Echtzeit

Rundbrief Nr. 10, Mai 2021

Mitteilungen
des GI/GMA/ITG-Fachausschusses
Echtzeitsysteme

GESELLSCHAFT FÜR INFORMATIK E.V.

VDI VDE VDI/VDE-Gesellschaft
Mess- und Automatisierungstechnik

ITG INFORMATIONSTECHNISCHE
GESELLSCHAFT IM VDE

Impressum

Herausgeber	GI/GMA/ITG-Fachausschuss Echtzeitsysteme https://www.real-time.de
Sprecher	Prof. Dr.-Ing. habil. Dr. h.c. Herwig Unger FernUniversität in Hagen Lehrgebiet Kommunikationsnetze 58084 Hagen herwig.unger@fernuni-hagen.de
Stellvertreter	Dipl.-Inform. (Univ.) Marcel Schaible FernUniversität in Hagen Lehrgebiet Kommunikationsnetze 58084 Hagen marcel.schaible@fernuni-hagen.de
Redaktion	PD Dr.-Ing. habil. Mario Kubek Dipl.-Ing. Jutta Düring FernUniversität in Hagen Fakultät für Mathematik und Informatik 58084 Hagen echtzeit@fernuni-hagen.de
ISSN	2199-9244

Redaktionell abgeschlossen am 05. Mai 2021

Einreichung von Beiträgen: Alle Leserinnen und Leser sind aufgerufen, das Mitteilungsblatt auch zukünftig durch Beiträge mit zu gestalten, um den Informations- und Meinungsaustausch zwischen allen an den Fragen der Echtzeitprogrammierung Interessierten zu fördern.

In dieser Ausgabe

1 Tagung Echtzeit 2021 – Call for Papers

Mario Kubek, Redakteur Echtzeit-Rundbrief
Homepage: https://www.real-time.de/CfP.html

Die Tagung „Echtzeit 2021“ findet erstmals im Hotel Bellevue in Boppard am Rhein statt. Das Leitthema dieses Jahr lautet:

Echtzeitkommunikation
11. und 12. November 2021

Zu folgenden und benachbarten Themen werden Vorträge über Methoden, praktischen Einsatz, Erfahrungen und Ausblicke erbeten. Exponate sind immer willkommen.

- Protokolle
- Datengetriebene Kommunikation
- Big Data
- Ad-hoc- und Sensornetze, IoT, CPS
- Echtzeitsignalverarbeitung
- Sicherheit und Fehlertoleranz
- Modellierung und Simulation
- Robotik, Medizintechnik, Smart Grids
- Mobilität, Transportwesen und Logistik
- Konferenzsysteme
- Weitere aktuelle Anwendungen
- Ausbildung

Stichtag für die Vortragsanmeldung ist Sonntag, der 23. Mai 2021. E-Mail: tagung@real-time.de.

2 Einreichung von Vorschlägen zum Graduiertenwettbewerb 2021

Mario Kubek, Redakteur Echtzeit-Rundbrief
Homepage: https://www.real-time.de/preise/grad-bedingung.html

Bis zu drei herausragende Diplom-, Bachelor- und Masterarbeiten sowie Dissertationen im Bereich Echtzeitsysteme werden mit je 500,- Euro prämiert. Die Abschluss- und Doktorarbeiten sollten zum Zeitpunkt der Einreichung abgeschlossen und bewertet, aber nicht älter als ein Jahr sein. Elektronisch vorzulegen sind:

- die eingereichte Abschluss- oder Doktorarbeit,
- eine Zusammenfassung der Arbeit,
- Angaben zur Person der Verfasserin / des Verfassers (CV)
- sowie eine Stellungnahme der Betreuerin / des Betreuers.

Das Ende der Vorschlagsfrist ist Sonntag, der 16. Mai 2021. Vorschläge sind zu richten an Frau Prof. Dr. Juliane Benra (benra@jade-hs.de).

3 Veranstaltungen mit Unterstützung des Fachausschusses

Mario Kubek, Redakteur Echtzeit-Rundbrief

12. Jahreskolloquium „Kommunikation in der Automation“
Homepage: https://www.jk-komma.de

Termin: 17./18. November 2021
Ort: Magdeburg

Das Jahreskolloquium „KommA“ des Instituts für industrielle Informationstechnik (inIT) der Hochschule OWL und des Instituts für Automation und Kommunikation (ifak) e.V. findet 2021 in Magdeburg statt. Das Kolloquium ist ein Forum für Wissenschaft und Industrie im deutschsprachigen Raum für alle technisch/wissenschaftlichen Fragestellungen rund um die industrielle Kommunikation.

4 Echtzeit 2020 – ein Resümee

Autor: Herr Bastian Seeger, Studierender der FernUniversität in Hagen

Die 41. Echtzeitkonferenz (https://www.real-time.de/archiv/ez20.html) fand am 20.11.2020 in Form einer Online-Konferenz statt. Natürlich ersetzt nichts den persönlichen Kontakt mit den Kollegen. Aufgrund der Coronasituation in diesem Jahr war ein Treffen in Boppard nicht möglich und so wurde die Software Zoom (https://zoom.us/) genutzt, um auf dem Höhepunkt bis zu 47 Teilnehmer virtuell miteinander zu vernetzen. So konnten auch tausende Kilometer entfernte Kollegen in der Kaffeepause fachkundige Gespräche in den Breakout-Rooms führen, dem Gefühl des ‘Heilig Grab’ in Boppard zumindest ein wenig nahe kommen und in diesen schwierigen Zeiten den persönlichen Kontakt und wissenschaftlichen Diskurs aufrecht erhalten.

Unter dem Titel *Kommunikationssicherheit im IoT* wurden während der Tagung interessante Bezüge zwischen den hochaktuellen Themen Sicherheit, KI und Echtzeitanforderungen im IoT hergestellt und aus verschiedensten Perspektiven beleuchtet. Obgleich einige Paper auf Englisch verfasst waren, konnte man sich auf die deutsche Sprache bei den Vorträgen einigen. Neben dem Eröffnungsvortrag und dem Vortrag und der Ehrung des diesjährigen Preisträgers Georg von der Brüggen, gab es jeweils drei Vorträgen zu *System Design*, *Sicherheit* und *aktuelle Anwendungen*. Aufgrund des diesjährigen Formates des Workshops wurden nur 11 Vorträge an einem Tag, anstatt der üblichen 14 auf zwei Tage verteilt, zugelassen. Neben den Papern enthält der Tagungsband[1] eine detailierte Einführung in OpenPEARL.

Die Zusammenfassung der Vorträge basiert sowohl auf dem jeweils im Tagungsband veröffentlichten Paper, als auch auf dem mündlichen Vortrag am 20.11.2020.

4.1 Eröffnungsvortrag und Preisträger

4.1.1 Die Anfänge der Fluglärm-Erfassung in Deutschland

Brigitta Holleczek, vertreten von Peter Holleczek, war von 1970 bis 1984 zuerst in der Werkstatt, dann in der Entwicklung und dem Vertrieb bei Siemens tätig und kam dabei mit einem neuartigen Projekt zur Realisierung von permanenten Fluglärmmessanlagen in Kontakt. Sie gibt der Öffentlichkeit nun Einblicke in die damalige Arbeit [Tagungsband S. 1-10].

Das Paper, mit einigen Anekdoten im Vortrag, bietet besondere Einblicke in die gesellschaftlich kontroverse Thematik der Fluglärmbelastung aus der Perspektive der Anfänge technischer Realisierbarkeit von Anlagen zur Messung von Fluglärm, wobei ein Fokus auf die Entwicklungsaspekte, sowie begleitende legislative Aspekte und Normungsanstrengungen gelegt wurde. Gerade weil die Publikationslage in diesem Bereich begrenzt ist, kann die vorliegende Arbeit von Frau Holleczek hier eine Lücke schließen.
Möglich, auch unter technischen und wirtschaftlichen Aspekten, wurden permanente Fluglärmmessungen durchsetzen legaler Rahmenbedingungen mit dem *Gesetz zum Schutz gegen Fluglärm* vom 30.03.1971 (novelliert 2007) und dem Aufkommen von Kleinrechnern und Mikroprozessoren in nicht länger saalfüllender Größe, sondern in einer Rack-Bauform. Dazu kam später eine Präzisierung durch die DIN 45643-1 *Messung und Beurteilung von Flugzeuggeräuschen* und DIN 45641 *Mittelung von Schallpegeln*, welche unter anderem die Berechnung des *äquivalenten Dauerschallpegels* mithilfe aufsummierter gewichteter Mittelwerte, sowie Grenzwerte und Schutzzonen, und auch Lärmereignisse inklusive deren Identifikation, Klassifikation als auch deren Zuordnung zu konkreten Flügen genauer standardisierte. Ohne Einschränkung der Allgemeinheit fokussiert sich die Arbeit auf die Entwicklung des verteilten Systems bei Siemens und zeigt auch anekdotisch die Schwierigkeiten einer technischen Umsetzung in der damaligen Zeit auf. Das verteilte Messsystem bestand aus einer zentralen Instanz, mit moderner Bedienung durch Tastatur und Bildschirm. Im 16kb Speicher wurden in ASS 300 mit ORG 320 in Assembler geschriebene Makros auf einem Siemens Typ 320 16 Bit Mikroprozessor

[1] Herwig Unger, Herausgeber. Echtzeit 2020 – Kommunikationssicherheit im IoT. Informatik aktuell, Springer Fachmedien Wiesbaden, 2020.

ausgeführt und Messwerte (6 bit und ein Paritätsbit) verarbeitet, wobei eine Pufferbatterie vor Stromausfällen schützte. Kommuniziert wurde digital mit 200 Bit/s mit der Messstation, wobei die zweiadrige Leitung der Deutschen Post auch im Analogmodus (defaultmäßig für 20 Sekunden) betrieben werden konnte, um direkt an der Station gemessene Werte 'mitzuhören'. Dafür wurde im Prototypen zuerst ein Z20 Fernwirksystem, später ein Intel 8086 in Kombination mit einem Rohde & Schwarz Schallpegelmesser genutzt. Die Sensorik hing stabil verbaut an einem Mikrofongalgen und erlaubte auch durch Korrelation mit anderen Messstation die Identifizierung von Fluglärmereignissen und der Bestimmung deren Dauer. Die Prototypanlage wurde in Nürnberg (1974-1977) errichtet, später folgten Stationen in Salzburg, Stuttgart, Hamburg, Zürich und Berlin Tegel im Beisein der damaligen 4-Mächte Standortkommandatur. Anschaulich wurde der Vortrag durch zahlreiche Grafiken und Messwertauswertungen, die in den 70ern auf Blattschreibern gedruckten 'liegenden Zeilengrafiken'.

In der dem Vortrag folgenden Diskussion kristallisierte sich deutlich heraus, dass großes Interesse am Vergleich mit modernen Realisierungen besteht, unter anderem die Frage, ob in jüngster Vergangenheit auch Neuronale Netze zur Identifikation von Fluglärmereignissen genutzt wurden, oder auch ob mittlerweile Parameter wie die Windstärke- und Windrichtung in die Schallpegelmessung mit einbezogen werden. Die bereits erwähnte Publikationslage erweist sich als dürftig, sodass vorliegende Arbeit hoffentlich der Anlass für weitere Veröffentlichungen, auch in Form einer Vergleichsbasis für moderne Systeme, darstellt.

4.1.2 Realistic Scheduling Models and Analyses for Advanced Real-Time Embedded Systems

Der Fachausschuss *Echtzeitsysteme* der Gesellschaft für Informatik zeichnete Georg von der Brüggen von der TU Dortmund für seine herausragende Dissertation aus. Der Vortrag stellte eine anschauliche, durch Beispiele gestützte Ergänzung zu dem vorliegenden Paper [Tagungsband S. 11-20] dar. Die folgenden Ausführungen basieren nur auf dem Paper und Vortrag der Tagung, und erheben keinen Anspruch auf eine umfassende Diskussion der ganzen Dissertation. Herr von der Brüggen hat einen entscheidenden Beitrag dazu geleistet, Schwächen bisheriger Evaluationsmetriken für Echtzeit Scheduling Algorithmen und Modelle hervorzuheben und gleichzeitig Wege aufzuzeigen, hier einen realistischeren Ansatz zu nutzen, um damit bessere, auf einen bestimmten Einsatzzweck abgestimmte Echtzeitsysteme, ohne eine Überprovisionierung von Ressourcen, zu konstruieren.

Zu den bekannten Metriken zählen *Speedup Factor* und *Parametric Utilization Bound*. Die drei aufeinander aufbauenden Komponenten System- und Taskmodell, der Scheduling Algorithmus mit einem Scheduling Test und die theoretische und empirische Performanceevaluation wurden aus verschiedenen Blickwinkeln vor dem Hintergrund dieser Metriken betrachtet. Es wurde Bezug genommen auf *periodic* und *sporadic task models* und das *Static-Priority Preemtive Scheduling*. Zur Auswertung von Utlization und Acceptance Rate anhand des *Rate-Monotonic Schedulings* wurde *Hyperbolic Bound* (linear), *Time Demand Analysis* (pseudo-polynomial) auf empirischer Seite und auf theoretischer Seite, als Baseline, *Liu und Layland Bound* genutzt.

Time Demand Analysis zeigt eine deutlich höhere Acceptance Rate als Hyperbolic Bound für das erzeugte Taskset, wobei beide natürlich über der von Liu und Layland gefundenen Obergrenze von 69,3 % Utlilization (bei der noch 100 % der Tasks geschedult werden können) liegen. Eine interessante Erkenntnis stellt sich beim Vergleich mit *Earliest Deadline First Scheduling* heraus, denn der Speedup Factor ist für alle drei Fälle $1/0,693 \approx 1,44$. Wird nun zusätzlich die Verteilung der Taskdauer mit berücksichtigt, so zeigt sich, dass zusätzliche Informationen hilfreich für die Bewertung sind und z.B. der Speedup Faktor keine ausreichende Entscheidungsgrundlage für die Realisierung eines Systems darstellt, da damit nur unter bestimmten Umständen das konkrete Verhalten des Systems in der Praxis beurteilt werden kann. Zur Augmentation dient ein neu entwickelter *Blocking Factor* γ der das maximales Verhältnis von Blocking Time zu Worst-Case-Execution-Time über einem gegebenen Taskset darstellt. Im Weiteren stellte Herr von der Brüggen einen eigens entwickelten *self-suspension* Algorithmus *SEIFDA* (Shortest Execution Interval First Deadline Assignment) vor, welcher auf einem hybriden self-suspension Modell beruht. In der Evaluation zeigte er überzeugende Acceptance Ratio Werte im Vergleich mit anderen Algorithmen z.B. einem klassischen *dynamic self-suspension model*.

Es liegt auf der Hand, dass die gewonnenen Erkenntnisse, gerade weil sie auf realistische Anwendungen abzielen, eben in jenen auch implementiert und erprobt werden sollten. Der Forschungsgemeinde stehen jetzt dank der Dissertation von Herrn von der Brüggen die dafür notwendigen Voraussetzungen zur Verfügung.

4.2 System Design

4.2.1 Hardware/Software Co-Design für eine Modulare Systemarchitektur

Präsentiert wurde die Arbeit [Tagungsband S. 21-30] von Carsten Weinhold, wobei er und Michael Roitzsch den Hauptautor Nils Asmussen bei der Ausarbeitung unterstützt haben. Die Arbeit entstand am Barkhausen Institut, welches sich dem Internet der Dinge und Verbesserungen dessen von allen Seiten her widmet. Hierbei werden unter anderem bekannte Aspekte aus Soft- und Hardwareentwicklung neu kombiniert und erweitert, wobei vorliegende Arbeit ein exzellentes Beispiel für die Mission des Barkhausen Instituts darstellt. Zum Tätigkeitsbereich des Instituts gehören z.B. auch Chipdesign, Antennenentwurf, Signalmodulation sowie hier besprochene Betriebssysteme mit Security und Privacy Aspekten. Die vorliegende Arbeit stellt einen wichtigen Beitrag hin zur flächendeckenden Umsetzung des Security by Default Paradigmas im IoT Bereich (aber auch mit Anwendungsmöglichkeiten in Cloud und anderen Gebieten) dar, indem ein Hardware/Software Co-Design System M^3 entwickelt wurde und weiter ausgebaut wird, welches die Kommunikation von Komponenten eines Systems begrenzt und somit gegenüber monolithischen und auch gegenüber auf Mikrokernel basierenden Systemen eine Separierung von Vertrauensbereichen schon in Hardware erzwingt.

Cyber-Physikalische Systeme haben eine besondere Bedeutung im Internet der Dinge, da sie aktiv auf unser Leben und unsere Umwelt Einfluss nehmen können. Hierbei sind sicherheits-

kritische Merkmale als Schwerpunkte bei der Entwicklung solcher Systeme zu setzen, um nicht zuletzt Gefahr für Leib und Leben auszuschließen bzw. zu minimieren.
Die Isolierung von einzelnen Komponenten, sowie die Verwaltung über Capabilities (Erlaubnis zur Kommunikation mit einer anderen Komponente) sind für eine sicheres Interagieren von verschiedensten *VPE* (virtual processing Einheiten) vonnöten. M^3 abstrahiert nun einzelne Kacheln, die wiederum z.B. Hardwarebeschleuniger, Software auf eine General Purpose Core oder beliebige andere Software- und Hardwarekonstrukte (TPU, GPU, DRAM usw.) sein können und denen VPEs zugeordnet werden. *TCUs* (Trusted Communication Units) isolieren die Kacheln vom NoC (Network on Chip), über das die Kacheln eine einheitliche Schnittstelle nach außen hin zur Verfügung stellen. Die Kernel Kachel kann nun einzelnen TCUs Kommunikation untereinander erlauben. Nicht autorisierte Kommunikation ist somit grundsätzlich nicht möglich.
Eine TCU hat konfigurierbare Endpunkte für Speicherzugriff und für das Senden und Empfangen von Nachrichten. Diese zwei Paradigmen sind ausreichend, um beliebige Kommunikation zu ermöglichen. Ein neu entworfenes File-Protokoll wurde bereits in Hardware realisiert und bietet Speicherzugriffe über den Memory Endpoint einer TCU. Ebenso umgesetzt wurde ein spezielles Dateisystem m3fs, sowie ein virtuelles Terminal mit Unterstützung von aus der Linux Welt bekannten Pipe-Mechanismen, wobei hier Datenströme von Kachel zu Kachel 'gepipt' werden können. Dabei wurden auch empirische Tests durchgeführt und es konnte gezeigt werden, dass die dezentral organisierte Kommunikation zwischen den Kacheln besser für viele parallele Tasks skaliert, da die zentrale Prozessorlast durch wegfallende zentral koordinierte Datentransfers geringer ausfällt. Dies ist besonders interessant für Echtzeitanwendungen, da die autonom arbeitenden Kacheln jeweils als separierte Systeme modelliert werden können. Ein besonderer Fokus wurde auch auf die Übertragung der M^3 technologie auf Rechenzentren gelegt, da diese Teil der IoT Infrastruktur sind und dort heute im Hinblick auf die Leistung notwendige, aber architekturbedingt unsichere Remote Direct Memory Access Methoden genutzt werden. Analog zur TCU böte sich mit SmartNICs (programmierbare NICs) hier eine Architektur an, um Netzwerkkommunikation zu reglementieren.

Aktuell laufen noch einige Forschungsvorhaben – so wird z.B. M^3 vom Simulator auf eine echte FPGA Hardware portiert und, wie bereits erwähnt, im Cloudbereich an der Caladan-Architektur geforscht. Zudem wird an einem Ohua-Compiler gearbeitet, der bestimmte Garantien für M^3 Programme zur Kompilierzeit bieten soll und Programmierern Komfortfunktionen wie Debugger bieten und bekannte prozedurale Programmierparadigmen auf verteilten Kachelstrukuren erlauben soll. Mit Bezug zur ECHTZEIT sind vielfältige weitere Forschungsthemen, wie die Evaluation von Echtzeiteigenschaften von M^3, denkbar.

4.2.2 Hard Real-Time Memory-Management in a Single Clock Cycle (on FPGAs)

Simon Lohmann, Master 2016 an der Bergischen Universität Wuppertal und dort wissenschaftlicher Mitarbeiter am Lehrstuhl für Automatisierungstechnik/Informatik, hielt den Vortrag. Er forscht für seine Promotion zum Thema Echtzeitdatenbanken mit Fokus auf *hard real time*. Co-Autor war Dietmar Tutsch. Da aktuelle Speicherverwaltungssysteme den im Rahmen der

Dissertation gesetzten Anforderungen für ein Echtzeitdatenbanksystem nicht genügen, wurde hier die Forschungslücke gefüllt und mit vorliegender Arbeit [Tagungsband S. 31-40] ein mittels FPGA realisiertes Memory-Management System konstruiert, das alle Operationen mit Worst Case O(1) realisiert und somit hart echtzeitfähig ist.

Klassische Lösungen zur Speicherverwaltung, die auf eine Reduzierung der Speicherdefragmentierungszeit abzielen, sind statisch allokierende Lösungen, denen aber eine zeitliche Dynamik fehlt, und die Memory Pool Methodik, bei der jedoch auch Fragmentierung und das Problem fehlender Größendynamik auftritt, sowie auch Overhead durch Aufteilung und Rekombination von Blöcken der Pools entsteht. Die vorgeschlagene neuartige Architektur setzt hingegen auf gleichgroße Memoryzellen (sinvollerweise entsprechend der Größe des Speicherbusses), bestehend aus Datenfeld und next Pointer, über den sich verkettete Listen von Memoryzellen realisieren lassen. Der FPGA muss nur zwei Zähler c_{free} (Anzahl freie Zellen), c_{rbnu} (Anzahl reservierter, aber nicht genutzter Zellen) und einen Pointer *ptrUnused* (der auf das erste Element der verketteten Liste von freien und nicht verwendeten Speicherzellen verweist) verwalten. Die Zähler können über Multiplexer und Addierer oder über Multi-Input-Addierer mit Koeffizient auf dem FPGA umgesetzt werden. Sieben Operationen (inklusive Idle Operation) dienen der Reservierung, Allokation und der Freigabe von Speicherzellen: *PickUpFreeCell, ReturnFreeCellByAddress, RequestReservation, PickUpReservedCell, ReturnReservationByAmount, ReturnFreeCellRingByAddress, Idle.* Es können also auch Mengen von Zellen allokiert und wieder freigegeben werden, wobei ein Zellenzugriff immer nur auf eine einzelne Zelle erfolgt – benötigt der Nutzer einen Block von Zellen, so muss eine Virtualisierung in Form einer MMU dazwischen geschalten werden. Dies ist für den Anwendungsfall von Herrn Lohmann aber nicht relevant, weswegen der Performancevorteil voll genutzt werden kann. Konkret laufen die Operationen folgendermaßen ab:

- ***PickUpFreeCell***: c_{free} wird dekrementiert und *ptrUnused* auf *memory(ptrUnused).next* gesetzt
- ***ReturnFreeCellByAddress(addressIn)***: c_{free} wird inkrementiert, *memory(addressIn).next=ptrUnused* und anschließend *ptrUnused* auf *addressIn* gesetzt
- ***RequestReservation(n)***: c_{free} wird um n dekrementiert und c_{rbnu} entsprechend erhöht
- ***PickUpReservedCell***: c_{rbnu} wird dekrementiert und *ptrUnused = memory(ptrUnused).next* gesetzt
- ***ReturnReservationByAmount(n)***: Inkrementiere c_{free} und dekrementiere c_{rbnu} um n
- ***ReturnFreeCellRingByAddress(addressIn)***: Zuerst *ptrUnused=memory(addressIn)* setzen, dann *memory(addressIn).next=ptrUnused* und dann $c_{free} = c_{free}$+*memory(addressIn).data*

Es scheint es naheliegend, auch im Hinblick auf die dem Vortrag folgenden Fragen zur MMU und virtuellen Speicherverwaltung oberhalb der Low-Level Implementierung des vorgestellten Systems, auch weitergehende Nutzungsszenarien mit Blöcken von Memoryzellen hinsichtlich der Möglichkeit, für diese Szenarien ebenfalls harte Zeitschranken zu garantieren, zu erforschen.

4.2.3 Künstliche Intelligenz in der Miniaturautonomie

Stephan Pareigis und Tim Tiedemann von der *autosys* Forschungsgruppe an der HAW Hamburg widmen sich in vorliegender Arbeit der Künstlichen Intelligenz in der Miniaturautonomie. Neben einer Einführung in Lernalgorithmen und den Herausforderungen im Zusammenhang mit Simulation und realen Situationen von Stephan Pareigis, gab Tim Tiedeman einen Ausblick auf künftige Anstrengungen hinsichtlich der Entwicklung von autonomen Miniaturluftfahrzeugen. Die Co-Autoren übernahmen den Mittelteil des Vortrags mit *Flexible und modulare Modell-Landschaft für autonome Miniaturfahrzeuge* (Luk Schwab), *Autonome Miniaturfahrzeuge* (Markus Kasten), *Selbstlokalisierung für ein autonomes Modellschiff* (Thorben Schnirpel) und *Verstärkungslernen im Gazebo-Simulator für ein autonomes Modellschiff* (Henri Burau). Ein weiterer Co-Autor ist Morten Stehr.
Das Paper [Tagungsband S. 41-50] und die Arbeit der Forschungsgruppe im Allgemeinen liefern wichtige neue Erkenntnisse, um die *Sim2Real*-Lücke bei der Entwicklung autonomer Systeme, sowohl zu Land, zu Wasser als auch in der Luft zu überwinden. Die Synergie aus der Nähe zum Miniaturwunderland Hamburg bietet ein einmaliges Umfeld, um Daten für Lernverfahren in einer sicheren, aber realistischen Umgebung zu sammeln und daraus Schlussfolgerungen für autonome Fahrzeuge in Originalgröße abzuleiten.

Obwohl das Multilayer Perceptron bereits in den 60er Jahren entwickelt wurde, gab es noch in den 90er Jahren Unsicherheit bei der Nutzung Neuronaler Netze zum Zweck der Approximation nichtlinearer Kontrollprobleme. Die vergangenen Jahre brachten dann einen Durchbruch sowohl bei Bilderkennung mit Convolutional Neural Networks, als auch bei selbstlernenden autonomen Systemen durch Reinforcement Learning Algorithmen.
Um Unfälle und hohe Kosten zu vermeiden, bietet es sich an, anstatt mit realen Fahrzeugen lieber im Miniaturformat zu experimentieren. Dazu wurde neben der Nutzung des Miniaturwunderlands auch eine eigene Laborumgebung 'Microwunderland' eingerichtet, um gezielt mit unterschiedlichen Belichtungs- und Umgebungsverhältnissen testen zu können. Das Microwunderland bietet verschiedene Verkehrssituationen in Stadt und Umland und erlaubt durch eine eingebaute Metallplatte und Magneten an Bäumen, Häusern und sonstigen Umgebungselementen einen schnellen Umbau. Das Faller-Car-Leitsystem ermöglicht einen Datensammelmodus, wobei das System auch durch einen Servo ersetzt werden kann, sodass im autonomen Modus auch die Steuerung des Fahrzeugs per Servo und Hall-Sensor (für die Messung des Lenkwinkels) in der Lenkung erfolgen kann. Mittlerweile liegt das ca. 13x2,5 cm große Fahrzeug (Maßstab 1:87) in Version 3 vor und erlaubt die onboard Auswertung von Daten mittels Raspberry 3 Zero inklusive Google TPU Edge Beschleuniger über USB. Ansätze mit einem Espressif ESP32 oder über einen aufwendig konstruierten FPGA mit 4-Kern ARM Chip wurden zugunsten der mit modernen Kamera-, Sensor- und Hardwarebeschleunigungssystemen kompatiblen und mit vielfältig vorhandenen Softwaremodulen ausgestatteten Architektur verworfen. Neben dem Landfahrzeug wurde die 'NHAWigatora' gebaut. Das im Maßstab 1:100 (97x14x20 cm) konstruierte Modellfrachtschiff besitzt IMU, Kamera, 2D Lidar und Infrarot Distanzsensoren. Durch Korrelation der Bewegung um die drei Achsen mit den Sensordaten des Lidars kann eine 3D Karte erstellt werden (SLAM). Die Infrarotsensoren dienen der

Abstandsmessung zur Küste und bieten eine billige Alternative zur Odometrie mit Lidar. Im *gazebo*-Simulator unter Anbindung von *usv_gazebo_plugin* (für Wassersimulation) und *open ai gym* (für Lernverfahren) konnte der Reinforcment Learning Algorithmus *Deep Q-Networks* mit 6x300x300x4 Neuronen (6 Inputfeatures und 4 Steuerungsoutputs) trainiert werden. Der Algorithmus lieferte überzeugende Ergebnisse für Raster- und Pfadtest, wobei die Bewegungen des Schiffs für das menschliche Auge aufgrund der Diskretisierung der Outputs des Algorithmus gewöhnungsbedürftig erscheinen (Schiff fährt viel rückwärts und seitwärts), wobei dies mit Algorithmen für einen kontinuierlichen Aktionsraum (z.B. DDPG oder SAC) und mit geigneten Lerning-Policies in Zukunft verbessert werden könnte.

Aktuell laufen Forschungsvorhaben, um autonome Luftfahrzeuge im Miniaturformat auf einem autonomen Miniatur-Frachtschiff landen zu lassen. Besonders im Bereich der Luftfahrzeuge sind Simulationsmöglichkeiten beschränkt, da z.B. Turbulenzen noch nicht vollständig verstanden werden und aber Tests in Realgröße auf der anderen Seite lebensgefährlich sind. Hinsichtlich von Luftfahrzeugen gelten besondere Anforderungen wie etwa die an das Gewicht. In diesem Bereich kommt es auch auf eine leistungsstarke Hard- und Software an, um in Echtzeit aus den Sensordaten Aktionen ableiten zu können (ohne abzustürzen, da einfaches Anhalten im Luftraum schwieriger als zu Land oder Wasser ist), wobei hier das Forum der Echtzeitkonferenz in den nächsten Jahren einen geeigneten Rahmen dafür bieten könnte.

4.3 Sicherheit

4.3.1 Integration realer Angriffe in simulierte Echtzeit-Ethernet-Netzwerke

Sandra Reider, Bachelorstudentin und studentische Mitarbeiterin an der HAW Hamburg, präsentierte die in Zusammenarbeit mit Philipp Meyer, Timo Häckel, Franz Korf und Thomas C. Schmidt im Rahmen des SecVI (Security for Vehicular Information) Forschungsprojekts an der HAW Hamburg entstandene Arbeit [Tagungsband S. 51-60]. Der Forschungsschwerpunkt liegt auf der Entwicklung neuer Architekturen für Fahrzeugnetzwerke und deren Sicherheit. Die vorliegende Arbeit fokussiert sich dabei auf den Aspekt der Simulation dieser Fahrzeugnetzwerke, insbesondere im Hinblick auf die Möglichkeit zur Einspielung aufgezeichneter realer Angriffe in diese zum Zweck der Analyse. Dies stellt einen entscheidenden Schritt in Richtung reproduzierbarer Tests auf verschiedenen simulierten Netzwerkarchitekturen und Skalierbarkeit durch Nutzung verschiedenster Angriffe, auch aus entfernten Quellen in universeller Form als pcapng File, was neue Möglichkeiten für die Erprobung experimenteller Fahrzeug-Netzarchitekturen bietet.

Klassische Fahrzeugnetzwerke sind in Domänen aufgeteilt, wobei die Kommunikation über langsame CAN Busse oder Flexray erfolgt. Moderne Anwendungsszenarien, wie vernetzte Infotainmentsysteme, stellen neue Anforderungen an die Geschwindigkeit der fahrzeuginternen Busse, aber auch an deren Sicherheit und Übertragungseigenschaften wie Echtzeitgarantien, insbesondere da Fahrzeugnetze sich zunehmend nach außen öffnen, subsumiert unter dem Begriff Car2X. Tests neuer Komponenten können im Fahrzeug oder in begrenzten Versuchsauf-

bauten stattfinden, oder durch geringeren Ressourceneinsatz und flexibel, bei gleichzeitiger Möglichkeit der Introspektion, in einer Simulation untersucht werden. Die verwendete Simulationsumgebung basiert auf einer ereignisorientierten Simulationssoftware *OMNeT++* sowie dem *INET* Framework, welches gängige, auf Ethernet basierte Protokolle zur Verfügung stellt. Die Forschungsgruppe hat auf github zudem *CoRE4INET* als Open Source Software veröffentlicht, was die Simulation z.B. um Echtzeiteigenschaften (TSN) ergänzt. Dazu kommt *FiCo4OMNet*, für CAN/Flexray, und ein Modul zur Simulation von Gateways, um beispielsweise CAN Frames durch Ethernet zu tunneln. Das Paper ergänzt diesen Aufbau nun um ein pcapng-Lesemodul und einen Paketgenerator, welcher sich in die Simulation von Steuergeräten einbinden lässt.
Es wurde ein Fallbeispiel eines DoS Angriffs auf ein Fahrzeugbordnetz vorgestellt. Das simulierte Fahrzeugnetz ist in 9 Zonen (vorne-Mitte-hinten-links-rechts) aufgeteilt und verbindet die Zonen durch einen Ethernet Backbone. Die Kommunikation zwischen Kamera (vorne rechts) und Sensorfusion (hinten links) wurde aufgezeichnet. Dies diente als Baseline für den Vergleich mit den vom Kommandozeilenprogramm t50 generierten UDP Paketen (mit hoher Priorität 6, bei 0-gering und 7-höchste Priorität), die einen DoS Angriff simulieren sollten. Es konnte eine erhöhte Latenz beobachtet werden, wobei bereits geringe Änderungen bei Latenz und Linkauslastung relevant für die Anomaliedetektion sein können.

Grundsätzlich soll der Paketgenerator in weiteren Iterationen erweitert werden. Forschungsmöglichkeiten auch im Hinblick auf autonome vernetzte Fahrzeuge und deren Onboard Netzarchitekturen bieten sich nahezu grenzenlos, wobei basierend auf vorliegendem Ansatz die Vergleichbarkeit von Lösungsmöglichkeiten hergestellt wurde. Insbesondere wies auch Philipp Meyer in der Fragerunde auf Forschungsanstrengungen rund um Anomaliedetektionssysteme hin, deren Eignung und Eigenschaften sich sehr gut in simulierten Netzwerken erforschen lassen. Auch die Echtzeiteigenschaften der Simulation stießen in der Fragerunde auf besonderes Interesse.

4.3.2 Sichere Mobilfunkkommunikation für ein Fahrzeugleitsystem

Christoph Maget von der FernUniversität Hagen hat als Beiprodukt seiner Dissertation ein Paper [Tagungsband S. 61-70] zur perfekt sicheren Kommunikation in Fahrzeugleitsystemen (Inter-Fahrzeugkommunikation) veröffentlicht. Vor dem Hintergrund wachsender Anforderungen und Einsatzzwecke im Industrial IoT rücken Aspekte der sicheren Kommunikation weiter ins Zentrum, wobei heute gängige Verfahren der (a)symmetrischen Verschlüsselung prinzipiell angreifbar sind, mit Ausnahme der bisher wenig genutzten ‘perfekten’ symmetrischen Maskierungsverfahren wie One-Time-Pad. Herr Maget hat mit der Vorstellung eines mit heutigen Standards konformen neuen Fahrzeugleitsystems mit perfekter Verschlüsselung zum einen die Praktikabilität dieses Vorhabens gezeigt, zum anderen auch konkrete Architektur und Entwurfsentscheidungen vorgestellt und ebenso die Grundlagen für die Vereinheitlichung bestehender, heterogener Strukturen im Bereich der Inter-Fahrzeugkommunikation geschaffen. Besonderes Augenmerk wurde auch auf die Echtzeiteigenschaften dieses Systems gelegt.

Fahrzeuge können im Ad-hoc- oder im Infrastrukturmodus vernetzt werden. Aus den Standards *ETSI TR 102 962* (IVS und zellenbasierte Kommunikation) und *DIN EN ISO 24534-3:2016-08* (Elektronische Identifizierung von Fahrzeugen) ergibt sich, dass Fahrzeugleitsysteme mit bestehenden zellenbasierten Netzwerken kompatibel sein und eine eindeutige Identifikation des Fahrzeugs gewährleisten müssen, weswegen nur ein Fahrzeugleitsystem mit zentraler Instanz infrage kommt. Hinsichtlich der Sicherheitsanforderungen kann mit der *DEN EN ISO/IEC 27000:2020-06* (zentrale Normenreihe für IT-Sicherheit) und der *VDI/VDE 2182 Blatt 1:2020-01* (überträgt die Anforderung der ISO 27000 auf automatisierte Maschinen und Anlagen) abgeleitet werden, dass über die ganze Lebensdauer des Produktes (bei Autos sind das häufig 20 Jahre) die Informationssicherheit gewährleistet werden muss, was die perfekte symmetrische Verschlüsselung bei Einhaltung organisatorischer Maßnahmen zur Vermeidung von Seitenkanalangriffen leisten kann. Gestützt werden diese Folgerungen durch den Anforderungskatalog der *DIN CEN/TS 17182:2019-03* (eCall Notrufsystem) und *DIN EN ISO 18750:2018-09* (beschreibt Datennutzung von Teilsystemen intelligenter Verkehrssysteme). Bisher wurde das One-Time-Pad für die Übertragung von fahrzeugbezogenen und streckenbezogenen Daten nicht genutzt, da der Vorrat an Masken mindestens so groß sein muss wie die zu übertragende Datenmenge, jedoch konnte die Arbeit zeigen, dass selbst unter Berücksichtigung vollständig autonomer Fahrzeuge (Nachrichten werden mit 10 Hz versendet) und einer hohen Fahrleistung gerade einmal $\approx 80GiB$ Daten pro Jahr anfallen, die an eine zentrale Instanz übertragen werden müssen. Der erforderliche Vorrat an Masken kann dank heute üblichen Speichermedien selbst für Jahrzehnte bei der Produktion des Autos oder während der Wartung/des Tankens provisioniert werden.
Konkret erfolgt die Provisionierung durch eine zentrale hoheitliche Behörde, die für einen 80 Bit Maskenanzeiger durch hinreichend gute Zufallsgeneratoren Schlüssel erzeugt und diese an Fahrzeug und das zentrale Relais (Übertragungsstation) verteilt. Über den Maskenanzeiger können Fahrzeug und zentrales Relais bei der Kommunikation den verwendeten Schlüssel identifizieren. Lediglich beim Relais liegt die Nachricht im Klartext vor – bei der Übertragung ist durch das One-Time-Pad Verfahren die Kommunikation perfekt geschützt. Empirische Messungen haben gezeigt, dass das One-Time-Pad auch sehr gut für Echtzeitanwendungen nutzbar ist, es über die Nachrichtenlänge linear skaliert und es bei kleiner Nachrichtenlänge sogar deutlich schneller als z.B. AES256 ist.

Fragen im Anschluss des Vortrages gab es zu Seitenkanalangriffen, die aber, wie bereits erwähnt, durch organisatorische Maßnahmen bekämpft werden können. Zudem wurde der Aspekt der Anonymisierung thematisiert, der sich mit einer zentralen Instanz und durchgängiger Authentifizierung bei fahrzeugbasierter Kommunikation schwer realisieren lässt. Die auf github veröffentlichte Software soll weiterentwickelt werden. Weitere Forschungsanstrengungen sollen hinsichtlich der Nachrichtenstruktur (Möglichkeit zur Einsparung von Daten) und des Übertragungsprotokolls unternommen werden. Insbesondere die Evaluation der Möglichkeit des Einsatzes einer Blockchain zum Speichern der Vertrauenswürdigkeitsparameter aller Teilnehmer des Fahrzeugleitsystems könnte interessante Forschungsvorhaben anstoßen.

4.3.3 Programmunbeeinflussbare Authentifikation von Eingaben auf berührungssensitiven Sichtfeldern

Robert Fitz von der HAW Hamburg wurde bei dem Vortrag von seinem Doktorvater Wolfgang A. Halang vertreten. Moderne Smartphones und Tablets bieten lediglich ein berührrungssensitives Sichtfeld (Touchscreen) als Eingabemöglichkeit, weswegen hier häufig Gesten- oder Pinauthentifizierung eingesetzt wird, die im öffentlichen Raum nicht zuletzt durch viele Kameras und die häufige Aufforderung zur Authentifizierung leicht ausspähbar sind. Biometrische Merkmale wie Gesicht, Stimme oder Fingerabdruck bieten Komfort, aber auch wenig Sicherheit, da sie leicht täuschbar sind (beispielsweise die Kamera mit einem Bild). Daneben ist immer die Gefahr eines fehlerhaften Prozessors, einer schadhaften Anwendung oder eines gehackten Betriebssystems bzw. grundsätzlich die Programmbeeinflussbarkeit der Authentifizierung gegeben. Durch Zugriff auf Sensordaten wie Gyroskop können Applikationen sogar Rückschlüsse auf den Pin bekommen.
Herr Fitz liefert mit seiner Arbeit [Tagungsband S. 71-80] ein komplett neues Paradigma einer programmunbeeinflussbaren, auf psychometrischen (und unter Umständen weiteren) Merkmalen basierenden Authentifikation, die mit den Eingabemöglichkeiten eines berührrungssensitiven Sichtfelds auskommt.

Das berührungssensitive Sichtfeld wird in der vorgestellten Architektur in einer Art Sandbox durch die Sichtfeldkontrolleinheit geschützt, die auf dem Sichtfeld verschiedene Modi wie *Anwendungsmodus*, *Auswahlbestätigung* und *Installationsmodus* unterstützt. Das Sichtfeld wird dabei in Bereiche eingeteilt, die besonders sicherheitskritische Aktionen wie die Passworteingabe vom restlichen Betriebssystem abschirmen. Die Sichtfeldkontrolleinheit kann z.B. als FPGA realisiert werden, andere Entwurfsentscheidungen sind denkbar, lediglich die Programmunbeeinflussbarkeit durch Hauptprozessor und darauf laufendes Betriebssystem und Applikationen muss sichergestellt sein. Hat die Kontrolleinheit einen separaten Prozessor mit sequenziell ablaufendem Code anstatt einer verbindungsprogrammierten Lösung, so sollte besonders auf eine einfache, verifizierbare Architektur geachtet werden. Tritt man in eine sicherheitskritische Phase ein, z.B. den Start des Geräts oder die Installation einer App, wird der Hauptprozessor angehalten und die Sichtfeldkontrolleinheit übernimmt die Authentifizierung beispielsweise über psychometrische Merkmale, bei der, neben Kenntnis einer Geste oder eines Passworts, auch die Tipp- oder Wischgeschwindigkeit mit einbezogen und überprüft wird. Dazu können Merkmale wie Geolokation, der typische Gang des Besitzers, Besitz von NFC Chips (NFC Lesegerät muss auf programmunbeeinflussbare Weise an die Sichtfeldkontrolleinheit angeschlossen sein) oder beispielsweise Lebendmerkmale wie Blutdurck oder andere Vitalparameter, bei geeigneter Sensorausstattung, herangezogen werden. Bei Erkennung eines Angriffes oder Diebstahls können dann Maßnahmen, wie ein Absetzen des Notrufs, ergriffen werden.

Die im Vortrag vorgestellte Lösung sollte die Grundlage für eine konkrete Implementierung bieten. Es böte sich die Gelegenheit, eine tatsächlich sichere und gleichzeitig auf den Nutzerkomfort ausgerichtete Lösung über die Grundlagenforschung hinaus zu treiben.

4.4 Aktuelle Anwendungen

4.4.1 Automated testbed for various indoor position systems and sensors for evaluation and improvement

Jan-Gerrit Jaeger, der auch den Vortrag übernahm, veröffentlichte mit Christoph Brandau und Dietmar Tutsch, alle von der Universität Wuppertal, ein Paper [Tagungsband S. 81-88] zu ihrer Entwicklung eines Teststandes bzw. einer Plattform für die Evaluation von Indoor Positions Systemen. Das Problem der Geolokation außerhalb von Gebäuden scheint durch GPS und ähnliche Systeme gelöst, jedoch bleibt die Positionsbestimmung in geschlossenen Räumen eine weiterhin schwierige Herausforderung mit steigender Anzahl von Anwendungsfeldern z.B. in Lagersystemen. Die Arbeit schafft die Voraussetzung, um bestehende und zukünftige Systeme zu erproben, in einem wissenschaftlichen Rahmen zu vergleichen und gezielt weiterentwickeln zu können. Dementsprechend kann diese Ausarbeitung die Grundlage für viele zukünftige Entwicklungs- und Forschungsvorhaben darstellen.

Die Testplattform besteht aus einer 1,8x1,8m großen Fläche mit 30cm Spanplatten als Begrenzung, um Lidar und andere Sensoren eine geeignete Reflektionsfläche zu bieten, ohne das Entstehen von Geisterobjekten, durch z.B. spiegelnde Oberflächen aus Glas, zu riskieren. Die modulare Lösung setzt sich aus einer Sensorkomponente mit zwei Lidar (aktuell nur 2D Mapping), Lichtsensoren und dem Messfahrzeug, aktuell ein Lego Mindstorm Roboter, zusammen. Die Positionsberechnung erfolgt 30 mal pro Sekunde durch eine kabelgebundene serielle Schnittstelle. Die Kabel werden dabei oben aus dem Roboter heraus geführt, um die Bewegung und die Sensorik nicht zu stören. Der Lego Mindstorm kann mit den Helligkeitssensoren einer dunklen Linie folgen und macht dies auch auf dem markiertem Kurs bis zu seinem Startpunkt, wobei die Abweichung während des Kurses maximal 2,5cm und am Startpunkt 4mm bei einer Winkelabweichung von 0,5 Grad beträgt. Bei Bedarf einer exakteren Trajektorik kann das Sensormodul auch auf einen anderen Roboter montiert werden, da das ROS Betriebssystem der Sensorik softwareseitig nicht mit dem Lego Mindstorm verbunden ist. Die Pakete *Laser Scan Matcher* und *GMapping* erlauben das Erstellen einer Karte und eine Positionsbestimmung auf derselben. Das manuelle Erzeugen einer Karte unter Nutzung des Pakets *Adaptive Monte Carlo Localization* brachte keine Verbesserung dieses Setups. IPS verfolgen nun verschiedene Ansätze, z.B. kann Bluetooth 5.1 über einen Sensorarray den Eingangswinkel eines Signals bestimmen. Klassischerweise wird die Signallaufzeit zur Positionierung herangezogen. Zur Evaluation des Teststandes wurde ein von Herrn Jaeger mitentwickeltes Ultrabreitband IPS, basierend auf IEEE 802.15.4-2011, eingesetzt, wofür drei Anker an den Seiten des Teststandes installiert wurden, die über Triangulation eine Positionsbestimmung des Sensorchips erlauben. Hierfür muss eine simple Kreisgleichung gelöst werden und bei Ungenauigkeiten der Messung eine bestmögliche Approximation der Position gefunden werden. Es konnte gezeigt werden, dass die Genauigkeit des IPS durch die Kalibrierung mit der Testplattform deutlich verbessert werden konnte.

Für die Zukunft ist geplant, die Kabelverbindungen zu optimieren und das Ultrabreitband IPS zu erweitern. Zudem sollen Störeinflüsse von auf dem Teststand platzierten Hindernissen mit berücksichtigt werden.

4.4.2 Automatisierte Erkennung von Transportbehältern bekannter Versender

Roman Gumzej von der Universität Maribor stellte seine Arbeit [Tagungsband S.89-98] zur sicheren Identifikation von Transportbehältern zum Zwecke der Vermeidung der Manipulation von Transportgütern und zur Beschleunigung der Logistikkette vor. Der Vortrag war eher kurz gehalten, weswegen nachfolgende Ausführungen hauptsächlich auf dem Paper basieren. Die Arbeit stellt eine neue Architektur und Protokolle mit sicherer One-Time-Pad Verschlüsselung (im Text auch *Vernam-Chiffre* genannt) vor, um Transportbehälter im Physikalischen Internet (PI) sicher und möglichst autonom, mit geringem menschlichem Aufwand, basierend auf und im Einklang mit den EU-Verordnungen, über zum Teil etliche Zwischenstationen zu verschicken – das vorgestellte System ist dabei nicht auf ISO Standard-Container beschränkt, wobei darauf hingewiesen wird, dass diese wohl aufgrund bereits vorhandener Infrastruktur wie LKWs, Flugzeuge und Containerschiffe, den Haupteinsatzzweck darstellen werden.

Die EU-Verordnungen *Nr. 648/2005* und *185/2010* definieren die Begriffe *zugelassener Wirtschaftsbeteiligter* (Hersteller), *bekannter Versender* (Distributor) und *reglementierter Beauftragter* (Frachtzentren). Unter Einhaltung der dort genannten Regeln können Pakete im Transit, die von bekannten Teilnehmern verschickt und bearbeitet werden, schneller an Zwischenstationen umgeschlagen werden, da ein mehrfaches Prüfen der Fracht, z.B. um zu verhindern, dass Sprengstoff in ein Flugzeug geschmuggelt wird, vermieden wird. Um die Anforderungen an einen bekannten Versender in letztgenannter Verordnung zu erfüllen, stellt die Arbeit ein spezielles Datenverarbeitungs- und Kommunikationsgerät vor, das außen (der Wartbarkeit wegen) am Container angebracht und vor Versand verplombt wird. Sensoren überwachen die Bewegung des Containers, Druck/Temperatur im Container und Ultraschallsensoren die Unversehrtheit des Containers und der Datenverarbeitungseinheit, sowie die Integrität der Versiegelung des Containers. Verschiedene Techniken wie z.B. eine spezielle Folie zum Manipulationsschutz der Datenverarbeitungs- und Kommunikationseinheit kommen dabei zum Tragen. Über NFC und lokale Netze (zum Stromsparen nur aktiv wenn nötig) erfolgt die Kommunikation. Der Versender hinterlegt eine 256 Bit Zeichenkette im Nurlesespeicher, auf die ein Prozessor (Harvard-Architektur mit Nurlese-Programmspeicher und verifiziertem Code) zugreifen kann. Übermittelt wird immer nur eine durch eine zufällige Selektionsfunktion (Bitmaske) bestimmte Teilmenge des Identifikationsmerkmals und dies auch nur per One-Time-Pad verschlüsselter Nachricht. Der reglementierte Beauftragte im Frachtzentrum liest die (maskierte) Identifikation des Containers, fragt dabei erst alle bekannten Versender an und erhält von dem konkreten Versender den Schlüssel, um die vom Container erhaltene Nachricht zu dekodieren. Im Gegensatz zu klassischen QR-Codes oder Strichcodes ist dieser Prozess deutlich weniger anfällig für Manipulationen. Ist ein Container über dieses Verfahren eindeutig identifiziert, so kann er beschleunigt bearbeitet werden.

Die Technologie ist aufgrund ihrer hohen wirtschaftlichen Relevanz sicher erst der Anfang eines weitaus komplexeren vereinheitlichten Logistiksystem, welches Zeit und somit Kosten einsparen wird. Weitere Forschungsvorhaben auch unter Einbeziehung der Blockchain-Technologie zur Nachverfolgung der Transportkette sind gut vorstellbar.

4.4.3 Eine Komplexitätsmetrik basierend auf der kognitiven Wahrnehmung des Menschen

Dem Feld der Beurteilung der Komplexität eines Softwaresystems hat Daniel Koß mit seiner Arbeit einen großen Dienst erwiesen, indem er die Schwachstellen bisheriger Metriken, z.B. Codelines, zyklomatische Komplexität, Halstead-Metrik und statische Codeanalyse nach Lint aufzeigt und sich für eine Metrik unter Einbeziehung der menschlichen Kognition stark macht. Gerade weil Komplexitätsmetriken ein Messinstrument für Menschen sein sollen, muss die subjektive Wahrnehmung der Komplexität eines Softwaresystems unter Berücksichtigung der Erkenntnisse der Kognitionspsychologie, neben der rein mechanischen funktionellen Komplexität, eine Rolle spielen. Hierfür gibt es dank vorliegender Arbeit [Tagungsband S. 99-108] eine passende *Gesamtkomplexitätsmetrik* als Summe der *Domänenkomplexitätsmetriken* von Software-, Hardware- und Spezifikationsartefakten.

In den Kognitionswissenschaften konnte gezeigt werden, dass bei der Simultanerfassung von Objekten zum einen der gemeinsame Kontext der Objekte (z.B. aufsteigende Zahlenfolge), zum anderen auch die 'magische' Zahl von Objekten, je nach Autor und Anwendung in der Größenordnung 4-9 angegeben, die ein Mensch parallel im Arbeitsgedächtnis halten kann, wichtig sind. Herr Koß führt detailliert auf, was er als 'lesenden Geist' bezeichnet, die Verarbeitung von Wahrnehmung und Sprache, die je nach Sprachaufbau dazu führt, dass z.B. Italienisch leichter phonologisch erfasst werden kann als Chinesisch, wobei Italienisch auch weniger Ambiguität als das Englische aufweist – hier wird auch der Bogen zur Syntax und Semantik von Programmiersprachen geschlagen. Als weitere Grundlage für seine Metrik zieht er das Baddeleysche Arbeitsgedächtnismodell mit seinen vier Komponenten heran und leitet aus genannten Phänomenen und Beobachtungen die Kategorien des *hierarchischen* und des *inhaltlichen* Zugangs zum Verständnis eines Softwaresystems ab.
Die Komplexitätsmetrik auf Grundlage der kognitiven Wahrnehmung ist nun die gewichtete Summe folgender Parameter: die Komplexität informationstragender Elemente (genannt ***Informationsinhaltsmetrik***) durch $(\sum_{i=1}^{n_v} \frac{I_i}{Z_i})$, die Komplexität von der Idealmenge kognitiv erfassbarer Informationen (genannt ***Informationsmengenmetrik***) durch $(\sum_{i=1}^{n_v}(I_i - \mu))$, die hierarchische Komplexität (genannt ***Umfangsmetrik***) durch $(n_m + n_v)$ und die Abweichung von der Idealvorstellung hierarchischer Komplexität (genannt ***Ebenenmetrik***) durch $(\sum_{i=1}^{n_m}(E_i - \mu))$, wobei $n_v \in \mathbb{N}$ (Anzahl informationstragender Elemente), $I_i \in \mathbb{R}_0^+$ (Informationsgehalt), $Z_i \in \mathbb{R}^+$ (kognitive Zugänglichkeit), $n_m \in \mathbb{N}$ (Anzahl hierarchietragender Elemente), $\mu \in \{3, 4, 5\}$ ('magische' Zahlen), $E_i \in \mathbb{N}$ und $E_i \geq \mu$ (Anzahl hierarchietragender Elemente). Der Vortrag enthielt zudem ein anschauliches Beispiel dieser Metrik für einen Codeabschnitt.

Wie aus der anschließenden Fragerunde hervor ging, wäre eine empirische Studie mit Versuchspersonen nützlich, um zu bestimmen, wie diese vier Komponenten der Metrik im Verhältnis zueinander gewichtet werden sollen. Daniel Koß lässt daher die konkreten Werte für die Gewichtungsparameter für zukünftige Forschungsanstrengungen sinnvollerweise zunächst offen. Interessant wäre auch eine vergleichende Studie zur Akzeptanz der Metrik in Entwicklungsteams.

5 PEARL bei AEG/ATM – ein Erfahrungsbericht

Autor: Herr Peter Pielmeier

In den 1970er Jahren entwickelte AEG die Prozessrechner Reihe AEG 80. De kleine Version 80-10 war ein 16-Bit Rechner mit 64k Adressraum und maximal 1 MB Hauptspeicher, abgeleitet von einer Rechneranlage von General Electric. Der Rechner wurde hauptsächlich in der Anlagen- und Gebäude-Steuerung eingesetzt, aber auch für militärische Anwendungen. Die Bundeswehr beauftragte damals die AEG mit der Entwicklung eines PEARL Compilers. Als ich Anfang 1979 bei AEG eintrat, war das Basic PEARL System fertig und wurde bei ersten Projekten eingesetzt. Ich betreute das Laufzeitsystem, mein Kollege den Compiler, welchen er auch zum großen Teil selbst entwickelt hatte. Der PEARL Compiler wurde selbst in PEARL entwickelt und diente praktischerweise selbst als Testfall. Das Laufzeitsystem hingegen war komplett in Assembler geschrieben.

Da der Speicher der AEG 80 sehr klein war, wurde die Compilierung in 8 unterschiedliche Läufe aufgeteilt. Neben der Wartung wurde das PEARL System weiterentwickelt. Die Hardware der AEG 80 wurde um neue Adressierungsmöglichkeiten und komplexere Instruktionen erweitert. Außerdem entwickelten wir, wieder im Auftrag der Bundeswehr, einen Debugger für PEARL. Davor konnte man nur auf Maschinenebene testen. PEARL wurde allerdings nur in wenigen militärischen Projekten eingesetzt. Damals waren die Wege noch kurz: praktisch alle Anwendungsentwickler hatten unsere Telefonnummern und Patches wurden zügig ausgeliefert.

An ein ziviles Projekt kann ich mich noch erinnern: das System ARD-Stern für den Hessischen Rundfunk, eine Schaltung aller ARD Sender mit Zentrale in Frankfurt. Vor der flächendeckenden Verfügbarkeit des Internets waren TV-fähige Leitungen extrem teuer. Zu meiner grossen Überraschung habe ich letztes Jahr einen Mitarbeiter des HRs getroffen, der dieses System immer noch bedient, der zentrale Rechner ist immer noch eine Uralt AEG 80 wobei die neuen Anschlüsse an die bestehende Peripherie angeflanscht wurden.

Später habe ich für andere IT Firmen gearbeitet und erst da erkannt, wie lächerlich klein unser Entwicklungs-Team damals war. Dafür war das Ergebnis gar nicht schlecht und mir hat es viel Spass gemacht, praktisch alles vom Design über Entwicklung, Test und Wartung durchzuführen. Ich persönlich denke gerne an PEARL zurück.

6 AK OpenPEARL Compiler

Autoren: Rainer Müller (HS Furtwangen) und Marcel Schaible (FernUniversität in Hagen)

Seit dem letzten Rundbrief wurde tatkräftig weiter an dem OpenPEARL Compiler gearbeitet und die noch fehlenden Strukturen (STRUCT) implementiert. Daneben wurden noch eine Reihe von Fehlern behoben. Als die nächsten größeren Schritte werden die Unterstützung von mehreren Modulen und die Signale angegangen.

6.1 Offene Punkte im Compiler

Folgende Sprachkonstrukte sind noch nicht (vollständig) umgesetzt:

- REF abschliessen
- TYPE
- REF CHAR()

6.2 Aufruf zum Mitmachen

Die aktive Beteiligung von weiteren Mitstreitern, sei es bei der Dokumentation, der Erstellung eines Tutorials oder der allgemeinen Verbesserung, insbesondere der Testabdeckung des Übersetzers, ist dringend notwendig und jederzeit willkommen.

6.3 Studentische Arbeiten

Seit dem letzten Rundbrief haben Studierende folgende Themen bearbeitet:

6.3.1 Fehlersuche und Visualisierung der Belegung von Synchronisationsmitteln in nebenläufigen Systemen

Bei der Entwicklung von nebenläufigen Anwendungen spielt die Synchronisation von gemeinsam genutzten Betriebsmitteln eine wichtige Rolle. Fehlerhafte Implementierungen können zu Verklemmungen (engl. deadlocks) führen. Um die Fehlersuche in OpenPEARL Programmen zu verbessern wurde durch eine studentische Arbeit zunächst die Möglichkeit des Aufzeichnens der Semaphor- und Bolt-Belegungen geschaffen. Hierzu wurde das Laufzeitsystem an den entsprechenden Stellen attributiert. Während des Programmablaufes werden dann die Belegungen in eine Trace-Datei zu späteren Analyse gespeichert. Zur Analyse wurde zusätzlich ein Werkzeug zu graphischen Darstellung der Belegung über der Zeit geschaffen.

6.3.2 I2C-Bus Unterstützung für den ESP32

In einer Bachelorarbeit im WS20/21 wurde die Portierung von OpenPEARL auf dem ESP32 Controller um die I2C-Schnittstelle erweitert. Damit sind alle bisher in OpenPEARL unterstützten I2C-Geräte auch am ESP32 nutzbar.

6.3.3 Pfadanalyse in PROC und TASK

In einer aktuellen Bachelorarbeit wird der mögliche Kontrollfluss untersucht um nicht erreichbare Programmstellen zu lokalisieren. Dazu wird aus dem abstrakten Syntaxbaum der PEARL-Anwendung ein Kontrollflussgraph erstellt und anschließend untersucht. Nach aktuellem Stand funktioniert dies. Nun wird versucht Programmstellen zu lokalisieren, die aufgrund von Variablenbelegungen nicht erreichbar sind.

6.3.4 Jenkinserweiterung für Code- und Test-Qualität und Laufzeittests am ESP32

In einer aktuellen Masterarbeit wird die Qualität des Quellcodes des Compilers und Laufzeitsystems untersucht. Dabei kommen sowohl unter anderem stilistische Metriken wie Schreibstil zum Einsatz, wie auch Metriken zur Testüberdeckung. Diese Tests werden in eine Jenkinsumgebung integriert. Eine Herausforderung ist hierbei der automatisierte Test von Anwendungsprogrammen am ESP32, um auch von dort Informationen über Testergebnisse in den Zustandsbericht zu integrieren und auch die erreichte Testüberdeckung zu erfassen.

6.3.5 Testsuite für Fehlermeldungen des Compiler

Für den Test des Compilers auf erwartete Fehlermeldungen wurde ein kleines Werkzeug entwickelt, bei dem mit Hilfe von speziellen Kommentaren die Vollständigkeit und Korrektheit von Fehlermeldungen automatisiert überprüft werden kann. Der Entwurf von Testfällen im Bereich der Deklarationen ist Teil einer aktuellen Bachelorarbeit.

Beispiel:

```
DCL so DATION OUT ALPHIC FORWARD;

/*$ ^ ERROR : Syntax error:mismatched input ';' expecting {'GLOBAL', 'CREATED'} */
```


EchtZeit