botschaften unter Verwendung der entsprechenden Data Base CAN Bibliothek (dbc-file) in einem dynamischen Verfahren zu dekodieren und die darin enthaltenen Signale zu extrahieren. Bei den CAN-Botschaften handelt es sich um reale Messdaten, wodurch eine genaue Abbildung der aktuellen Fahrsituation sichergestellt ist. Werden dem Algorithmus alle relevanten Signale und Informationen übergeben, wird zur Laufzeit die Müdigkeit des Fahrers errechnet und entsprechend visualisiert.

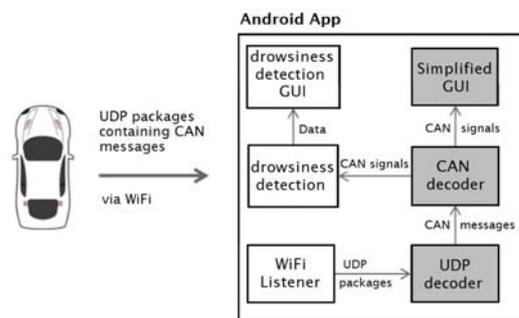


Abbildung 1: Vereinfachte Programmdarstellung

Den Programmablauf kann man anhand Abbildung 1 erkennen. Es werden zuerst die UDP Pakete mithilfe des WiFi Listener empfangen, anschließend werden einzelne Signale durch die Verarbeitung von UDP- und CAN-Decoder extrahiert und auf der GUI ausgegeben beziehungsweise an den Drowsiness-Detection Algorithmus weitergegeben.

* Diese Arbeit wurde durchgeführt bei der Firma Elektrobit Automotive GmbH, Böblingen

[1] Knipling, Ronald R. and Wierwille, Walter W. Vehicle-Based Drowsy Driver Detection: Current Status and Future Prospects. Atlanta, GA : IVHS America Fourth Annual Meeting, 1994

Ein modularer Content-Server für den Test von Connected Car Modulen mit Fokus auf Telefonie und Web

Lia Schepke*, Astrid Beck, Harald Melcher

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Seit seiner Erfindung im 19. Jahrhundert wurde das Telefon stetig weiterentwickelt. Der einfache, analoge Telefonapparat mit Handvermittlung wurde zu einer weltumspannenden Technologie. Nach der Entwicklung von drahtgebundenen Telefonen zu Mobiltelefonen war als nächster Schritt die Verbindung dieser Geräte mit dem Internet naheliegend. Das Smartphone dient heute nicht mehr nur dem Austausch von Sprach- und Textnachrichten. Vielmehr sind diese Geräte universell einsetzbare Computer, die sich als Kommunikationszentrale für die verschiedensten Dienste anbieten [1].

Um eine Vielzahl von Telefonen miteinander zu verbinden, werden sogenannte Telefonanlagen eingesetzt. Diese Anlagen ermöglichen unter anderem eine Protokollierung von Telefonaten und können mehrere Nebenstellen eines Telefonanschlusses zur Verfügung stellen. Auch hier hat sich die Technik weiterentwickelt und so sind moderne Telefonanlagen mittlerweile IP-basiert. Über Vermittlungsprotokolle, wie z.B. SIP (Session Internet Protocol), wird die Kommunikation zwischen zwei Endgeräten, z.B. Software SIP-Telefone, über Internet oder LAN ermöglicht. Damit sowohl Festnetz-, als auch Mobilfunknetze über diese modernen Telefonanlagen gesteuert werden können, werden Gateways eingesetzt, die die analogen und digitalen Signale in SIP umwandeln [2].

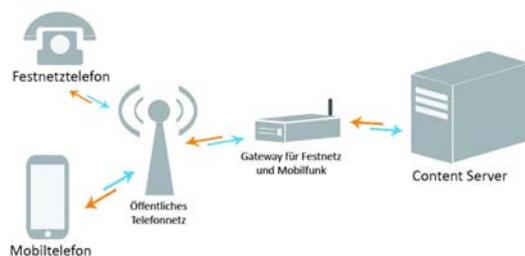


Abbildung 1: Funktionsweise eines Anrufs über den Content-Server.

Diese Bachelorarbeit befasst sich mit der Konzeptionierung und Umsetzung eines Content-Servers, der Testingenieure beim Testen unterschiedlichster Telefon-, Internet

und Infotainmentfunktionen im Fahrzeug unterstützen soll. Bisher mussten sich die Tester gegenseitig bzw. selbst anrufen oder Nachrichten schicken, um diese Funktionen zu prüfen. Daher soll nun durch den Einsatz des Content-Servers eine Optimierung und größtmögliche Automatisierung erreicht werden.

Der zukünftige Content-Server soll über die Funktionen einer universellen Telefongegenstelle hinaus auch die Möglichkeit bieten, neuartige Features aus dem Connected Car-Bereich standardisiert überprüfbar zu machen. Neue Elemente aus vernetzten Fahrzeugen könnten, z. B. die Verwendung von Apps oder das Streaming über Infotainmentsysteme sein. Darüber hinaus soll gewährleistet werden, dass der Content-Server jederzeit um neue Module erweiterbar ist, um ihn dem Wandel der Technik anzupassen.

Als Basis für den Content-Server wird ein Linux-Server verwendet, auf dem die Software FreeSWITCH läuft. FreeSWITCH bietet die Aufgaben einer Telefonanlage an und außerdem können über die Software unterschiedliche Aktionen implementiert werden, wie z.B. ein Skript für das Versenden einer Email. Für den Content-Server wird zusätzlich ein Gateway verwendet, welches in diesem Fall sowohl GSM als auch ISDN unterstützt, um den Server mit dem öffentlichen Telefonnetz zu verbinden.

Um die Administration und Konfiguration des Servers nicht nur über ein Terminal steuern zu können, sondern auch um eine Benutzerfreundlichkeit und einfache Bedienbarkeit den Administratoren zu bieten, wird dieser Server über eine eigens aufgesetzte, lokale Webseite gesteuert. Das ganze Prinzip der Webseite basiert auf LAMP (Linux, Apache, MySQL und PHP). So werden über PHP die relevanten Daten, z.B. Szenarien des Servers, dynamisch aus der Datenbank geladen und auf der Webseite dargestellt. Dadurch wird gewährleistet, dass der Inhalt der Webseite live im Server Anwendung findet.

Die Konfigurationswebseite wird außerdem dafür verwendet, Regeln für die einzelnen Anrufer aufzusetzen und die Länge und Kosten der Gespräche aufzuzeichnen. Neue Module

*Diese Arbeit wurde durchgeführt bei der Firma P3 systems GmbH, Stuttgart

können ebenfalls über einen Skriptimport hinzugefügt werden.

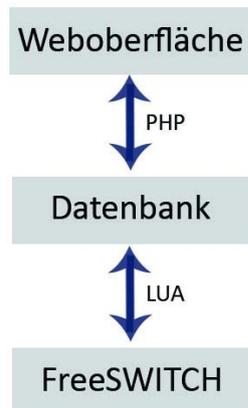


Abbildung 2: Aufbau des Content-Servers.

Ein Tester hat nun die Möglichkeit den Content-Server über die für ihn freigeschalteten Telefonnummern anzurufen und unterschiedliche Anrufe durchzuführen. Mehr als 25 Szenarien können bereits über den Content-Server per Anruf angefordert werden. Dazu gehören unter anderem folgende Aktionen:

- Automatischer Rückruf mit Abspielen einer Sounddatei

- Versenden einer SMS durch Anruf (ohne Abnehmen)
- Aufzeichnung des Telefonates und Versenden der Sounddatei in einer Email.

Diese Basis-Dienste werden variiert, um möglichst viele Testfälle abdecken zu können. Mit Hilfe der Anrufe wird festgestellt, ob diese Funktionen vom Infotainmentsystem korrekt unterstützt werden.

Über eine Online-Anbindung des Content-Servers können ebenfalls Module, die keine Telefonverbindung benötigen, von den Testern benutzt werden. Dazu zählen beispielsweise folgende Szenarien:

- Streaming (Audio & Video)
- Messenger
- Soziale Netzwerke

Durch die Umsetzung des Content-Servers im Rahmen dieser Bachelorarbeit werden sowohl die Testprozesse im Connectivity-Umfeld weiter optimiert, als auch innovative Möglichkeiten geschaffen, um Apps und Online-Features standardisiert und verlässlich zu testen.

- [1] Anatol Badach (2007,3), *Voice over IP Die Technik. Grundlagen, Protokolle, Anwendungen, Migration, Sicherheit*, Carl Hanser Verlag, München Wien
- [2] Ulrich Trick, Frank Weber (2009,4), *SIP, TCP/IP und Telekommunikationsnetze. Next Generation Networks und VoIP-konkret*, Oldenbourg Verlag, München

Bildquellen: Eigene Darstellung

Analyse charakteristischer Sensorsignale und Modellierung eines Regelkreises zur dynamischen Ansteuerung eines Aktors

Alexander Schoch*, Werner Zimmermann, Reiner Marchthaler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Moderne Technik im Innenraum von Kraftfahrzeugen hat im neuen Jahrtausend deutlich an Bedeutung gewonnen. Der technische Fortschritt treibt die Automobilindustrie an – er ermöglicht es, sich als Hersteller gegenüber den vielen Mitbewerbern in der Automobilbranche abzugrenzen und sich den fortwährend wechselnden Bedürfnissen des Kundenkreises anzupassen. Neben der größer werdenden Anzahl von Assistenz- und Infotainmentfunktionen spielt bei hochwertigen Fahrzeugen der Komfort für die Insassen eine immer bedeutendere Rolle. Mit dieser Arbeit soll ein System optimiert werden, welches das Geräuschniveau in

einem häufig gefahrenen Geschwindigkeitsbereich deutlich verbessert. Mittelpunkt der Problemstellung ist dabei die Regelung eines physikalischen Prozesses im Fahrzeuginnenraum, dessen Regelgröße eine charakteristische Form annimmt. Durch Vorverarbeitungsalgorithmen und Transformationen werden aus dem Eingangssignal wichtige Informationen gewonnen, die anschließend Regelalgorithmen zur Verfügung gestellt werden. Ein Aktor dient anschließend dazu, den Prozess nach gewünschten Kriterien zu beeinflussen, welches sich direkt auf die Regelgröße auswirkt und dadurch neue Information für die Regelung bereitstellt.

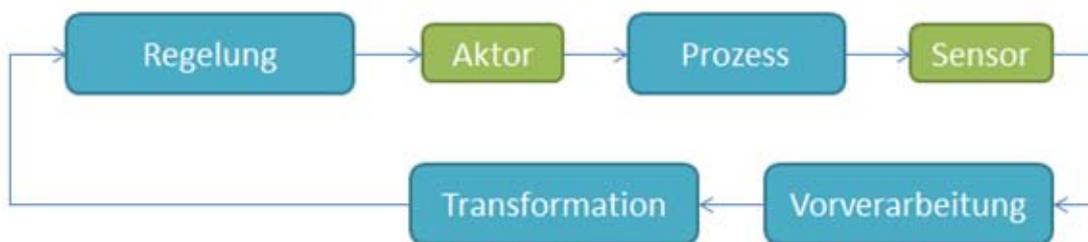


Abbildung 1: Regelkonzept

Ziel ist es, mit Hilfe einer Echtzeitanalyse der Eigenschaften des Prozesses, gewisse Charakteristika zu erkennen und diese auf den Prozess anzuwenden. Dazu dienen mechanische, sowie auch elektronische Mittel, um die Lösung des Problems zu einem geregelten System zu formen. Dabei sollen auch Rahmenbedingungen, wie beispielsweise Echtzeitfähigkeit, Robustheit gegen Störeinflüsse und Flexibilität berücksichtigt werden. Die modellbasierte Entwicklung wird mit dem Prinzip

des „Rapid Prototyping“ vorangetrieben. Mit Tools wie MATLAB/Simulink werden Modelle erstellt, welche in C-Code kompiliert werden. Der entstandene Code wird anschließend auf die Zielhardware gespielt und kann dadurch in Echtzeit ausgeführt werden. Um während des Betriebs das Verhalten des Systems überwachen zu können, kann mit der Nutzung des Real-Time-Interfaces einer Software wie dSPACEControlDesk die Zielhardware angesteuert werden.

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

Connectivity am Beispiel des Elektrofahrzeugs eScooter LiBERTé-App-Entwicklung und Entwicklung der Kommunikation zwischen eScooter und mobilem Endgerät

Valon Sejdijaj*, Astrid Beck, Andreas Rößler

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

CONNECTIVITY ALS HERAUSFORDERUNG

Die Optimierung von Elektromobilitätskonzepten hinsichtlich des effektiven Endkundennutzens gehört in der heutigen Zeitepoche zu den größten Herausforderungen der Automobilindustrie. Denn für viele Automobilhersteller ist bereits klar, dass die Mobilität für den Menschen ein großes Stück Lebensqualität bedeutet. [1] Damit auch in den nächsten Jahrzehnten diese Lebensqualität für die Gesellschaft gewährleistet bzw. verbessert werden kann, beschäftigen sich viele Unternehmen, unter anderem der Entwicklungsdienstleister Bertrandt, aktuell mit der Anbindung von mobilen Endgeräten an Elektrofahrzeuge. Die Anbindung mobiler Endgeräte mit Fahrregelsysteme, Infotainment, Navigations- und Assistenzsystem soll die Elektromobilität zukünftig weiter vorantreiben sowie ein Stück weit bereichern. So sollen die Fahrer von E-Fahrzeugen durch das Abrufen von Informationen, wie den aktuellen Ladezustand, die Restreichweite oder durch Alarmierung von kritischen Zuständen des E-Fahrzeugs, frühzeitig die Fahrt optimieren können.

In der Bertrandt Technikum GmbH entwickeln Ingenieure aus den Bereichen Entwicklung Komponenten, Elektronikentwicklung, Modell- und Fahrzeugaufbau gemeinsam mit angehenden technischen Produktdesignern an einem Projekt **LiBERTé Freiheit auf zwei Rädern**, den Elektroroller eScooter LiBERTé. Im Rahmen dieser Arbeit soll für den eScooter Prototyp eine Applikation für mobile Endgeräte auf Android Basis entwickelt werden, welches zukünftig eine Interaktion zwischen

Elektroroller und Fahrer ermöglicht. Ziel der Abschlussarbeit ist die Konzeptionierung und Entwicklung einer mobilen Applikation, mit der die Fahrer über eine Bluetooth-Verbindung zum eScooter



Abbildung 2: eScooter LiBERTé

Fahrzeugparameter wie zum Beispiel

- die aktuelle Geschwindigkeit,
- den Gesamtkilometerstand,
- die Restreichweite,
- den Akkufüllstand,
- die Motor- und Akkutemperatur

auslesen sowie die Wegfahrsperr deaktivieren, das Fernlicht ein-/ausschalten, den Fahrmodus wechseln, den eScooter in den Standby Modus setzen oder das Fahrprofil als PDF exportieren können. Für die Realisierung der Mobilen Applikation liegt folgende Idee zugrunde: alle Fahrzeugparameter sollen vom Haupt-Steuergerät mit Lade- und Rekuperationselektronik mittels eines CAN-Bluetooth-Moduls dem Endnutzer an ein mobiles Gerät übertragen und in einer übersichtlichen und einfachen Form visualisiert werden.

*Diese Arbeit wurde durchgeführt bei der Firma Bertrandt Technikum GmbH, Ehningen

[1] Bernard Bäker: Moderne Elektronik im Kraftfahrzeug V. Elektromobilität der Zukunft – Elektrische Energiespeicher für mobile Anwendungen – Prädiktive Betriebsstrategien und zugehörige E/E-Architekturen. Renningen: Expert Verlag, 2010.

Bildquellen: Intranet der Bertrandt Technikum GmbH

Konzeption und Entwicklung eines Systems zur Belegungserkennung und Reservierung von Parkplätzen

Thies Spannkebel*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die Parkplatzsuche ist in der heutigen Zeit ein alltägliches und mitunter zeitintensives Problem. Diese Entwicklung beobachtet auch das Fraunhofer Institut für Arbeitswirtschaft und Organisation IAO. Dessen Kompetenzzentrum für energetische und informationstechnische Mobilitätsschnittstellen KEIM versucht in enger Zusammenarbeit mit der Hochschule Esslingen Lösungen zum Einsatz des Smartphones als Schnittstelle in das Mobilitätssystem zu entwickeln.

Für eine effektivere Parkplatzsuche wurde ein System zur Belegungserkennung und Reservierung von Parkplätzen konzipiert und entwickelt. Dieses besteht aus zwei Kernkomponenten: die Belegungserkennung der Parkplätze (s. Abb. 1) und die Suche und Reservierung der Parkplätze mittels einer multiplattformfähigen mobilen Applikation namens Simple Parking.



Abbildung 1: Belegungserkennung

Bei den üblichen Lösungen für die Belegungserkennung von Parkplätzen werden meist Bewegungssensoren in die einzelnen Stellplätze eingelassen, was relativ teuer und aufwendig ist. Ein effizienteres Konzept stellt ein Bilderkennungssystem auf Basis von optischen Sensoren (Kameras) dar. Hier können oftmals bereits vorhandene Infrastrukturen genutzt werden. Ein solches System hat die Forschungsgruppe Real-time Computer Vision des Instituts für Neuroinformatik an der

Ruhr-Universität Bochum (RUB) entwickelt. In Kooperation mit der RUB wurde das System namens Video-based Parking Space Detection an der Hochschule Esslingen positiv getestet und als Grundlage für diese Arbeit ausgewählt [1].

Für die Suche und Reservierung von Parkplätzen wurde eine App entwickelt, die auf mehreren Plattformen lauffähig ist. Da native Apps diese Voraussetzung nicht erfüllen, wurde mit Hilfe von jQuery mobile eine multiplattformfähige Web-App auf Basis von JavaScript, HTML5 und CSS realisiert. Die Umwandlung der Web-App in eine Hybrid-App erfolgte mittels eines nativen Containers durch das Framework PhoneGap. Dadurch besteht ein uneingeschränkter Zugriff auf alle Hardwarefunktionen des Smartphones [2].

Damit der Benutzer den Belegungsstatus von Parkplätzen via mobiler App abrufen kann, wurde ein Datenbankserver eingerichtet, der mittels Webserver die nötigen Informationen für den mobilen Client bereitstellt. Der Nutzer kann daraufhin mit Hilfe der App einen freien Parkplatz zur gewünschten Uhrzeit reservieren. Das System (s. Abb. 2) erkennt automatisch, wenn der Nutzer den reservierten Parkplatz ansteuert und belegt. Es schließt daraufhin den Reservierungsvorgang ab.



Abbildung 2: Architektur des Systems

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

- [1] Tschentscher M.; Neuhausen M.: Video-based parking space detection, RUB, Bochum, 2012
 [2] Franke F.; Ippen J.: Apps mit HTML5 und CSS3, 2. Auflage, Galileo Computing, Bonn, 2013

Bildquellen:

- Abbildung 1: <http://www.ini.rub.de 12/2013>
- Abbildung 2: Eigene Darstellung

Gegenüberstellung von Client-Server-Frameworks auf der Basis von JavaEE und Node.js im Enterprise-Umfeld

Ruven Spiller*, Andreas Rößler, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

HTML5 wird aktuell als die Zukunft in der Entwicklung von Webapplikationen gehandelt. Inzwischen ist die Technologie bereits soweit fortgeschritten, dass es durchaus üblich ist HTML5 produktiv einzusetzen. Nicht zuletzt treibt der stetig wachsende mobile Markt die Verbreitung voran. Die Abschlussarbeit beschäftigt sich mit der Frage, ob HTML5 dem Anspruch der Industrie an Software in Unternehmen genügt.

Um eine qualitative Aussage über diese Thematik machen zu können, wurde bei der Cellent Finance Solutions AG in Stuttgart das vorherrschende – auf JavaEE basierende – Framework mit einer HTML5 Lösung verglichen. Konkret handelt es sich dabei um Captain Casa als etablierte Lösung in Java und AngularJS bzw. SailsJS auf Basis von NodeJS als HTML5 Kandidat. Bei beiden Lösung besteht ein Client-Server-Szenario.

Da ein Vergleich zweier grundlegend verschiedenen Technologien nicht trivial ist, wurden die Kriterien des Vergleichs auf bestimmte Aspekte, die Software in Unternehmen zu erfüllen haben, eingegrenzt:

- Infrastruktur
- Interoperabilität
- Sicherheit
- Performanz
- Support
- und weitere

Beim Thema Sicherheit veröffentlicht die non-profit Organisation OWASP (Open Web Application Security Project) jährlich eine Top-10 Liste für die gängigsten Schwachstellen in Web-Applikationen [1]. Abbildung 1 zeigt einen potenziellen Angriffsweg der genutzt werden könnte. Besonders bei den üblichen Arten von Angriffen, sollte das Framework in der Lage sein, Schutzmechanismen zur Verfügung zu stellen.

Ein weiterer Anspruch in Unternehmen ist die vorhandene Infrastruktur. Durch die Vielzahl an Anwendungen, die in Geschäftsprozesse involviert sind, muss die Software in der Lage sein, sich einfach und vor allem stabil in bestehende Soft- und Hardwarelandschaften eingliedern zu lassen.

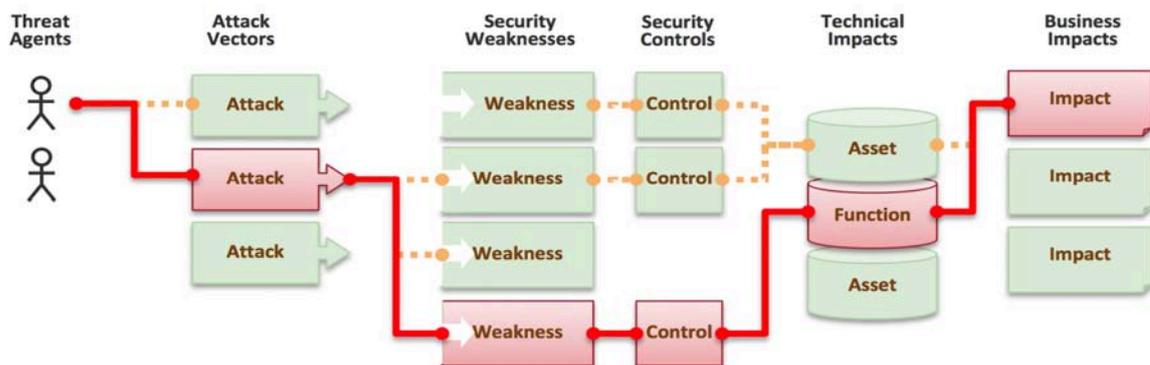


Abbildung 1: OWASP Angriffsvektoren

Das betrifft die Unterstützung von gängigen Datenbanken (bspw. Oracle, Mysql, MongoDB). Es sollte aber auch die Möglichkeit bestehen, Tabellen und Reports in das gängige Excel- oder Word-Format exportieren und importieren zu lassen.

Um eine Grundlage für den tatsächlichen Vergleich haben zu können, wurde eine

Anwendung entwickelt, die in der jeweiligen Technologie umgesetzt wird. Anhand dieser Anwendung werden Untersuchungen zur Entwicklungszeit, Performanz, Benutzerakzeptanz und weiteren Punkten durchgeführt. Als Szenario für das Programm dient eine einfache Support-Software zur Erstellung von Tickets, Generierung von Statistiken und Echt-

*Diese Arbeit wurde durchgeführt bei der Firma Cellent Finance Solutions AG, Stuttgart

zeitkommunikation in Form eines Chats. Die Anwendung ist lediglich zu Vergleichszwecken erstellt und produktiv nicht einsetzbar.

Während der Entwicklungsphase haben sich für beide Lösungen typische Vor- und Nachteile ergeben. Die Entwicklung der Oberfläche ist durch den Captain Casa Editor schnell und einfach erledigt. Dabei wird bereits auf den in Java 8 kommenden Standard JavaFX gesetzt. Auf HTML5-Seite ist durch die Verwendung von Twitter Bootstrap die Erstellung einer GUI ebenfalls zügig erledigt. Beide Frameworks bieten häufig genutzte Ein- und Ausgaben wie Textfelder, Dropdowns und Tabellen. Dabei sehen die Elemente auf Seiten von HTML5 allgemein moderner aus (vgl. Abbildung 2).

Bei der Backendentwicklung bremsen das stetige Kompilieren und deployen des Java-codes etwas aus. Die Möglichkeit des einfachen Debuggens ist jedoch sehr hilfreich bei der Fehlersuche. Bei Javascript hingegen treten viele Ungereimtheiten erst zur Laufzeit auf und es bedarf viel eher an einem sauberen Code gegenüber Java-Anwendungen.

Benötigt man allgemein Unterstützung findet man zum Thema HTML5 und Java mehr als ausreichend Quellen im Internet. Auch die einzelnen Frameworks zu HTML5 sind weit verbreitet. Captain Casa liefert hierzu ausschließlich das offizielle Forum und die Dokumentation, die qualitativ aber hochwer-

tige Informationen bereit halten.

Betrachtet man die Performanz, stehen Javascript-Anwendungen Java-Applikationen in nicht mehr viel nach. Durch das immer stärkere Aufkommen von HTML5-Apps sind die Browserhersteller dazu übergegangen ihre Javascript-Engines stetig zu optimieren um gegenüber der Konkurrenz einige Millisekunden schneller zu sein. Besonders Ansätze wie asm.js [2] könnten dazu führen, das Javascript auch in Zukunft noch weitere Optimierungen in der Ausführungsgeschwindigkeit erfährt.

Resultierend aus den Ergebnissen des Vergleichs sieht der Autor HTML5 als Frontend-Werkzeug durchaus in der Lage die Aufgaben von Unternehmenssoftware zu erfüllen. Auf Serverseite hingegen wird es schwierig mit Java in Konkurrenz zu treten. Die Anzahl der vorhandenen Bibliotheken und Pattern aus JavaEE sind sehr umfangreich und auch erprobt. NodeJS, als sehr junges System, muss an dieser Stelle noch wachsen und sich beweisen, obwohl es inzwischen auch Projekte gibt, die eine JVM (Java Virtual Machine) mit NodeJS umsetzen. So könnte man theoretische vereinzelt Java-Klassen auch weiterhin verwenden. Die Geschwindigkeit, die heutige Javascript-Interpreter bieten, reichen für typische Unternehmensaufgaben aus.

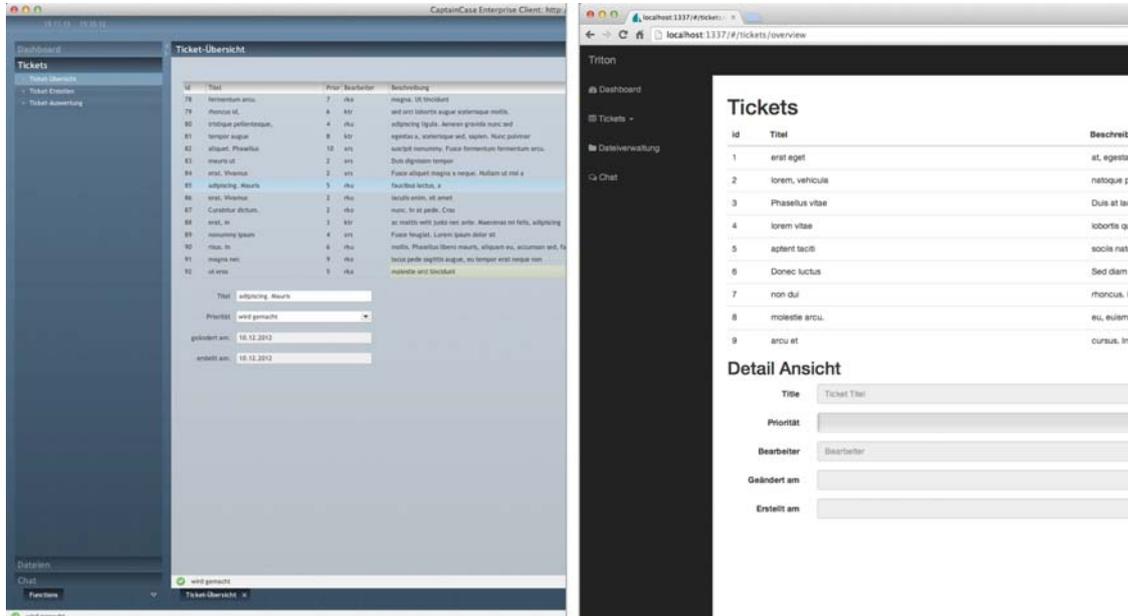


Abbildung 2: Gegenüberstellung der Oberflächen

[1] OWASP Top 10 2013

[2] <http://asmjs.org/spec/latest> 12/2013

Bildquellen:

- Abbildung 1: OWASP Top 10 2013
- Abbildung 2: Ruven Spiller 2013

Integration eines Lua-Skript-Interpreters in ein Komponentenframework zur flexiblen Erweiterung des Funktionsumfangs von Embedded-Sensoren über nachladbare Skripte

Dominik Spindler*, Manfred Dausmann, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

In umfangreichen Softwareprojekten stellt die Wiederverwendung von Programmcode ein essentielles Instrument zur Steigerung der Effizienz bei der Entwicklung von neuen Produkten dar. Neben der Entwicklungszeit und den damit verbundenen Kosten sinkt durch die Verwendung von bereits vorhandenen und geprüften Softwareteilen vor allem auch die Fehlerrate und somit der Aufwand für Softwaretests. Im konkreten Fall wird dieser sogenannte „code reuse“ durch die Verwendung eines eigens entwickelten, komponentenbasierten Frameworks sichergestellt. Zusätzlich zum komponentenbasierten Ansatz kommt zur internen Strukturierung ein Schichtenmodell zum Einsatz. Verschiedene Abstraktionsebenen erlauben einen Austausch tiefer liegender Programmteile ohne Anpassungen an anderen Komponenten vornehmen zu müssen. Dies ermöglicht zum Beispiel den Einsatz auf verschiedenen Hardwareplattformen. Abbildung 1 zeigt den schematischen Aufbau des verwendeten Framework-Stacks.

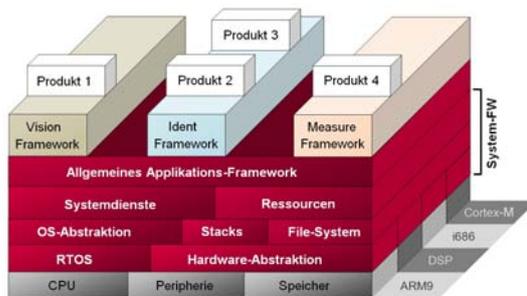


Abbildung 1: Framework-Stack

Um die Flexibilität bei der Applikationsentwicklung weiter zu steigern soll das Framework um die Möglichkeit erweitert werden Programmcode auszuführen, der in einer interpretierten Sprache verfasst wurde. Diese Bachelorarbeit beschäftigt sich aus diesem

Grund mit der Integration eines Interpreters für die Skriptsprache „Lua“ als Framework-Komponente und der Erstellung von Lua-Bibliotheken zur Anbindung an die bestehenden Komponenten des Systems.



Abbildung 2: Logo des Lua-Projekts

Lua ist eine mächtige, multiparadigmatische [1] Skriptsprache, die sich trotz ihrer Flexibilität durch eine hohe Performance und Einfachheit auszeichnet. Das Entwicklerteam legt einen besonderen Schwerpunkt auf die Möglichkeit der Einbindung des Lua-Interpreters in bestehende Software und eingebettete Systeme. Da sich die Flexibilität von Software durch den Einsatz einer eingebundenen Skriptsprache deutlich steigern lässt, kommt Lua heute auch in vielen proprietären Programmen zum Einsatz. Ein Beispiel hierfür ist „Adobe Photoshop Lightroom“, das es Nutzern ermöglicht eigene Plugins in Lua zu erstellen [2]. Die Kombination aus kompilierten und interpretierten Programmteilen erlaubt eine einfache Wiederverwendung bestehender, allgemeiner Softwaremodule für neue, spezialisierte Funktionen. Die denkbaren Einsatzmöglichkeiten im Rahmen der Bachelorarbeit sind vielfältig und reichen vom Ausbessern von Fehlern in bereits ausgelieferten Geräten bis hin zur Implementierungen von kundenspezifischen Applikationen ohne jegliche Anpassung der Basisapplikation. Die zeitnahe Verfügbarkeit solcher maßgeschneiderter Lösungen ermöglicht eine erhöhte Kundenbindung und bedeutet zugleich eine Differenzierung gegenüber den Wettbewerbsprodukten.

* Diese Arbeit wurde durchgeführt bei der Firma Leuze electronic GmbH + Co. KG, 73277 Owen

[1] <http://www.itwissen.info/definition/lexikon/Multiparadigma.html> [22.11.2013 14:48]

[2] www.lua.org/about.html [22.11.2013 15:08]

Bildquellen:

- Abbildung 1: Leuze electronic
- Abbildung 2: lua.org 12/2013

Portierung eines Mikrocontroller basierten, echtzeitfähigen Bus-Kommunikationsinterfaces auf Zielsystem PowerPC mit embedded FPGA

David Stier*, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

In einem heutigen Fahrzeug werden viele Sensoren und Komponenten benötigt, die dem Fahrer das Bedienen seines Fahrzeuges erleichtern. Dazu gehört z.B. das Lenkstockschaltermodul, das u.a. folgende Sensoren und Schalter enthält:

- Lenkwinkelsensor
- Schalter für Blinker und Scheibenwischer

Zur Sicherstellung der Qualität der Lenkstockschaltermodule werden entwicklungsbegleitende Prüfungen benötigt. Dafür werden bis zu 5 DUTs¹ gleichzeitig z.B. einer Lebensdauerprüfung unterzogen. Dabei werden die DUTs unter definierten Bedingungen in einer Klimakammer von einem Prüfroboter betätigt. Die Sensorwerte werden hierbei vom CASTLE^{IT2} ausgelesen. Die Messwerte werden mittels eines PCs mit der hauseigenen Software MP-Win³ ausgewertet.

Für die Prüfung werden die DUTs über Bussysteme wie z.B. CAN⁴ oder LIN⁵ an das CASTLE^{IT} angeschlossen. Ebenfalls können Protokolle wie z.B. ISO-TP⁶ oder KWP2000⁷ zum Einsatz kommen.

Der CAN Bus ist ein serielles Bussystem und wurde für die Automobilindustrie entwickelt. Er wurde von Bosch entwickelt und ist seit 1991 im Einsatz. Der CAN Bus ermöglicht die Übertragung von bis zu 8 Byte Nutzdaten pro Frame und wird aufgrund seiner Echtzeitfähigkeit und hohen Übertragungssicherheit bis heute verwendet[1].

Da der CAN Bus nur maximal 8 Byte Nutzdaten pro Frame unterstützt, müssen größere Datenmengen segmentiert werden, dies kann z.B. mithilfe des Protokolls ISO-TP geschehen.

Damit können Botschaften mit bis zu 4095 Byte Daten übertragen werden. Das Protokoll teilt die Nutzdaten in 8 Byte große Blöcke auf und sendet diese nacheinander. Ebenfalls ermöglicht ISO-TP eine Flusskontrolle, um die gesamte Kommunikation zu überwachen [1].

Jeder DUT verwendet ein eigenes CASTLE^{IT}. Die einzelnen CASTLE^{IT}s befinden sich in einem Rack. Das Rack versorgt die CASTLE^{IT}s mit Strom und stellt die Verbindung zum PC zur Verfügung.



Abbildung 1: Links ein PIM und rechts die Versorgungseinheit des Racks

Zu den Aufgaben des CASTLE^{IT}s gehören:

- Ermöglichung des Datenaustausches zwischen Prüfling und MP-Win
- Durchführung einer Restbussimulation
- Überprüfung der Prüflingsdaten auf Grenzwerte
- Schalten und Messen der Prüflingsversorgung mit Strom und Spannung

*Diese Arbeit wurde durchgeführt bei der Firma ITronic GmbH, Erdmannhausen

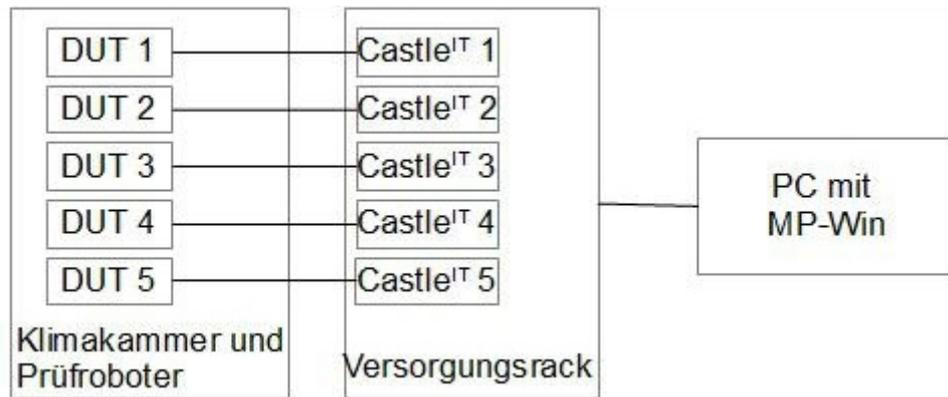


Abbildung 2: Schematischer Aufbau einer Lebensdauerprüfung

Damit das CASTLE^{IT} mit dem DUT kommunizieren kann, muss es Informationen über diesen DUT haben. Diese bestehen aus vorgefertigten Nachrichten mit deren Hilfe das CASTLE^{IT} den Prüfling ansprechen kann. Außerdem sind dort alle Nachrichten für die Restbussimulation vorhanden. Ebenfalls befinden sich dort die Grenzwerte auf die das CASTLE^{IT} die Prüflingsnachrichten testen kann. Da diese Informationen für unterschiedliche DUTs verschieden sind, muss das CASTLE^{IT} diese vor der Lebensdauerprüfung von MP-Win runterladen. Somit wird realisiert, dass mit dem selben CASTLE^{IT} unterschiedliche DUTs getestet werden können. Diese Daten liegen in einer editierbaren Form auf dem PC und können bei Bedarf geändert werden. Während der Prüfung kann MP-Win Messwerte vom Prüfling abfragen. Dafür sendet es eine Anfrage an das CASTLE^{IT}. Daraufhin fragt das CASTLE^{IT} den DUT in seinem Nachrichtenformat ab. Der erhaltene Messwert wird anschließend an MP-Win gesendet.

Bei einer Restbussimulation werden dem DUT zyklisch Botschaften gesendet, die es für seine Aufgaben benötigt. Bei einem Lenkstockschaltermodul wäre dies z.B. die Fahrzeuggeschwindigkeit. Ebenfalls prüft das CASTLE^{IT} ob die Sensordaten zum erwarteten Zeitpunkt eintreffen und sendet bei Nichteinhaltung eine Fehlermeldung an MP-Win.

Falls bei der Kommunikation mit dem Prüfling eines der oben genannten Protokolle zum

Einsatz kommt, werden diese vom CASTLE^{IT} selbstständig abgearbeitet. Somit können die reinen Nutzdaten weiterverarbeitet werden.

Das CASTLE^{IT} überprüft zudem die eingehenden Prüflingsdaten auf Grenzwerte. Sollte dabei eine Über- bzw. Unterschreitung festgestellt werden, sendet das CASTLE^{IT} ebenfalls eine Fehlermeldung an MP-Win.

Das CASTLE^{IT} verwendet momentan eine Philips ARMv7 CPU und soll in Zukunft einen Freescale MPC5554 PowerPC erhalten. Die Hardware wurde bereits entwickelt und mit rudimentären Testroutinen in Betrieb genommen. Im Laufe dieser Bachelor-Thesis soll die Software auf die neue Hardware portiert werden. Die Hauptanforderung hierbei ist es, die harten Echtzeitbedingungen für die Nachrichtenverarbeitung einzuhalten.

Begriffserklärungen

1. DUT – Device under Test (einzelner Sensor)
2. CASTLE^{IT} – Computer Aided System Testequipment for Laboratory Environment
3. MP-WIN – Multi Prüf-Win
4. CAN – Controller Area Network
5. LIN – Local Interconnect Network
6. ISO-TP Protokoll für CAN Bus zur Übertragung größerer Datenmengen
7. KWP2000 – Key Word Protocol 2000

[1] Werner Zimmermann Ralf Schmidgall Bussysteme in der Fahrzeugtechnik Protokolle, Standarts und Softwarearchitektur 4.Auflage Vieweg+Teubner Verlag

Bildquellen:

- Abbildung 1: ITronic GmbH
- Abbildung 2: eigene Darstellung

Effiziente Implementierung eines modellbasierten Algorithmus zur Verarbeitung des Fahrerwunsches an einem eBike

Thomas Stoof*, Reiner Marchthaler, Walter Lindermeir

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Aufgrund der zu erwartenden Probleme von konventionellen Antriebstechniken, wie fossiler Erschöpfung und der zu hohen Belastung der Umwelt durch Schadstoffemissionen, spielen Fahrzeuge mit Elektromotoren eine Schlüsselrolle in zukunftsfähigen Mobilitätskonzepten. Eine rasante Entwicklung zeichnet sich hierbei im Bereich der Elektrofahrräder, den sogenannten eBikes, ab. Bereits mehr als 50 europäische Fahrradmarken setzen auf das Bosch eBike-System, das damit Marktführer in Deutschland und Europa ist [1]. Das Portfolio des Bosch eBike-Systems beinhaltet die Komponenten Drive Unit, Battery Pack, Charger sowie HMI, bestehend aus Boardcomputer und Bedieneinheit.



Abbildung 1: Komponenten des Bosch eBike-Systems

Die eBikes haben bereits weite Käuferschichten erschlossen und sind auf dem Vormarsch. Zu den Zielgruppen gehören nicht nur die älter werdende Generation, die trotz Alter nicht auf die Mobilität mit dem Fahrrad verzichten möchte, sondern auch Pendler, die schnell und unverschwitz am Arbeitsplatz ankommen möchten. Des Weiteren wird auf ein junges, modernes und Lifestyle-orientiertes Publikum abgezielt. Ferner wird der Gesellschaft mit Hinblick auf die Energiewende ein Mobilitätskonzept angeboten, das umweltbewusst und sparsam zugleich ist.

Um noch verstärkt die bereitwillige, aber noch aufgrund der Kostenintensität unentschlossene Käuferschicht ansprechen zu können, ist eine ständige Verbesserung und Weiterentwicklung des Bosch eBike-Systems notwendig. Insbesondere die Erkennung des Fahrerwun-

sches ist ein Schlüsselfaktor für ein angenehmes Fahrgefühl. Aus dem Fahrerwunsch wird je nach Betriebsmodus eine entsprechende Drehmomentvorgabe für den Motorregler berechnet, die sich dann in einer Unterstützung für den Fahrer niederschlägt. Um auf die Drehmomentvorgabe schließen zu können ist eine Sensierung des vom Fahrer ausgeübten Drehmoments an der Tretwelle erforderlich. Diese Sensierung erfolgt üblicherweise mit einem Drehmomentsensor, der mit hohen Kosten verbunden ist. In der Erkennung des Fahrerwunsches steckt daher hinsichtlich der Kostenreduzierung ein hohes Potential.



Abbildung 2: eBike mit Komponenten des Bosch eBike-Systems

In einer vorangegangenen Master-Thesis wurde mit Matlab-Simulink ein modellbasierter Algorithmus entwickelt. Dieser Algorithmus dient zur Verarbeitung des Fahrerwunsches anhand einfacher Sensorik. Das Matlab-Simulink-Modell wurde nun im Rahmen der Bachelor-Thesis auf einer Prototyp-Plattform mit einem 32-Bit Mikrokontroller implementiert. Der ausgewählte Mikrokontroller, der keine Floating-Point-Unit besitzt, ist bezüglich Speicher und Taktrate eng auf die Anforderungen zugeschnitten. Aufgrund dieser Voraussetzungen musste zuerst das Matlab-Simulink-Modell und der dahintersteckende Algorithmus bezüglich Laufzeit und Speicherbedarf analysiert, optimiert, auf Festkomma-Arithmetik angepasst und diskretisiert werden. Die Fixed-Point-Toolbox von Matlab-Simulink stand dabei nicht zur Verfügung. Anschließend

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

wurde das vorab optimierte Modell manuell in performanten C-Code umgesetzt und gegen den aus Matlab-Simulink automatisch generierten C-Code des modellbasierten Algorithmus auf Speicherbedarf und Laufzeit verglichen.

Durch die Optimierung und die Anpassungen des Modells sowie geschickter Modellierung ist es gelungen sowohl den aus Matlab-Simulink automatisch generierten C-Code und den manuell implementierten C-Code zur Verarbeitung des Fahrerwunsches auf dem Prototyp-Target in Betrieb nehmen zu können. Beim Vergleich zwischen automatisch generiertem und manuell implementiertem C-Code konnten Unterschiede im Speicherbedarf und der Laufzeit festgestellt werden, wobei der manuell geschriebene Code sich als ressourcenschonender und schneller erwies.

Die Reduzierung an Speicherbedarf und der Verkürzung der Laufzeit durch den manuell implementierten C-Code steht im Gegensatz zum großen Vorteil der modellbasierten Entwicklung mit Matlab-Simulink.

Mittels Simulation des optimierten Modells mit echten Fahrdaten und der Bewertung des Algorithmus zur Verarbeitung des Fahrerwunsches bei Testfahrten konnten in allen Betriebsmodi zufriedenstellende Ergebnisse und eine plausible Drehmomentvorgabe festgestellt werden. Damit ist es gelungen einen komplexen Algorithmus zur Verarbeitung des Fahrerwunsches eines eBikes auf einem ressourcenarmen und vergleichbar langsamen Mikrocontroller ohne Floating-Point-Unit zu portieren. Für die Weiterentwicklung gilt es nun den Algorithmus weiter zu optimieren und gegebenenfalls die Parametrisierung anzupassen.

[1] Robert Bosch GmbH: Marktführer Bosch eBike Systems im Portrait <http://www.bosch-epresse.de> 26.11.2013

Bildquellen:

- Abbildung 1: Robert Bosch GmbH: Technik auf einen Blick <http://www.bosch-ebike.de> 26.11.2013
- Abbildung 2: Robert Bosch GmbH: Das bewährte eBike-System <http://www.bosch-ebike.de> 26.11.2013

Entwurf, Aufbau und Programmierung eines mechatronischen Systems zur manuellen motorunterstützten Führung eines elektrisch angetriebenen Fahrzeugs

Samed Sulanc*, Werner Zimmermann, Nikolaus Kappen

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

In unserer heutigen Zeit nutzen wir Menschen vermehrt technische Geräte, die uns viele sowohl alltägliche als auch undurchführbare Arbeiten erleichtern oder gar abnehmen. Sie spielen eine große Rolle als Helfer in unserem Alltag und ermöglichen einen Komfort, den wir vor einigen Jahren noch nicht kannten.

Diese Abschlussarbeit beschäftigt sich zentral mit der intuitiven Handhabung eines von einem Menschen an Handgriffen geführtes, elektrisch angetriebenes Fahrzeug. Der Benutzer soll dabei das Fahrzeug auf Ebenen und Steigungen bis zu 25 Prozent mit möglichst geringer eigener Krafteinwirkung führen [1]. Dadurch ist eine körperliche Entlastung gewährleistet.



Abbildung 1: Beispielanwendung von Kraftsensoren

Um dieses Ziel zu erreichen, wird ein ausgeklügelter Regelungsentwurf und das Zusammenspiel von Elektronik und Embedded Software benötigt. Der Steuergriff ist statisch am Fahrzeug angebracht und erfasst kleinste Verformungen durch die Krafteinwirkung des Benutzers.

Die vom Benutzer gewünschte Bewegungsrichtung kann somit elektrisch durch Kraftsensoren ermittelt werden. Aus diesen elektrischen Signalen werden die Stellgrößen für die Antriebsmotoren erzeugt, wodurch dem Benutzer ein sanftes motorunterstütztes Führen des Fahrzeuges ermöglicht wird.

Das System besteht aus zwei Modulen, die über einen Controller Area Network (CAN)-Bus miteinander kommunizieren. Ein Multi-Sensor-Modul erzeugt aus der gemessenen Verformungsarbeit zwei Soll Drehzahlen, die dann an das Motor-Control-Modul übertragen werden. Das Motor Control Modul regelt anhand der übertragenen Soll Drehzahlen die beiden Motoren eines elektrischen Differentials. (siehe Abbildung 2)

Multi-Sensor-Modul:

Zur Erfassung der Dehnungen und Stauchungen der Steuergriffe werden Dehnungsmessstreifen (DMS) in verschiedenen räumlichen Ebenen angeordnet und in Brückentopologien temperaturkompensiert verschaltet. Sie ändern bereits bei den kleinsten Verformungen ihren elektrischen Widerstand und sind ideal geeignet, um die verrichtete Verformungsarbeit zu messen und als elektrische Signale wiederzugeben, jedoch wird eine hochwertige Auswertelektronik benötigt.

Durch die Widerstandsänderung in den DMS entstehen Differenzspannungen am Ausgang der DMS-Brückenschaltungen. Diese werden über Differenzverstärker an die Analogeingänge eines Microcontrollers weitergeleitet. Dort werden sie gefiltert und skaliert einer integrierten 'Closed Loop' Regelung zur Verfügung gestellt. Die von der Regelung erzeugten Sollwerte werden weiter an das Motor Control Modul ausgegeben.

Motor-Control-Modul:

Die empfangenen Sollwerte dienen als getrennte Führungsgrößen einer unterlagerten Geschwindigkeits- und Stromregelung für

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

jeden Motor des elektrischen Differentials. Die Konstanz bzw. Variation der beiden Motordrehzahlen bewirkt sowohl das Geradeaus-

fahren als auch das Fahren entlang von Links- und Rechtskurven bis hin zur Drehung des Fahrzeugs um seine Hochachse.

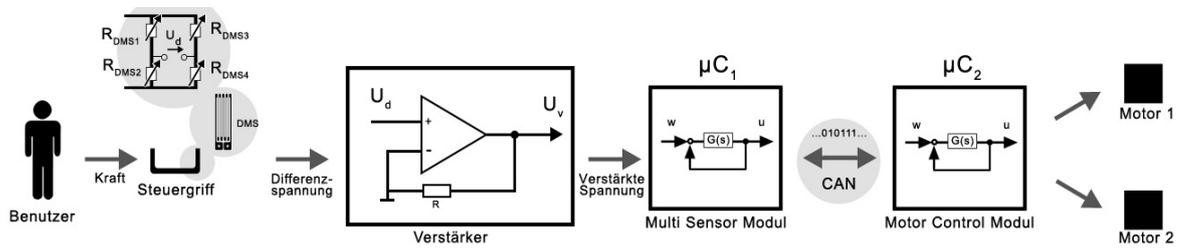


Abbildung 2: Vereinfachte Ansicht des Systems

[1] Quelle Kärcher GmbH

Bildquellen:

- Abbildung 1: Verfahrenssysteme touch2move, www.expresso.de 28.11.2013
- Abbildung 2: Eigene Darstellung

Konzeption, Aufbau und Softwareentwicklung eines modularen Messdatenerfassungssystems zur Erprobung von Reinigungsmaschinen

Mesih Tasci*, Werner Zimmermann, Karlheinz Höfer

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Bei vielen neuen Entwicklungen ist es nötig, das Produkt zu testen bevor es in Serie geht, Daten wie beispielsweise Ströme, Spannungen, Temperaturen, Drehzahlen zu erfassen und auszuwerten. Gerade deshalb gibt es in den meisten Unternehmen eine zusätzliche Abteilung, die solche Versuche durchführt. Um die Daten erfassen zu können, werden Datenlogger mit analogen und digitalen Eingängen benötigt. Oft kommen hier komplett fertige Lösungen in Betracht. Der Grund dafür ist leicht zu erklären. Die Unternehmen wollen nicht viel Zeit und Geld in die Entwicklung stecken, die Geräte sind leicht zu bedienen und kommen auch schon mit fertiger Software.

Nachteile solcher Systeme sind:

- sehr teuer
- Software oft fehlerbehaftet
- unflexibel
- keine Steuerung möglich

Derzeit wird in der Versuchsabteilung bei Kärcher ein neuer Bodenreiniger entwickelt. Die verfügbaren Datenlogger auf dem Markt sind für die Anforderungen aber ungenügend, da auch analoge und digitale Ausgänge verlangt werden, um Funktionsmuster zu steuern. Wie können die Anforderungen also am besten realisiert werden? Die Antwort liegt nahe: Es findet eine Eigenentwicklung statt, die die Anforderungen erfüllen soll. Es wird ein Konzept erstellt, Komponenten von verschiedenen Herstellern mit den gewünschten Ein- / Ausgängen bestellt und mit einem Minicomputer zu einem Messsystem verbunden. Ausgelesen werden müssen in diesem Fall Ströme, Spannungen, Temperaturen und Drehzahlen über das Modbus-Protokoll. Zusätzlich werden Daten wie Drehzahl, Gaspedalstellung, Ladezustand der Batterie, Lenkwinkel und Fahrtrichtung über CAN verschickt. [1] Diese müssen anhand einer Tabelle dekodiert werden. Das System basiert auf Linux. Auf dem Linux

Betriebssystem ist eine virtuelle Maschine mit Windows installiert, auf diese Weise ist es möglich, die Vorteile beider Betriebssysteme zu vereinen. Das Linux Betriebssystem wird mit einem Echtzeit Kernel betrieben und kommuniziert mit den Modulen. Die Daten werden solange gespeichert, bis sie von der Windows-Seite über das Interface angefordert werden. Die Kommunikation zwischen Windows und Linux findet über Sockets statt. Auf der Windows-Seite werden die Daten an eine Messdatenerfassungs-Software weitergegeben, gespeichert und ausgewertet. Der Minicomputer und die Module werden mechanisch sturzsicher und Spritzwasser geschützt in Gehäuse eingebaut. Der Endnutzer konfiguriert das System über einen Tablet-PC, der sich per Fernzugriff mit dem Minicomputer verbindet.



Abbildung 1: Bedienung des Datenloggers

Vorteile der Eigenentwicklung:

- flexible Anzahl von Ein-/Ausgängen
- erweiterbar mit neuen Modulen
- spritzwassergeschützt und sturzsicher
- Steuerung von Funktionsmustern möglich
- Einstellungen können leicht vorgenommen werden
- kostengünstig duplizierbar
- nur eine externe Software wird benötigt
- minimale Einarbeitung erforderlich

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

[1] Versuchsabteilung Kärcher

Bildquellen: Eigene Darstellung erstellt durch www.kaercher.com, www.footage.shutterstock.com, www.fibox.com 12/2013

Profiling von Software in C und C++ anhand eines realen Embedded Projektes

Nick Newill Tchouante Kembe*, Manfred Dausmann, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

In der Automobilindustrie sind die elektronischen und elektrischen Komponenten ein immer wichtigerer Anteil geworden. Mit der „Intelligenz“ steigt auch die Masse an im Fahrzeug eingesetzte Software. Die Komplexität der Software steigt somit in erheblichem Maße. Das lässt sich überprüfen mit den Steuergeräten, Fahrerassistenzsystemen und Multimedia-Geräten, die sich heutzutage im Auto befinden. Ein Überblick über die hochintelligente Kapazität verschafft uns die Abbildung 1.

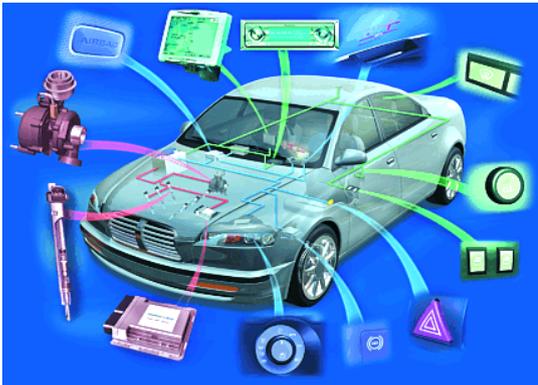


Abbildung 1: Elektronik und Software im Auto

Um die damit verbundenen Anforderungen zu erfüllen, muss immer komplexere Software im eingebetteten Bereich entwickelt werden, was von den Entwicklern eine konsistente Software- und Fehleranalyse verlangt. Die Analyse und Vergleich von laufenden Programmen ermöglicht die Aufdeckung von Softwarebereichen, die kritisch sind. In diesem Kontext gewinnt die Laufzeitanalyse an Bedeutung und erfordert gedankliche Überlegungen. Die drei Grundaspekte dieser Analyse sind das Messen der Geschwindigkeit – das Zählen und Messen der Funktionsaufrufe zur Optimierung des Programms –, die Speicherbenutzung und die Nebenläufigkeit der Prozesse.

Die Firma Robert Bosch GmbH entwickelt die Core-Funktionen – also die eigentliche Plattform – dieser Multimedia-Geräte für Fahrzeuge basierend auf einem eigenen Prinzip. Die Plattform besteht aus zwei Teilen:

- Kombisubsystem (KSS): hier werden funktionale Anforderungen bearbeitet.
- Graphisches Subsystem (GSS): stellt die Schnittstelle zum Benutzer dar.

Die beiden Subsysteme kommunizieren miteinander mittels einer Interprozessorkommunikation, um die Daten tauschen zu können.

Ziel der Bachelorarbeit ist es, die Plattform zur Laufzeit analysieren und die Daten prüfen zu können, um deren Konsistenz gewährleisten zu können. Dafür soll ein Programm entwickelt werden, das auf einer Seite die Laufzeitdaten misst und das auf der anderen Seite diese Daten dem Benutzer zur Verfügung stellt. Das heißt, dass das Programm die Ergebnisse speichert und diese ausgibt. Das ermöglicht den Vergleich der Daten, was wiederum für eine Fehleranalyse ein ernst zunehmender Vorteil ist.

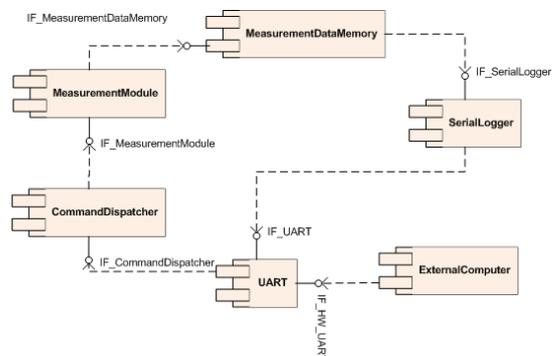


Abbildung 2: Real Time Tracer

Die Abbildung 2 stellt eine Übersicht über das zu entwickelnde System dar und zeigt die wichtigsten Komponenten und deren Interfaces. Die Komponenten sind die folgenden:

- ExternalComputer: präzisiert hier den externen Rechner. Von da aus können die Messungen gesteuert (Start, Stopp, Ausgabe) werden.
- UART: leitet die Daten vom ExternalComputer bis zum CommandDispatcher und liefert dem Externen Rechner die Ergebnisse.

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

- CommandDispatcher: empfängt die Eingabeparameter und sendet sie zu dem Messmodul (MeasurementModule).
- MeasurementModule: empfängt die Befehle für Messungen und leitet diese dementsprechend zu den Submodulen, um die spezifizierten Messungen durchzuführen. Die Messungen werden dann zurückgegeben.
- MeasurementDataMemory: empfängt die Ergebnisse der Messungen und speichert sie ab. Hier ist ein Ringpuffer vorgesehen, um die Strukturen bzw. Objekte – also die Ergebnisse – hinzu zu fügen und zu entfernen. Die Daten werden nach Eintreten von vordefinierten Triggern ausgegeben.
- SerialLogger: protokolliert die Messungen.

Es wird dem Benutzer ein breites Spektrum von Konfigurationsmöglichkeiten zur Verfügung gestellt. Welche Messungen sollen durchgeführt werden? Was geschieht wenn der Speicher voll ist, während es noch gemessen wird? Wann sollen die Ergebnisse weitergeleitet werden? Wie lange soll es gemessen werden? Mit welcher Frequenz (Häufigkeit)? Die Antwort zu diesen Fragen kann im größten Teil mittels der vom Benutzer eingegebenen Parameter gefunden werden.

Die vom UART zum externen Rechner weiterzuleitenden Daten sind die Ergebnisse der Messungen. Sie fokussieren auf bestimmte Zeiten. Wie viel Zeit vergeht zwischen dem Ankommen des Herunterfahren-Befehls und der Abschaltung des Subsystems? Von den Betriebsmodi sind es 3, die uns interessieren: „NORMAL“, „STARTUP“ und „HERUNTERFAHREN“. Bei der Umschaltung zwischen diesen drei Zuständen kann auch verzögert werden, was von großem Interesse für unsere Analyse ist. Die CPU-Ressourcen auf den 3 Ebenen – Task, Adressraum und Kernel – und deren Verbrauch spielen eine wichtige Rolle in der Geschwindigkeit und müssen deshalb unter die Lupe genommen werden. Wie viel Zeit geht verloren beim Schreiben bzw. Lesen der Flash-Speicher. Die aktuellen und durchschnittlichen Datenraten der Interprozessorkommunikation und die Signalqualität (BER: Bit Error Rate) untersucht man auch ganz genau, um eventuelle Mankos zu detektieren.

Eine Marktanalyse erfolgt letztendlich, um die sich schon auf dem Markt befindenden Profiling-Tools zu analysieren. Die Marktanalyse soll die Entscheidung vereinfachen. Es soll entschieden werden zwischen dem selbst entwickelten Profiling-Tool und den auf dem Markt erhältlichen Tools.

Bildquellen:

- Abbildung 1: http://www.siemens.com/innovation/de/publikationen/zeitschriften_pictures_of_the_future/pof_herbst_2005/das_mitdenkende_auto/software.htm 12/2013
- Abbildung 2: Eigene Abbildung

Konzeption und Realisierung eines Test-Management-Werkzeugs für Fahrzeugtests als Windows 8-Tablet-App auf Basis von C# und XAML

Bernd Theissler*, Manfred Dausmann, Dominik Schoop

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die heutigen Anforderungen an ein Testsystem im Automotive-Bereich sind zahlreich: viele Steuergeräte, Funktionen, Varianten, Testfälle und Testdurchführungsvarianten sollen abgedeckt werden. Die Berner & Mattner Systemtechnik GmbH hat zusammen mit einem namenhaften deutschen Automobilhersteller eine Toolsuite entwickelt, um den Testprozess optimal und vollständig zu unterstützen.

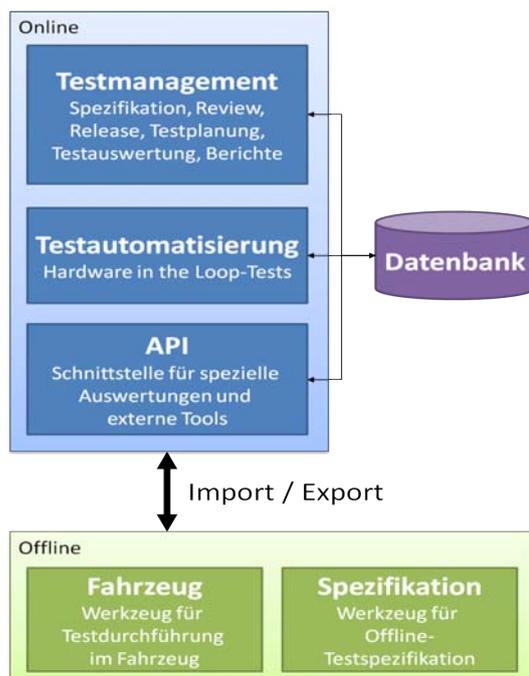


Abbildung 1: Übersicht des Test-Management-Systems für Fahrzeug- und HiL-Tests

Ein großer Teil der Tests werden hierbei direkt im Fahrzeug, beispielsweise auf Erprobungs- und Testfahrten, durchgeführt. Für diese Tests im Fahrzeug steht bereits eine klassische Windows-PC-Anwendung zur Verfügung. Diese Anwendung wird auf gewohnte Weise per Maus und Tastatur bedient. In einem typischen Einsatzszenario des Programms, wie das Bearbeiten von Testfällen während der Fahrt, ergeben sich aus der klassischen Bedienung der Software einige Nachteile für den Anwender. Beispielsweise ist das sichere Betä-

tigen von Schaltflächen per Maus beziehungsweise per Touchpad bei einem Laptop während einer Testfahrt unter Umständen nur schwer möglich.

Genau hier setzt das Thema der Bachelorarbeit an. Die bestehende Software für Fahrzeugtests soll als Windows 8-Tablet-App umgesetzt werden. Dabei soll und kann die Anwendung nicht einfach eins zu eins als App adaptiert werden, sondern die Umsetzung muss direkt auf die Vorteile und Eigenheiten von mobilen touchfähigen Geräten zugeschnitten sein. Die Vorteile eines touchfähigen Tablets, gegenüber einem Laptop, sind die direkte Interaktionsmöglichkeit mit der Bedienoberfläche einer Anwendung, die aus der Geräteform bedingte einfachere Handhabung und die in der Regel größere Mobilität im Hinblick auf die Akkulaufzeit und dem geringeren Gewicht.

Die Herausforderungen bei der Konzeption und Realisierung der Software als App sind umfangreich. Das User Interface und der Workflow der Applikation sind so zu gestalten, dass sich bereits mit der PC-Software vertraute Nutzer schnell zu Recht finden und keine lange Eingewöhnungsphase benötigen. Gleichzeitig muss die grafische Bedienoberfläche aber so gestaltet werden, dass beispielsweise ein sicheres Bedienen mittels Finger und Touchgesten, auch während einer Testfahrt, möglich ist. Entsprechend müssen die Bedienelemente in ihren Ausmaßen skaliert und angeordnet werden, um eine Fehlbedienung zu vermeiden. Hier gilt es besonders die Bedienelemente auf der grafischen Oberfläche so zu arrangieren, dass nicht mehrere dieser Elemente mit einem Finger gleichzeitig berührt werden können.

Bei der Adaption der bestehenden Bedienoberfläche nach Windows 8, ist ebenfalls zu beachten, dass einige klassische Bedienelemente aus dem Desktopsoftwarebereich wie Tabs, Dateibäume oder Tabellen in dieser Art in Windows 8-Apps nicht mehr zur Verfügung stehen. Diese Elemente müssen entsprechend ersetzt oder neu gestaltet werden. Die Bedienoberfläche beziehungsweise die App als Ganzes, sollte ferner den von Microsoft

* Diese Arbeit wurde durchgeführt bei der Firma Berner & Mattner Systemtechnik GmbH, Stuttgart

empfohlenen Designrichtlinien für Windows Store-Apps entsprechen, um sich optimal in die Windows 8-Umgebung zu integrieren. Die Richtlinien für die Gestaltung der Benutzeroberfläche geben beispielsweise Ratschläge über das Layout der Bedienoberfläche, über die Navigation durch die App oder über die Gestaltung der Benutzerinteraktionen per Fingereingabe [1].

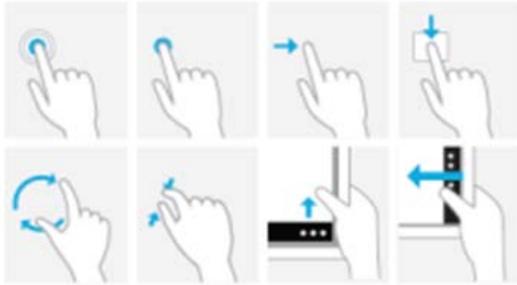


Abbildung 2: Übersicht der von Windows 8 unterstützten Touchgesten

Zudem wird bei Windows 8-Apps viel Wert auf ein schnelles und dynamisches Nutzererlebnis gelegt. Dies bedeutet, dass die Benutzeroberfläche beziehungsweise die App bestimmte Kriterien hinsichtlich der Performance erfüllen muss. Beispielsweise ist ein Kriterium, wie schnell das User Interface auf Benutzereingaben reagieren muss, um für den Benutzer als schnell, dynamisch oder träge wahrgenommen zu werden [2].

Ein weiterer wichtiger Punkt ist die Portierung der vorliegenden PC-Anwendung von Visual Basic 6 nach C#.NET beziehungsweise die Neuentwicklung der Geschäftslogik und der Architektur der Anwendung. Hier muss etwa das Exportieren und Importieren von Testpaketen, welche als XML- oder MDB-Dateien vorliegen können, zwischen dem Test-Management-System und der Fahrzeugtestsoftware möglichst flexibel und für zusätzliche Dateitypen erweiterbar gestaltet werden. Außerdem soll durch die Neuentwicklung der Architektur der Anwendung eine Trennung der grafischen Oberfläche, der Geschäftslogik und

der Daten der Anwendung realisiert werden.

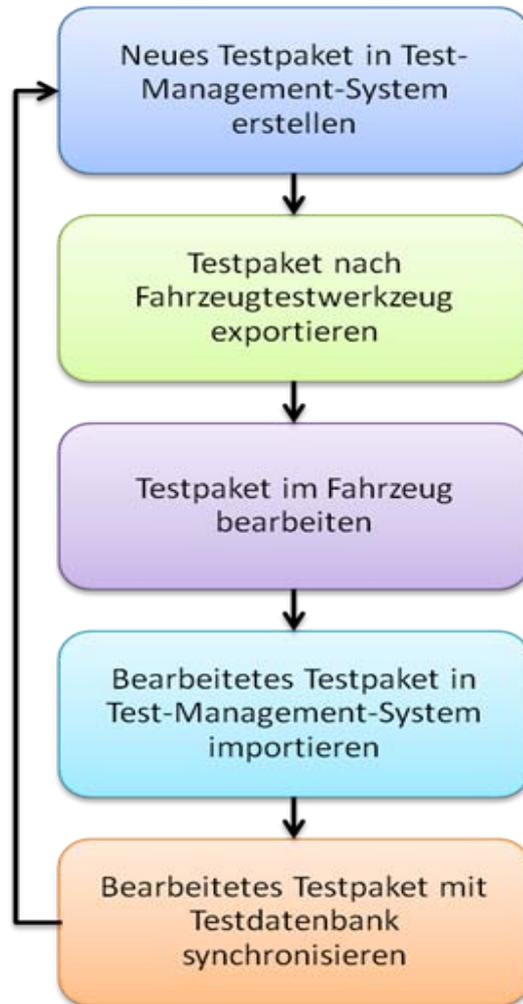


Abbildung 3: Prinzipieller Ablauf zur Bearbeitung eines Testpakets

Durch die Realisierung der PC-Anwendung des Fahrzeugtestwerkzeugs als Windows 8-App soll es dem Benutzer, durch die Verwendung von Touchgesten und eine darauf ausgelegte Bedienoberfläche, auf einfache und intuitive Art möglich sein, sicher und schnell Testpakete bearbeiten zu können.

[1] <http://msdn.microsoft.com/de-de/library/windows/apps/hh465424.aspx>
12/2013

[2] <http://msdn.microsoft.com/de-de/library/windows/apps/dn391697.aspx>
12/2013

Bildquellen:

- Abbildung 1: Eigene Darstellung
- Abbildung 2: <http://msdn.microsoft.com/de-de/library/windows/apps/hh465424.aspx>
12/2013
- Abbildung 3: Eigene Darstellung

Nutzerorientierte Plattform für den privaten Verleih von Büchern, Musik, Spielen und Filmen mittels einer Android-Applikation

Tobias Tropper*, Astrid Beck, Reinhard Schmidt

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die Zeiten eines statischen Internets sind vorüber und das Web 2.0 ist angesagter als jemals zuvor. Das Internet wird nicht mehr genutzt wie ein Buch, aus dem man lediglich seinen Inhalt ausliest. Jeder möchte vorhandene Inhalte bearbeiten, neue Inhalte erstellen und verbreiten. Seit Facebook, Google+ und andere soziale Netzwerke weit verbreitet sind, ist das umso leichter geworden. Doch das Internet begrenzt sich nicht nur auf das Teilen von Inhalten. Ein neuer Trend wird immer beliebter. Es handelt sich dabei um sogenannte Sharing Communities. Es gibt mittlerweile Plattformen für das Teilen von Autos, Wohnungen und Häusern, Alltagsgegenständen wie Bohrmaschinen oder Kameraobjektiven, Digitalen Medien wie Filmen oder Spielen, und sogar Lebensmitteln.

Gleichzeitig wächst die Anzahl der Smartphone-Nutzer, durch die Erreichbarkeit des Internets, das mit ihnen von überall aus genutzt werden kann und die hohe Funktionalität im Taschenformat, von Jahr zu Jahr.

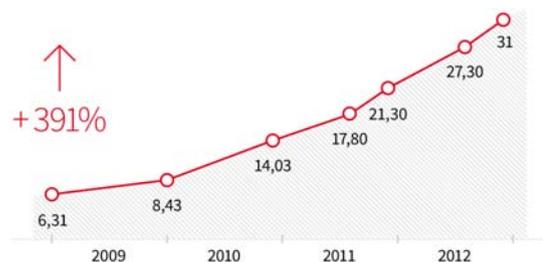


Abbildung 1: Anzahl der Smartphone-Nutzer in Deutschland in Millionen

Die Smartphone-Sparte ist in zwei große Lager gespalten. Den größten Marktanteil hat nach dem 2. Quartal 2013 mit 79,3% Googles Betriebssystem Android. Mit 13,2% liegt Apples iOS deutlich dahinter. Andere Anbieter wie Windows Phone (3,7%), BlackBerry OS (2,9%) und Symbian (0,2%) fallen kaum mehr ins Gewicht.

Ziel dieser Bachelor-Thesis war der Entwurf und die Implementierung einer Applikation, die Usern eine Plattform bietet, auf der sie

untereinander Bücher, Musik, Spiele und Filme leihen und verleihen können. Um eine größtmögliche Verbreitung zu ermöglichen wurde Android als Betriebssystem gewählt. Die Entwicklung sollte unter Berücksichtigung des User Centered Design Process geschehen.

Hierbei wird der Benutzer von vorneherein mit in die Konzeption und Entwicklung der Software einbezogen, um eine größtmögliche Gebrauchstauglichkeit zu gewährleisten.

Speziell auf die in DIN EN ISO 9241-110 definierten Kriterien wurde geachtet.

- **Aufgabenangemessenheit:** Alle Aufgaben des Users werden unterstützt.
- **Selbstbeschreibungsfähigkeit:** Durch Meldungen, Beschriftungen und eindeutige Icons ist alles ausreichend verständlich.
- **Steuerbarkeit:** Der User kann die Ablaufreihenfolge selbst bestimmen.
- **Erwartungskonformität:** Gewohnheiten und Erwartungen werden unterstützt.
- **Fehlertoleranz:** Trotz Eingabe eines Fehlers kann der User sein Ziel mit kleinsten Korrekturen erreichen.
- **Individualisierbarkeit:** Es besteht die Möglichkeit einer Anpassung an individuelle Bedürfnisse.
- **Lernförderlichkeit:** Der User benötigt zur Nutzung keine spezielle Vorbildung und kann Funktionen schnell erlernen.

Bei Usability-Tests, die im Entwicklungs-kreislauf in verschiedenen Projektphasen durchgeführt werden, wird die Anwendung mit Usern auf Schwächen getestet und verbessert.

Durch Prototypen, auch Mockups genannt, ist es möglich, Funktionen und Abläufe schon vor der Programmierung auf Funktionalität und Erreichbarkeit zu testen. Sollten die Testpersonen Schwierigkeiten beim Auffinden von Funktionen oder Verbesserungsvorschläge haben, können diese Dinge frühzeitig abgeändert werden und sparen somit Zeit und Kosten.

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



Abbildung 2: Prototyp der Applikation

Um eine bessere Nutzerbindung an die App zu erreichen und auch um den Usern mehr Freude beim Gebrauch zu bereiten und somit eine längere Gebrauchszeit mit der App zu schaffen ist es wichtig begeistern und fesseln zu können.

Geprägt wurde der Begriff Gamification im Sinne eines ernsthaften Hintergrunds mit spielerischen Aspekten. Auf der Jagd nach Punkten und Trophäen sind in Zeiten einer "verspielten Gesellschaft" [1] sämtliche Aufgaben und Anreize durch ein Spielsystem oder durch spielerische Lösungswege zu bewerkstelligen.

Durch Belohnungssysteme können Lernaufgaben oder unbeliebte Tätigkeiten mit Freude durchgeführt werden und User somit eine bessere Leistung erreichen.

Dies sollte nicht die Hauptaufgabe der App sein. Doch wenn durch einzelne Aktionen, wie Verleihen, Leihen oder Artikel einstellen, eine Möglichkeit darin besteht, Punkte auf einem Punktekonto zu generieren, ist die Anwendung mehr als ein reines Verleihportal. Es wird den Usern die Möglichkeit geboten, sich miteinander zu messen und gegeneinander anzutreten.

In die App wurden zwei Belohnungs-Systeme eingeführt: Die Möglichkeit, sich durch Punktgewinne ein höheres User-Level zu erspielen und sich für besondere Leistungen einmalige Auszeichnungen (Badges) zu verdienen.

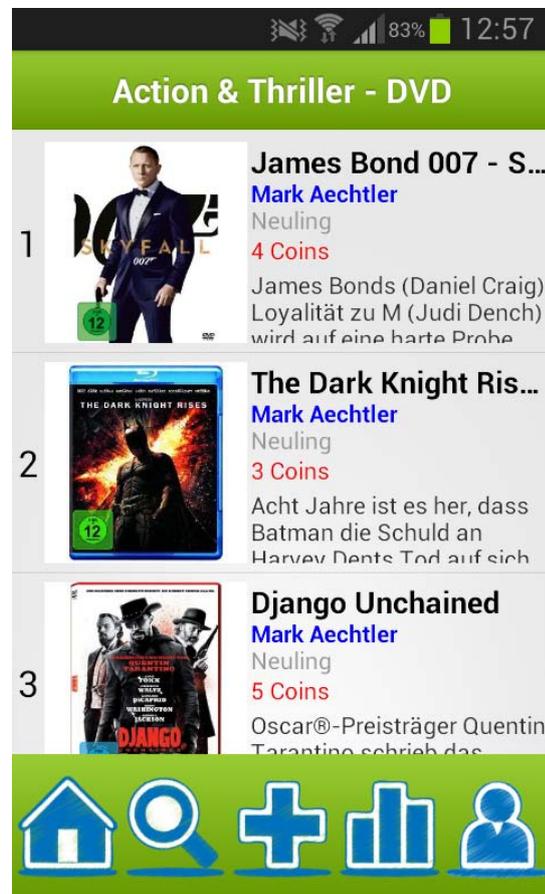


Abbildung 3: Screenshot der Applikation

[1] Nora Stampfl: Die Verspielte Gesellschaft

Bildquellen:

- Abbildung 1: comScore MobiLens
- Abbildung 2-3: Eigene Darstellung

Entwicklung und Integration eines Steer-by-Wire Systems für ein modulares Elektrofahrzeug

Ivan Vishev*, Werner Zimmermann, Reiner Marchthaler, Eduard Zimmer

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

"Das Projekt E-Auto ist dabei abzusterven"[1]. Um diese Aussage bestätigen oder widerlegen zu können, bedarf es einer gründlichen Untersuchung. Die Firma GIGATRONIK sieht Zukunftschancen für das E-Auto und beschäftigt sich mit den Fragen der Elektromobilität.

Um das Entwicklungsprojekt der Firma GIGATRONIK besser verstehen zu können, möchte ich zunächst den Begriff modulare Programmierung einführen. Der Begriff kommt aus der Softwareentwicklung und steht für eine bekannte und bewährte Vorgehensweise zum Aufbau erweiterbarer und anpassungsfähiger Software. Hierbei wird ein Programm in einzelne, austauschbare und variable Blöcke aufgeteilt, was die Flexibilität der Software erhöht. Jeder Block hat eine Schnittstelle, um sich ins Gesamtsystem zu integrieren. Auf diesem modularen Prinzip, auch Baukastenprinzip genannt, entwickelt die Firma GIGATRONIK zusammen mit der Firma Lauer & Weiss ein Modulares Elektro- Fahrzeug, kurz MEF.

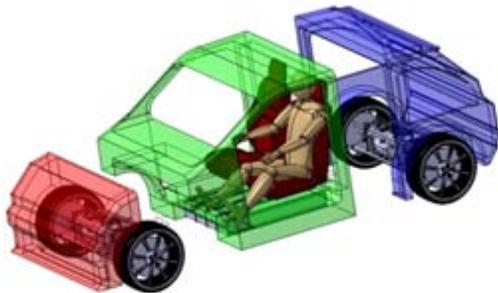


Abbildung 1: MEF

Die Entwicklung dieses Prototypen geschieht angelehnt an die Regeln der modularen Programmierung. Das Fahrzeug wird aus

Front-, Mittel- und Heckmodul bestehen, welche es in unterschiedlichen Ausprägungen geben wird. Jedes Modul wird definierte Schnittstellen haben, um mit den anderen Modulen elektrisch und mechanisch verbunden werden zu können und dann im Gesamtsystem zu agieren. Ein solcher Aufbau eröffnet neue Gestaltungswege und bietet die Möglichkeit in kürzester Zeit den spontanen Kundenwünschen zu entsprechen. Beispielsweise kann über die Leistungsfähigkeit, die Art des Antriebes als Front-, Heck oder Allradantrieb, oder die Bauform des Fahrzeugs als 2- oder 4-Sitzer, mit oder ohne erweiterte Ladefläche entschieden werden. Die Module werden durch möglichst einfache Zusammenführung miteinander verbunden. Es laufen keine Hydraulikleitungen durch das Gesamtfahrzeug und es gibt keinen Antriebsstrang, der außerhalb der einzelnen Modulgrenzen verläuft. Genau diese Unterschiede im Vergleich zu einem konventionellen Fahrzeug machen den modularen Aufbau möglich. Die Ziele des MEF-Projektes liegen unter anderem in der Entwicklung und Erprobung neuer Technologien mit allen ihren Herausforderungen. Nicht zuletzt dient das Projekt als eine Quelle für spannende und herausfordernde Themen für studentische Abschlussarbeiten und Praktika.

Steer-by-Wire System

Eine der vielen Komponenten, die für das MEF Fahrzeug entwickelt werden, ist die Lenkung. Aufgrund der starren mechanischen Verbindung zwischen Lenkrad und Lenkgetriebe ist eine rein mechanische Lenkung für das modulare Konzept nicht ausreichend flexibel. Stattdessen soll ein Steer-by-Wire System eingesetzt werden.

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

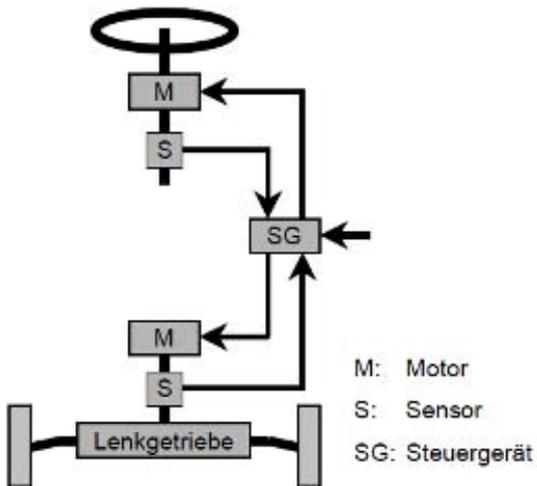


Abbildung 2: Steer-by-Wire System

Bei dem Steer-by-Wire System besteht keine mechanische Verbindung zwischen Lenkrad und Lenkgetriebe. Diese Tatsache trägt entscheidend zur Gestaltungsvariation bei, da das Lenkrad beliebig im Fahrzeug platziert werden kann. Durch den Wegfall der Lenksäule im Innenraum des Fahrzeugs steigt die passive Sicherheit. Es steigt auch das Angebot an Zusatzfunktionalitäten, wie beispielsweise eine variable Lenkübersetzung oder das Lenkgefühl [2].

Steer-by-Wire Systeme kommen heutzutage bereits in einigen Fahrzeugen zum Einsatz, angefangen bei Traktoren und Landmaschinen bis hin zu Flughafenfahrzeugen und Pistenraupen. Allerdings haben diese Fahrzeuge keine oder nur eine eingeschränkte Straßenzulassung und dürfen nur auf Feldwegen oder privatem Gelände bewegt werden. Der Grund dafür liegt vermutlich in der aus Sicht der Gesetzgebung nicht ausreichenden Sicherheit des Steer-by-Wire Systems. Bei einer konventionellen mechanischen Lenkung liegt die Wahrscheinlichkeit für einen Ausfall bei etwa $3 \cdot 10^{-8}$ [4]. Die durchschnittliche Ausfallwahrscheinlichkeit einer Autobatterie, an welche ein Steer-by-Wire System angeschlossen sein wird, liegt deutlich höher bei etwa $2 \cdot 10^{-5}$ [4].

Die Ausfallwahrscheinlichkeit eines nicht redundant aufgebauten Gesamtsystems hängt direkt von der höchsten Ausfallwahrscheinlichkeit einer Komponente ab. Somit gilt ein nicht ausreichend redundantes Steer-by-Wire System als unsicherer und kann nicht für die Straße zugelassen werden.

Aufgabenstellung

In meiner Arbeit entwickle ich ein Steer-by-Wire System für ein elektrisches Golf-Cart, welches als Aggregateträger für MEF dient. Zunächst wurde vorhandene Lenkung grundlegend untersucht und daraus die Anforderungen für die Komponenten des Steer-by-Wire Systems abgeleitet. Auf Basis dieser Anforderungen wurden die geeigneten Komponenten ausgesucht und beschafft. Eine große Herausforderung dieser Arbeit ist die mechanische Integration des Antriebsmotors für das Steer-by-Wire System. Hierzu muss der Motor in geeigneter Weise mit dem Lenkgetriebe des GolfCarts verbunden werden. In der Entwurfsphase werden die notwendigen Anforderungen an die Lenkung formuliert und weiter benötigte Komponenten, wie z.B. Sensoren ermittelt und Leitungssätze definiert. Nach einem Entwurf wird die Lenkung mit Hilfe von MatLab simuliert, um die geeigneten Reglerparameter für die Motorregelung zu bestimmen. Anschließend wird das System ins Fahrzeug eingebaut und gegen zuvor spezifizierte Tests verifiziert. In der Testphase wird vor allem die Einhaltung der Anforderungen an den Strom- und Spannungsverbrauch, und die Lenkwinkelzeiten der geprüft. Bei der Lenkwinkelzeit handelt es sich um die Zeit, welche die Räder brauchen um vom Winkel A zum Winkel B verstellt zu werden geprüft. Darüberhinaus sollen Schnittstellen für verschiedene mögliche Einheiten zur Lenkwinkelvorgabe wie Lenkrad oder Sidestick, Force-Feed-Back Einheit und spätere Fahrerassistenzsysteme geschaffen werden. Das Lenkungssystem soll hierfür eine geeignete Schnittstelle bereitstellen.

[1] Michael Bauchmüller, Süddeutsche Zeitung (2013)

[2] Pfeffer/Harrer Lenkungshandbuch (2011)

[3] Sieglin, Beitrag zu Energievers. eines innovativen DbW Fahrzeugkonzepts (2009)

Bildquellen:

- Abbildung 1: GIGATRONIK intern
- Abbildung 2: TU Darmstadt

Konzeption und Entwicklung einer Monitoring-Funktion für die Zustände von OS-Tasks und für AUTOSAR-Runnables

David Zentner*, Reinhard Keller

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2013/2014

Die vorliegende Arbeit befasst sich mit dem Ermitteln des Laufzeitverhaltens von Software für Steuergeräte im Automobil. Relevant sind dabei die Laufzeiten von Tasks, *Runnables* und Interrupt-Service-Routinen (ISR). *Runnables* sind nach AUTOSAR Funktionen der Anwendungsschicht und übernehmen die eigentliche Aufgabe des Steuergeräts. Des Weiteren werden *Runnables* im Kontext von Tasks ausgeführt. Die ISR sind nach AUTOSAR in zwei Varianten unterteilt. Die Kategorie 1 ist unabhängig vom Betriebssystem, was bedeutet, dass diese keinen Zugriff auf Betriebssystemfunktionen haben. Im Gegensatz dazu sind die ISR der Kategorie 2 sichtbar für das Betriebssystem, weshalb sie dessen Funktionen aufrufen können und zum Beispiel Tasks aktivieren dürfen. Mit dem aus dieser Arbeit resultierenden Prototyp ist es möglich, den zeitlichen Verlauf der Steuergerätesoftware darzustellen (vgl. Abb. 1). Abbildung 2 zeigt die zugehörige Legende.

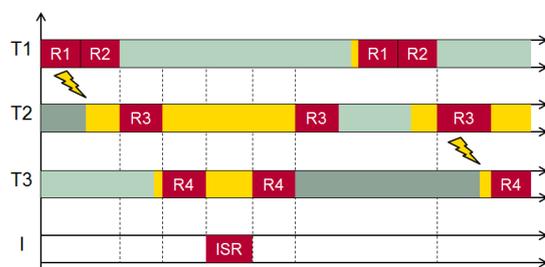


Abbildung 1: Möglicher Softwareverlauf

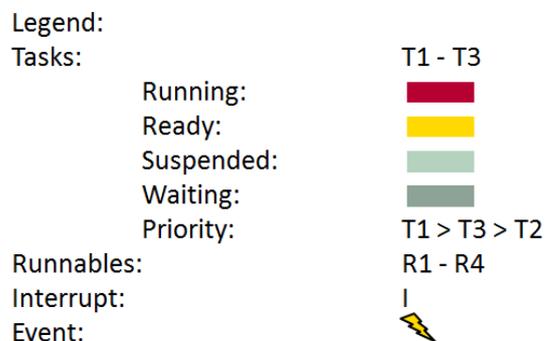


Abbildung 2: Legende

Für die Entwickler von Embedded Software im Automobilbereich ist es wichtig zu wissen, ob es Engpässe bei der Ausführung der Software gibt. Diese Engpässe entstehen beispielsweise durch zu häufiges Aktivieren von Tasks, dem gehäuftem Auftreten von ISR und auch durch falsche Konfiguration der Software. Die Einschätzung, ob solche Engpässe auftreten können, wird in verschiedenen Phasen der Softwareentwicklung durchgeführt. Sind die Anforderungen definiert, muss festgelegt werden, welche Hardware zu verwenden ist. Dabei muss geprüft werden, ob die vorgesehene Hardware den Anforderungen entspricht oder bereits potentielle Laufzeitprobleme erkannt werden können. Sobald die Software vollständig verfügbar ist, kann auf Analyse-Software zurückgegriffen werden, z.B. auf den Timing Analyzer der Vector Informatik GmbH. Darin können die implementierten Tasks und ISR zusammen mit ihren zyklischen Aktivierungszeiten und den Worst-Case-Laufzeiten angegeben werden. Aus den Angaben berechnet der Timing Analyzer dann Engpässe in der Laufzeit. Bei diesem Verfahren treten Ungenauigkeiten auf, da angenommen werden muss, dass alle Tasks und ISR zyklisch aufgerufen werden, was häufig jedoch nicht der Realität entspricht. Bisher nicht möglich ist jedoch das Nachvollziehen des tatsächlichen Software-Ablaufs im Steuergerät und die Erstellung einer dazu gehörenden Fehleranalyse. Hier setzt die vorliegende Arbeit an.

Im ersten Schritt war zu untersuchen, welche Daten im Steuergerät zur Erfassung des Laufzeitverhaltens genutzt werden können. Dabei genügt es, die Zustandsübergänge zu erkennen. Während *Runnables* und ISR nur je zwei Zustände besitzen, können Tasks jedoch bis zu 4 Zustände annehmen. Nach AUTOSAR sind zwei unterschiedliche Task-Varianten, *Basic Tasks* und *Extended Tasks*, möglich. Zustände und Übergänge einer *Extended Task* sind in Abbildung 3 dargestellt. Bei einer *Basic Task* entfällt der Zustand *Waiting*, da dieser Typ nicht für das Warten auf Ereignisse unterbrechbar ist.

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

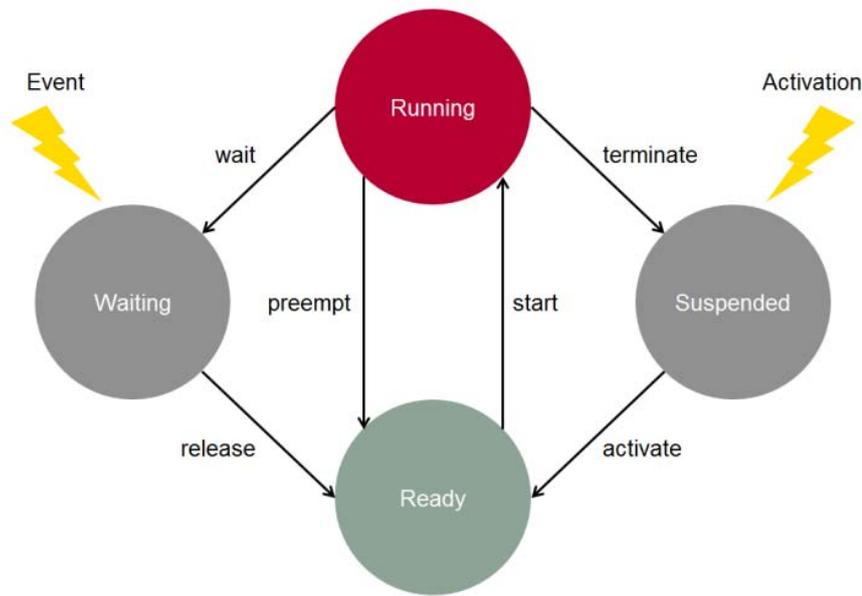


Abbildung 3: Extended Task

In AUTOSAR-Software gibt es unterschiedliche Methoden zur Erkennung dieser Zustandsübergänge. Im Falle der *Runnables* und der Interrupts der Kategorie 2 gibt es sogenannte Hook-Funktionen. Diese erlauben es, vor und nach der Ausführung des *Runnables*, bzw. des Interrupts, eigene Funktionen auszuführen, in denen auf die notwendigen Daten zugegriffen werden kann. Während die Hooks eine Änderung in der Basissoftware (BSW) vermeiden, gibt es für Tasks und Interrupts der Kategorie 1 keine solche Möglichkeit. Das bedeutet, dass direkt in die BSW-Funktionen eingegriffen werden muss, um die benötigten Daten zugänglich zu machen. Die Basissoftware besteht aus Softwaremodulen, die sich unterhalb der RTE (Laufzeitumgebung) befinden. Einige der Module sind in Abbildung 4 dargestellt.

Für eine Analyse durch den Steuergeräteeintegrator müssen die gewonnenen Daten zudem an ein Monitoring-Tool, beispielsweise CANoe der Vector Informatik GmbH, übermittelt werden. Dabei wird auch eine Kommunikationsschnittstelle zu dem auswertenden Computer benötigt. Die Ermittlung aller benötigten Kommunikationsarten war der zweite

Schritt in der vorliegenden Arbeit. Neben den üblichen Kommunikationsprotokollen wie CAN und FlexRay besteht oft auch die Möglichkeit, über eine Debugger-Schnittstelle zu kommunizieren. Für diese Art der Kommunikation steht die VX1000-Hardware der Vector Informatik GmbH zur Verfügung. Sie kommuniziert über XCP on Ethernet mit Monitoring-Tools wie CANoe und über JTAG oder Nexus mit dem Steuergerät. Während die herkömmliche JTAG-Schnittstelle nur Daten auslesen kann, solange die Steuergerätesoftware angehalten ist, können über Nexus ab der Klasse 2+ auch zur Laufzeit Daten aus dem Steuergerät ausgelesen werden. Der Vorteil der VX1000-Hardware und von Nexus ist, dass Live-Streaming der Daten möglich wird. Somit können bei jedem Zustandswechsel die Daten ausgelesen und an das Monitoring-Tool übertragen werden. Eine Schwierigkeit mit der VX-Hardware ist jedoch, dass ein festes Messraster von 15µs besteht und der Sendepuffer bei zu vielen Events, hier also zu vielen Zustandsübergängen, überlaufen kann. Daraus folgt, dass bei diesem Lösungsansatz das Puffern der Daten im Steuergerät sinnvoll ist, um die Robustheit zu erhöhen, vgl. [1].

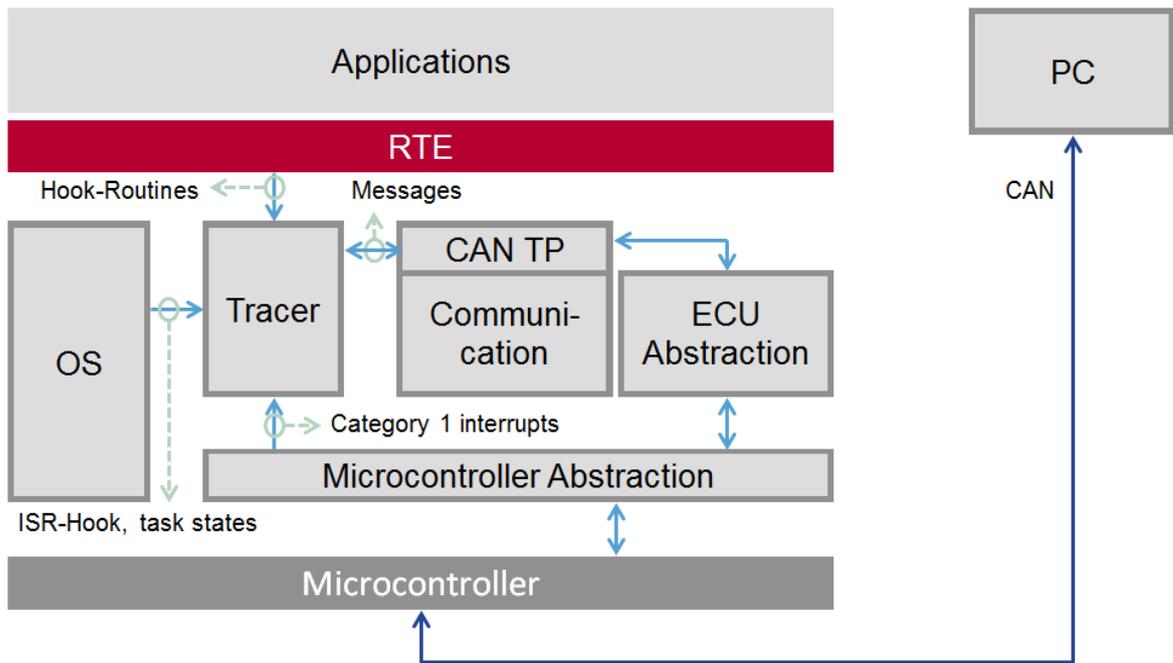


Abbildung 4: Kommunikationsmodell

Für die Realisierung eines Prototyps im Rahmen der Arbeit wird die Kommunikation über CAN verwendet, da die VX-Hardware den Vorteil des Live-Streamings durch das Messraster verliert. Des Weiteren ist im Gegensatz zur Nexus-Schnittstelle die CAN-Kommunikation weit verbreitet. Andererseits ist CAN mit der im Automobilbereich üblichen höchsten Datenrate von 500kBit/Sekunde nicht in der Lage, alle Daten zur Laufzeit zu versenden. Daher ist das Puffern im Steuergerät zur Vermeidung von Datenverlust unvermeidbar. Ein weiterer Nachteil ist, dass die CAN-Kommunikation selbst Interrupts auslöst und somit das Messergebnis zudem verfälscht, da Interrupts mitgemessen werden. Auf eine Übertragung während des Messvorgangs wird deshalb ganz verzichtet. Zudem ist

die Messdauer begrenzt, da die Daten im RAM zwischengespeichert werden und RAM nur begrenzt verfügbar ist. Somit wird die Messung nach Auftreten einer bestimmten Anzahl an Events beendet. Wenn dies zu vermeiden ist, kann in einen weiteren Mess-Modus geschaltet werden. In diesem Modus werden die bestehenden Daten überschrieben, sobald der Puffer übergelaufen ist. Der Anwender kann nun selbst bestimmen, wann die Messung beendet werden soll. Mit Beendigung der Messung werden die Daten aus dem Puffer automatisch verschickt.

Die Entwicklung eines Tools zur Interpretation der übertragenen Daten war nicht Bestandteil der vorliegenden Arbeit und bietet die Grundlage für eine weiterführende Arbeit.

[1] Vector Informatik GmbH. vector.com. [Online] 07 2013. [Zitat vom: 15. 11 2013.] http://vector.com/portal/medien/cmcc/datasheets/VX1000_DataSheet_EN.pdf.

Bildquellen:

- Abbildung 1, 2 und 4: Selbst erstelltes Bild
- Abbildung 3: OSEK VDX Workshop. Vector Informatik GmbH. Stuttgart Weilimdorf : s.n., 2011.

