

Echt Zeit

Nr. 8, September 2018

Mitteilungen
des GI/GMA/ITG-Fachausschusses
Echtzeitsysteme



GESELLSCHAFT FÜR INFORMATIK E.V.



VDI/VDE-Gesellschaft
Mess- und Automatisierungstechnik

ITG INFORMATIONSTECHNISCHE
GESELLSCHAFT IM VDE

Impressum

Herausgeber GI/GMA/ITG-Fachausschuss Echtzeitsysteme
<http://www.real-time.de>

Sprecher Prof. Dr.-Ing. habil. Herwig Unger
FernUniversität in Hagen
Lehrstuhl für Kommunikationsnetze
58084 Hagen
herwig.unger@fernuni-hagen.de

Stellvertreter Prof. Dr. Dieter Zöbel
Universität Koblenz-Landau
Institut für Softwaretechnik
56016 Koblenz
zoebel@uni-koblenz.de

Redaktion Prof. Dr.-Ing. habil. Herwig Unger
Dipl.-Ing. Jutta Düring
FernUniversität in Hagen
58084 Hagen
echtzeit@fernuni-hagen.de

ISSN 2199-9244

Redaktionell abgeschlossen am 24. September 2018

Einreichung von Beiträgen:

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

In dieser Ausgabe:

- 1 Workshop Echtzeit 2018: Programm und Aufruf zur Teilnahme
- 2 11th Conference on Autonomous Systems 2018
- 3 Graduiertenwettbewerb 2018
- 4 Die neue Norm DIN 66253 „SafePEARL“
- 5 Zufallsgesteuerter Testfallgenerator
- 6 Bekanntmachung auf WikiCFP:

1 Workshop Echtzeit 2018: Programm und Aufruf zur Teilnahme

Jutta Düring, Fachausschuss Echtzeitsysteme

Der Workshop „Echtzeit 2018“ mit dem diesjährigen Leitthema „**Echtzeit und Sicherheit**“ findet am 15. und 16. November 2018 wie gewohnt in Boppard am Rhein statt.

Der Tagungsband erscheint in der Reihe „Informatik aktuell“ des renommierten Springer-Verlages. In Anbetracht des interessanten Tagungsprogrammes und der anregend-freundlichen Atmosphäre sollten Sie sich unter <https://www.real-time.de/echtzeit.html> zur Teilnahme anmelden. Stichtag für den Frühbuchertarif ist der 14. Oktober 2018.

Erster Workshop-Tag: Donnerstag, der 15. November 2018

13:00 Uhr Begrüßung

13:10 Uhr **Eröffnungsvortrag: Von Algol68 zu SafePEARL**
Wolfgang A. Halang, Marcel Schaible

13:50 Uhr Pause (ohne Kaffee)

14:00 Uhr **Sitzung 1: Funktionale Sicherheit** (Leitung: Frau Schiedermeier)

Funktionale Sicherheit von autonomen Transportsystemen in flexiblen Fertigungsumgebungen

Philip Kleen, Janis Albrecht (Fraunhofer IOSB-INA)

Secure Real-time Communication

Dimitrios Savvidis, Dietmar Tutsch (Bergische Universität Wuppertal)

Entwicklungsvorschläge für ISO 26262 konforme MCUs in sicherheitskritischer Avionik

Georg Seifert, Sebastian Hiergeist, Andreas Schwierz (Technische Hochschule Ingolstadt)

15:30 Uhr Pause

16:00 Uhr **Sitzung 2: Lehre** (Leitung: Frau Benra)

Autonomes Fahren in der Lehre

Andreas Werner, Robert Kaiser (Hochschule RheinMain)

Automatische Evaluierung von Anforderungen bezüglich der Informationssicherheit für das zukünftige industrielle Netzwerkmanagement

Marco Ehrlich, Henning Trsek, Jürgen Jasperneite (Hochschule Ostwestfalen-Lippe)

SWAN: Systemweite statische Laufzeitanalyse echtzeitfähiger Betriebssysteme

Simon Schuster (Friedrich-Alexander Universität Erlangen-Nürnberg)

17:30 Uhr Preisverleihung

17:45 Uhr Abendessen

19:30 Uhr Mitgliederversammlung des Fachausschusses Echtzeitsysteme

Zweiter Workshop-Tag: Freitag, der 16. November 2018

9:00 Uhr **Sitzung 3: Applikationen von Echtzeitsystemen** (Leitung: Herr Cseke)

Spezifikation projektspezifischer Software

Jens Lehmann (OntoTec GmbH)

Timekeeper – Zeiterfassung mittels RFID und Raspberry Pi

Denise Papaioannou, Mario Kubek (Fachhochschule Südwestfalen, FernUniversität in Hagen)

Schutz automatisierungstechnischer Programme vor Umkehrentwicklung

Sergej Gertje (FernUniversität in Hagen)

10:30 Uhr Pause

11:00 Uhr **Sitzung 4: Sicherheitsgerichtete Echtzeitsysteme** (Leitung: Herr Halang)

Latenzen von POSIX Betriebssystemen im Kontext von Hypervisoren in Real-Time Systemen

Ludwig Thomeczek, Andreas Attenberger, Václav Matousek, Jürgen Mottok (Continental Automotive GmbH, OTH Regensburg, Universität Pilsen)

fastAN(BD) - eine Methode zur schnellen Dekodierung und Integritätsprüfung ANBD-kodierter Daten

Stefan Widmann (FernUniversität in Hagen)

Parametrierbare Übergabeschnittstellen im Entwurfsprozess für sicherheitsgerichtete Systeme

Daniel Koß (FernUniversität in Hagen)

12:30 Uhr Verabschiedung

12:45 Uhr Mittagsimbiss (optional)

2 11th Conference on Autonomous Systems 2018

Jutta Düring, Fachausschuss Echtzeitsysteme

Homepage: <http://www.confautsys.org>

Die Konferenz „Autonomous Systems“ findet vom 28. Oktober bis 2. November 2018 in Cala Millor auf Mallorca statt. Der Tagungsband erscheint in den „Fortschritt-Berichten“ des VDI-Verlages.

Monday, 29th October 2018

09:00 Uhr **Keynote: Prof. Dr. Chunrong Yuan, Technische Hochschule Köln**
Mobile Autonomous Systems: Sensing, Reasoning and Acting

10:00 Uhr Coffee Break

- 10:30 Uhr **Session 1:**
NATURAL LANGUAGE PROCESSING AND TEXT MINING
An Associative Ring Memory to Support Decentralised Search
Herwig Unger, Mario Kubek

Word embeddings and semantic indexing techniques: speed, features and suitability
Thomas Efer

Knowledge Discovery of Topic Evolution using Topic Modeling Technique
Akara Prayote

NLP of Microblog Name Entity Recognition
Maleerat Sodanil
- 12:30 Uhr Lunch
- 14:00 Uhr **Session 2: THEORY I**

How tall can be a Swiss Guardian, before he loses control?
Gerd K. Heinz

Distributions of Points
Hanno Lefmann

Blind Censoring for Instant Messaging
Günter Fahrnberger
- 15:30 Uhr Coffee Break
- 16:00 Uhr **Session 3: MACHINE LEARNING AND DATA MINING**

Machine Learning in Amazon Webservices
Jürgen Nützel

Time Series Imputation and Prediction Based on Machine Learning
Phayung Meesad, Kornsirint Rojanawan

The trend of AI technologies in Image Processing
Hong Zhu

N.N.
Gerhard Sartorius
- 18:30 Uhr Together Dinner
- 19:30 Uhr Informal Discussion Rounds

Tuesday, 30th October 2018

Social Event: Tagesausflug

Wednesday, 31st October 2018

- 09:00 Uhr **Keynote: Prof. Dr. Phayung Meesad**
King Mongkut's University of Technology North Bangkok
Deep Learning and Applications
- 10:00 Uhr Coffee Break

- 10:30 Uhr **Workshop/Tutorial for PhD Students, Part I: „Functional Programming Languages and Embedded/Real-Time Systems: Do they fit?“**
Prof. Dr. Theodor Tempelmeier
- 12:30 Uhr Lunch
- 14:00 Uhr **Workshop/Tutorial for PhD Students, Part II: „Functional Programming Languages and Embedded/Real-Time Systems: Do they fit?“**
Prof. Dr. Theodor Tempelmeier
- 15:30 Uhr Coffee Break
- 16:00 Uhr **Session 4: ENERGY**
A novel Microgrid coined
Zhong Li

Design, Analysis and Implementation of High-Step-Up Converters in Renewable Energy Systems
Guidong Zhang, Zhiyang Wang and Yun Zhang

Research on Information Network Vulnerability of Intelligent Substation
Ruiwen He
- 18:30 Uhr Together Dinner
- 19:30 Uhr Informal Discussion Rounds

Thursday, 1st November 2018

- 09:00 Uhr **Key Presentation:**
UNBOXING INNOVATIVE WEB SEARCH APPLICATIONS
Introducing DocAnalyser V2 – (Re)Searching With Web Documents
WebEngine – The First Integrated, Decentralised Web Search Engine
Mario Kubek, Herwig Unger
- 10:00 Uhr Coffee Break
- 10:30 Uhr **Session 5: THEORY II**
Where being sloppy pays off
Thomas Böhme

A Grid-Fault Ride-Through Control Strategy of Variable Frequency Transformer with SCCs Under Unbalanced Networks
Sizhe Chen

A Fully Neurocomputing based Traffic Modelling-and-Simulation Concept
Nkiediel Alan Akwir, Muhindo Kule Mutengi, Witesyavwirwa Vianney Kambale, Jean Chamberlain Chedjou and Kyandoghere Kyamakya

Graph Theoretical Problems in Traffic Management – A Brief Survey
Nkiediel Alan Akwir, Muhindo Kule Mutengi, Witesyavwirwa Vianney Kambale, Jean Chamberlain Chedjou and Kyandoghere Kyamakya
- 12:30 Uhr Lunch

14:00 Uhr **PhD Student Presentations (25 min. each) by:**

Mark Hloch

Yonchanok Khaokaew

Further Short Presentations (5 min. each)

15:30 Uhr Coffee Break

16:00 Uhr **Session 6: ARCHITECTURE AND APPLICATIONS**

fastAN(BD) – a Fast Method for Integrity Checking and Decoding of AN(BD)-coded Data

Stefan Widmann

Notes on the Design of a Statically Safe Microprocessor

Marcel Schaible

Architecture for Trust-based Machine to Machine Communication

Christoph Maget

18:30 Uhr Together Dinner

19:30 Uhr Informal Discussion Rounds

Friday, 2nd November 2018

9:00 Uhr Individual Discussions

11:00 Uhr Departure following the private schedules

3 Graduiertenwettbewerb 2018

Jutta Düring, Fachausschuss Echtzeitsysteme

Wir gratulieren dem Preisträger des Graduiertenwettbewerbs 2018:

- **Simon Schuster** schloss im November 2017 sein Studium der Informatik an der Friedrich-Alexander Universität Erlangen-Nürnberg mit der Masterarbeit ab. Das Thema der prämierten Arbeit lautet „*Systemgewahre statische Laufzeitanalyse von Universalbetriebssystemen*“.

4 Die neue Norm DIN 66253 „SafePEARL“

Wolfgang A. Halang

Homepage: <https://www.din.de/de/wdc-beuth:din21:284094216>

Im März 2018 trat die neue Norm DIN 66253 „SafePEARL“ in Kraft, die eine – völlig neue und unter Programmiersprachen weltweit einzigartige – sicherheitsgerichtete Echtzeitprogrammiersprache basierend auf PEARL90 (DIN 66253-2) und Mehrrechner-PEARL (DIN 66253 Teil 3) definiert. Gefördert vom Bundeswirtschaftsministerium im Rahmen des Programms „Transfer von Forschungs- und Entwicklungsergebnissen durch Normung und Standardisierung“ wurde sie im Projekt „Normung der Echtzeitprogrammiersprache PEARL hinsichtlich funktionaler Sicherheit“ von Mitgliedern des GI/GMA/ITG-Fachausschusses Echtzeitsysteme erarbeitet. Gemäß

Beschluss 2016/09/15-01 des DIN-Normenausschusses NA 043-01-22 AA „Programmiersprachen“ vom 15. September 2016 wurden DIN 66253 Teil 3 und DIN 66253-2 mit Inkrafttreten der neuen Norm zurückgezogen und durch diese ersetzt.

4.1 Entwicklungsgeschichte

Die Entwicklung der deutschen Echtzeitprogrammiersprache PEARL (Process and Experiment Automation Realtime Language) geht auf den Wunsch zurück, automatisierungs- und kerntechnische Anwendungen angemessen und mit hoher Produktivität formulieren zu können. Schon um das Jahr 1967 herum nahmen deshalb Ingenieure und Physiker von Firmen und Forschungsstellen die Arbeit an der Sprachspezifikation auf, die im April 1973 dann als erster Forschungsbericht des vom Bund geförderten Programms Prozessdatenverarbeitung (PDV) veröffentlicht wurde. Unmittelbar danach wurden erste Übersetzer erstellt und es begann auch die Normung von PEARL, wozu 1974 der Arbeitskreis DIN/FNI AK 5.8 gegründet wurde. Aus seinen Arbeiten gingen eine einfache und eine recht komplexe Version der Sprache hervor, die in den Teilen 1 und 2 der DIN 66253 als „Basic PEARL“ 1981 und als „Full PEARL“ 1982 genormt wurden. Daran schloss sich im Jahre 1989 der Teil 3 „Mehrrechner-PEARL“ an, der durch konzeptionelle Klarheit und Eleganz besticht und ein bis heute von anderen unerreichtes Muster zur Strukturierung und Programmierung verteilter Systeme darstellt.

Mit PEARL hatte die deutsche Informationstechnik auf dem Gebiet der Echtzeitsysteme eine international führende Stellung erreicht, denn die Sprache ist ihren Konkurrenten wegen der Klarheit und niedrigen Komplexität ihrer Konstrukte sowie ihrer inhärent sicherheitsfördernden leichten Lesbarkeit und Verständlichkeit technisch weit überlegen. Deshalb haben ihre Konzepte auch bei der Standardisierung von Industrial Real-Time FORTRAN, einer Sammlung von Unterprogrammen zur Erweiterung von FORTRAN um Echtzeitfähigkeit, durch den International Purdue Workshop on Industrial Computer Systems Pate gestanden. Allein die französische Entwicklung LTR reichte in etwa an PEARL heran, ist jedoch leider verschwunden. Auf dem Weltmarkt ist als einzige mit PEARL konkurrierende Sprache Ada verblieben, die sehr komplex ist und deren Echtzeitfähigkeiten nur schwach ausgeprägt sind. Die im Laufe von 25 Jahren gemachten Erfahrungen fanden 1998 in der Norm DIN 66253-2 ihren Niederschlag, mit der Basic und Full PEARL durch PEARL90 abgelöst wurden.

4.2 Charakterisierung des bisherigen PEARL

Die bisherigen Versionen von PEARL erfüllen die allgemeinen Anforderungen an strukturierte Hochsprachen für den Bereich eingebetteter Echtzeitsysteme. Einerseits bieten sie die üblichen Konstrukte technisch-wissenschaftlich orientierter Sprachen mit besonders strenger Typprüfung an und andererseits stellen sie komplette Instrumentarien zur Definition, Einplanung und Synchronisation nebenläufiger Prozesse und zur Kommunikation mit konventioneller und Prozessperipherie bereit. Um die Entwicklung großer Software-Pakete zu unterstützen, bestehen PEARL-Programme aus separat übersetzbaren Modulen, die sich selbst wiederum aus System- und Problemteilen zusammensetzen. Zur Förderung der Portabilität kapseln erstere die notwendigen Hardware-Beschreibungen und die Zuordnung symbolischer Namen zu Geräten ein, die dann von den eigentlichen, in den Problemteilen enthaltenen Automatisierungsprogrammen verwendet werden. Möglichkeiten zur Bitverarbeitung, zur Behandlung von Ausnahmezuständen, zur

Verwaltung und Maskierung von Unterbrechungen sowie zur Testunterstützung gehören ebenfalls zum Sprachumfang von PEARL.

Die nebenläufige Verarbeitung asynchroner Rechenprozesse realisiert PEARL mit dem Konzept der Task. Als Elemente paralleler Ausführung werden Tasks nach einem sehr einfachen und fast minimalen, aber trotzdem vollständigen Modell mit den vier Zuständen bekannt, bereit, laufend und zurückgestellt verwaltet. Zustandsübergänge in diesem Modell werden durch externe, interne, Synchronisations- und Zeitereignisse veranlasst. Wegen der Asynchronität der Task-Abläufe müssen sich diese bspw. beim Zugriff auf gemeinsam genutzte Betriebsmittel synchronisieren, wozu PEARL klassische und verallgemeinerte Semaphore als Sprachkonstrukte anbietet.

Von anderen höheren Programmiersprachen unterscheidet sich PEARL besonders deutlich durch explizite Unterstützung des Zeitbegriffs. Die beiden Datentypen CLOCK für Zeitpunkte und DURATION für Zeitdauern stehen zusammen mit entsprechenden arithmetischen Operationen vordefiniert zur Verfügung. Während Ada als einzigen Zeitbezug nur die inhärent ungenaue Verzögerung des Ablaufs von Tasks kennt, die diese innerhalb ihrer Rümpfe selbst verwalten müssen, werden die Aktivierungen von PEARL-Tasks im Hauptprogramm in problemorientierter Weise eingeplant, und zwar einerseits als sporadische Reaktionen auf ankommende Unterbrechungssignale, auf Erreichen absoluter Zeitpunkte oder auf Verstreichen von Wartezeiten oder andererseits periodisch mit in verschiedenen Formen anzugebenden Frequenzen und Geltungsdauern. Hier zeigt PEARL eine besondere Stärke. So benötigt man in der für die Programmierung eingebetteter Systeme derzeit vorherrschenden, jedoch nicht echtzeitfähigen Sprache C im Zusammenwirken mit einem Betriebssystem überflüssig komplizierte, semantisch intransparente und mehrere Dutzend Anweisungen umfassende Codesegmente, um die periodische Aktivierung einer Task einzuplanen, wozu in PEARL eine einzige, auch für Nichtfachleute unmittelbar verständliche und selbstdokumentierende Anweisung ausreicht.

Genau wegen dieser Nähe von PEARL-Code zu Klartext eignet sich PEARL nicht nur zur eigentlichen Programmierung, sondern auch schon zur Spezifikation von Automatisierungsanwendungen und insbesondere für die Lehre. In PEARL-Darstellung lassen sich Denkweise und Konzepte der Echtzeitprogrammierung sehr leicht vermitteln und verstehen. Das hat positive Konsequenzen auf Qualität und Sicherheit der später von den Absolventen erstellten Systeme. Derzeit wird PEARL an rund 50 Hochschulen in Deutschland und Europa für die Ingenieurausbildung eingesetzt. PEARL erreicht somit einen hohen Grad an Ergonomie durch Unterstützung harmonischer Mensch-Maschine-Interaktion.

4.3 Überblick über SafePEARL

Im Hinblick auf Einsetzbarkeit zur Programmierung sicherheitskritischer Steuer- und Regelungsaufgaben in (verteilten) Echtzeitumgebungen und mit dem Ziel, rechnergestützte Automatisierungssysteme beherrschbarer und damit einen Schritt hin zur Ermöglichung ihrer sicherheitstechnischen Abnahme zu machen, werden in der neuen Norm bei Beschränkung auf das Notwendigste für solche Aufgaben wirklich erforderliche Sprachkonstrukte in knapper tabellarischer Form identifiziert und inhärent sichere, ineinander geschachtelte Teilmengen davon gebildet, die den jeweiligen Anforderungen der Sicherheitsintegritätsstufen SIL1, SIL2, SIL3 und SIL4 nach DIN EN 61508 (VDE 0803) genügen. Der dabei verfolgte und bereits beim Schritt von Ada zu SPARK erfolgreich verwendete Ansatz ist, den Einsatz als unsicher erkannter Sprachmerkmale schrittweise auf höheren Stufen zu verbieten. So muss nicht für jede Sicherheitsintegritätsstufe

eine neue Sprache definiert und gelernt werden. Außerdem können Übersetzer prüfen, ob Programme die für die jeweils gewählte Stufe geltenden Sicherheitsauflagen erfüllen. Das Prinzip, Teilmengen einer Sprache für kritische Anwendungen zu definieren, gestattet es, Programme nach bestimmten Sicherheitsanforderungen zu entwickeln und Code für sicherheitskritische und -unkritische Systemteile nahtlos miteinander zu verbinden.

Mit SafePEARL und seinen geschachtelten, spezifischen Teilmengen für die vier Sicherheitsintegritätsstufen wurde eine einzigartige Sprache zur Programmierung verteilter Echtzeitsysteme geschaffen, die sämtliche bekannten Sprachmittel zur Förderung funktionaler Sicherheit in sich vereinigt und sich an der menschlichen Verständnisfähigkeit orientiert. Betrachtet als sozialer Prozess zur Erreichung eines Konsenses, wird Programmverifikation durch Merkmale wie Zusammensetzung und Wiederverwendung zugelassener Komponenten, Programmierung auf der Spezifikationsebene durch Erstellen von Entscheidungstabellen und allgemein durch das Bemühen erleichtert, in allen Aspekten äußerste Einfachheit zu erreichen. Die Technischen Überwachungsvereine können in der Sprache geschriebene Software mit großer Vertrauenswürdigkeit und vertretbarem Aufwand prüfen. Anwendung der Sprache zur Entwicklung eingebetteter Systeme verspricht sowohl das Risiko für Menschenleben, Umwelt und Anlagen als auch die Wartungskosten zu senken, weil durch die inhärent sicheren Sprachkonstrukte von vornherein weniger Fehler gemacht werden.

Um den Anforderungen von SIL1 zu genügen, sind in der entsprechenden Sprachteilmenge Konstrukte nicht mehr zugelassen, deren Anwendung zu nichtdeterministischem Verhalten führt, denn nur voll deterministisches Systemverhalten kann letztendlich die sicherheitstechnische Abnahme programmgesteuerter Geräte für sicherheitskritische Aufgaben ermöglichen. Vorhersehbarkeit des Systemverhaltens ist von zentraler Bedeutung für den Echtzeitbetrieb. Sie ergänzt die Forderung nach Rechtzeitigkeit, da letztere nur dann garantiert werden kann, wenn das Systemverhalten exakt vorhersehbar ist, und zwar sowohl in der Zeit als auch bzgl. der Reaktionen auf externe Ereignisse. Auf der Sicherheitsintegritätsstufe SIL1 sind Sprachelemente zur Festsetzung von Fertigstellungsfristen von Prozessabarbeitungen, zur Behandlung von Fristüberschreitungen, zur Begrenzung der Anzahl von Schleifendurchläufen, zur Zeit- und Ereignisüberwachung, zur Unterstützung der Abschätzung von Prozesslaufzeiten und Anwendung zeitgerechter Zuteilungsalgorithmen, zur allmählichen Leistungsabsenkung im Fehlerfall, zur Handhabung transienter Überlast sowie zur Entdeckung und Behandlung von Fehlern mittels Diversität und allmählicher Leistungsabsenkung vorgesehen. Die nur unsicher zu handhabenden Semaphore und Bolts sind durch neue Synchronisationskonstrukte zum Schutz von Betriebsmitteln ersetzt, die die Möglichkeiten bieten, maximale Wartezeiten vor kritischen Abschnitten und maximale Aufenthaltszeiten in diesen anzugeben, entsprechende Ausnahmen zu behandeln und Systemverklebungen zu vermeiden.

Für den Einsatz von SafePEARL in Umgebungen, die den Anforderungen nach SIL2 genügen müssen, wird der Gebrauch dynamischer Sprachelemente wie variabler Felddimensionen und rekursiver Prozeduraufrufe zugunsten von Verlässlichkeit unterbunden, denn diese dynamischen Konstrukte führen im Allgemeinen zu Unvorhersagbarkeit im Hinblick auf Kapazität und Zeitanforderungen, was die Arbeit zeitgerechter Zuteilungsalgorithmen unmöglich macht.

Speicherprogrammierbarere Steuerungen spielen in der Automatisierungstechnik eine große Rolle. Für sie lässt sich auf der Grundlage vorprogrammierter, sicherheitszertifizierter Funktionsblockbibliotheken mit Hilfe der Funktionsplansprache nach DIN EN 61131-3 leicht verifizierbare Software für sicherheitskritische Anwendungen erstellen. Die der Sicherheitsintegritätsstufe SIL3 zugeordnete Teilmenge von SafePEARL erlaubt, Funktionspläne textuell aufzustellen. Die

Teilmenge kennt im Wesentlichen nur noch Prozeduraufrufe und Parameterübergaben. Schleifen sind bspw. nicht mehr erlaubt. Die Norm DIN EN 61131-3 definiert weiterhin die spezielle, unter Programmiersprachen einzigartige Ablaufplansprache zur Partitionierung von Programmorganisationseinheiten, d.h. Programmen und Funktionsblöcken, in Schritte und deren Verbindung entlang gerichteter Kanten mittels Transitionen. Mit jedem Schritt ist eine Menge von Aktionen und mit jeder Transition eine Übergangsbedingung assoziiert. Deshalb sind in SafePEARL neue Sprachkonstrukte zur Formulierung sequentieller Ablaufsteuerungen definiert, die aus Sicherheitsgründen die Möglichkeit paralleler Abläufe explizit ausschließen.

Die der höchsten Sicherheitsintegritätsstufe SIL4 zugeordnete Teilmenge von SafePEARL umfasst neu eingeführte Konstrukte zur Formulierung von Ursache-Wirkungstabellen. Diese eignen sich besonders gut für die sicherheitsgerichtete Programmierung, weil sie streng, aber leicht nachvollziehbar und mit höchster Vertrauenswürdigkeit verifiziert werden können.

Bei der in die neue Norm überführten Spracherweiterung Mehrrechner-PEARL handelt es sich um die weltweit einzige Normierung einer Sprache zur Programmierung verteilter Systeme. Sie stellt Möglichkeiten zur Beschreibung von Konfigurationen, zur Parameterversorgung von Übersetzern, Bindern und Ladern, ohne dabei ausführbaren Code zu erzeugen, zur Kommunikation zwischen Rechnerknoten mittels Nachrichtenaustausch sowie zur dynamischen Rekonfiguration zur Verfügung. Die Architekturbeschreibung eines verteilten Systems umfasst Angaben über seine konstituierenden Rechnerknoten bzw. Einzelprozessorsysteme, deren Betriebszustände, die physikalischen Netzverbindungen, die Peripherieanschlüsse an die einzelnen Knoten sowie über Konfiguration und Verteilung der Software. Das Grundelement der Software-Verteilung ist die Collection, die aus einer Gruppe von Modulen, d.h. aus einem auf einem Einprozessorsystem lauffähigen Programm besteht. Einem Rechnerknoten ist als Software immer genau eine Collection zugeordnet. Mithin ist die Collection auch die Austauschereinheit bei dynamischen Rekonfigurationen, die Reparatur, Wartung und allmähliche Leistungsabsenkung im Rahmen von Fehlerbehandlungen unterstützen. Collections kommunizieren untereinander mit Hilfe von Ports, an die sie Nachrichten senden bzw. von wo sie diese empfangen. Es wird sowohl asynchrone Kommunikation als auch synchrone mit Wartezeitüberwachung und Ausweichaktionen unterstützt. Zulässige Kommunikationsstrukturen sind 1:1, 1:n und n:1.

4.4 Anwendungsbereich

Durch ihre Fokussierung auf Entwicklung eingebetteter, echtzeitfähiger Systeme unter besonderer Berücksichtigung der nötigen Kommunikationsschnittstellen im verteilten Umfeld stellt die Norm eine geeignete Basis für die Programmierung vernetzter eingebetteter Systeme dar, die auf mittels Software implementierter Steuerung oder Regelung über (globale) drahtgebundene oder drahtlose Kommunikationsnetze beruhen. Diese sogenannten cyber-physikalischen Systeme zeichnen sich durch einen in der Regel hohen Vernetzungsgrad ihrer mechanischen, elektronischen und informatischen Komponenten aus und können in mannigfaltigen Bereichen wie in dynamischen Automatisierungssystemen oder in Prozess-, Verkehrs- und Energieversorgungssystemen zum Einsatz kommen.

Hierbei spielen die Aspekte funktionale Sicherheit und Zuverlässigkeit eine entscheidende Rolle, denn programmierbare elektronische Systeme werden zunehmend in einer Vielzahl sicherheitsgerichteter Anwendungen eingesetzt. Sie übernehmen Aufgaben zur Überwachung oder Steuerung

von medizinischen Geräten, chemischen Anlagen, Anti-Blockier-Systemen, Luft- oder Weltraumfahrzeugen, Maschinen in der Fertigungstechnik sowie Kraftwerken und Energieverteilungssystemen. Diese Überwachungs- und Steuerungssysteme müssen deshalb hohe Sicherheitsanforderungen erfüllen, da falsch erstellte Systeme oder schlichtweg Fehler darin zum Versagen der Systemfunktionen führen können. Dies kann schwere Schäden verursachen oder gar Menschenleben gefährden. Der industrielle Bedarf an sicherheitsgerichteten, programmgesteuerten Systemen ist darum hoch und wird durch die fortschreitende Automatisierung von Prozessen in Zukunft noch weiter deutlich zunehmen. Hinzu kommt das steigende Sicherheitsbewusstsein in der Gesellschaft. Der so umrissene Bedarf, rechnergestützte Systeme mit hinreichendem Vertrauen in ihre Verlässlichkeit erstellen zu können, der ihre Zulassung für sicherheitskritische Steuer- und Regelaufgaben durch die Aufsichtsbehörden auf der Basis formeller Abnahmen erlaubt, war Anlass zur Entwicklung dieser an funktionaler Sicherheit orientierten Norm.

5 Zufallsgesteuerter Testfallgenerator

Marcel Schaible, FernUniversität in Hagen

marcel.schaible@fernuni-hagen.de

Projekt-Homepage: <https://sourceforge.net/projects/openpearl/>

5.1 Einleitung

Übersetzer für Programmiersprachen müssen zuverlässig korrekten ausführbaren Maschinencode erzeugen. Mittels umfangreicher Testsuiten wird versucht dieses Ziel zu erreichen. Übersetzer sind zwar theoretisch verstanden, in der Praxis aber oft, insbesondere durch die teilweise konkurrierenden Optimierungsziele, kompliziert und damit anfällig für Fehler. Um die Testabdeckung des OpenPEARL Übersetzers [1] weiter voranzutreiben soll ein Testfallgenerator, welcher nach vorgegeben Quell-Code-Schablonen zufällig syntaktisch und semantisch korrekte Programme erzeugt, erstellt werden. Zufallsgesteuerte Testfallgeneratoren (ZTG) werden erfolgreich bei vielen Open-Source Übersetzer wie z.B. [3] und CLANG/LLVM eingesetzt.

5.2 Zufallsgesteuerter Testfallgenerator

Die Testfälle werden unter Zuhilfenahme der OpenPEARL Grammatik erzeugt. Ein OpenPEARL Programm ist aus Modulen, Typ-Definitionen, Variablen und Prozedur-Deklarationen zusammengesetzt. Eine Prozedur kann aus Typ-Definitionen, Variablen-Deklarationen und einer Liste von Anweisungen bestehen. Eine Anweisung hingegen kann ein Ausdruck, eine Schleife, Verzweigung oder eine Kontrollflussanweisung wie ein Prozeduraufruf sein. Der im Nachfolgenden verwendete Begriff Objekt beschreibt Datentypen, Variablen und Prozeduren.

ZTG führt während der Generierung unter anderem Buch über Variablenzugriffe und die Aufruf-Hierarchie.

Mittels einer konfigurierbaren Wahrscheinlichkeitstabelle werden z. B. die maximale Aufruftiefe oder die zu betrachtenden Datentypen bestimmt.

In einer weiteren Berechnungsstufe wird mittels einer Filterfunktion die Gültigkeit des gerade generierten Objektes analysiert. Wird dabei ein potentiell fehlerhaftes Objekt erkannt, wird dieses verworfen und ein neues Objekt erzeugt.

Falls das Objekt als Ziel eine Variable oder eine Prozedur benötigt wird, wird dieses per Zufalls-generator erzeugt.

Ist die aktuelle Grammatikregel ein sogenanntes Nicht-Terminal, wendet RTFG rekursiv o. g. Vorgehen an und erzeugt damit ein gültiges Programmfragment.

Danach wird mittels vordefinierter Programmschablonen Quellcode für den OpenPEARL und PEARL90 Übersetzer sowie den Simulator generiert.

Mittels eines Simulator, welcher eine PEARL Laufzeitumgebung simuliert, kann z.B. in einer Skriptsprache wie Python der Testfall ablaufen.

In Abbildung 1 wird der prinzielle Ablauf darstellt.

1. Der ZTG erzeugt zufällig anhand von vorgegeben Parametern Quellcode für den OpenPEARL und für den PEARL90 Übersetzer. Alle Programme sind semantisch äquivalent. Weiterhin wird eine Eingabedatei für den Simulator erzeugt.
2. Die Übersetzer erzeugen die jeweiligen ausführbaren Programme.
3. Die ausführbaren Programme und die Simulation werden gestartet.
4. Die jeweiligen Ergebnisse werden verglichen. Falls Unterschiede festgestellt werden, muss ein Übersetzer oder der Simulator ein fehlerhaftes Ergebnis produziert haben.

Da OpenPEARL in einigen Punkten von der PEARL90 Sprachnorm abweicht bzw. diese präzisiert, ist noch zu prüfen, inwieweit der PEARL90 Übersetzer der Fa. Werum verwendbar ist.

5.3 Zusammenfassung und Ausblick

Der Testfallgenerator wird in mehreren Teilprojekten (Simulator und der eigentliche Testfall-Generator) an der FernUniversität in Hagen entwickelt.

Aktive Beteiligung

Unabhängig von diesem Projekt ist die aktive Beteiligung von weiteren Mitarbeitern, sei es bei der Sprachdefinition, der Umsetzung weiterer OpenPEARL-Sprachkonstrukte oder der allgemeinen Verbesserung, insbesondere der semantischen Analyse und der Dokumentation des Übersetzers, jederzeit willkommen.

Weitere Informationen befinden sich auf der Projektseite und in [1].

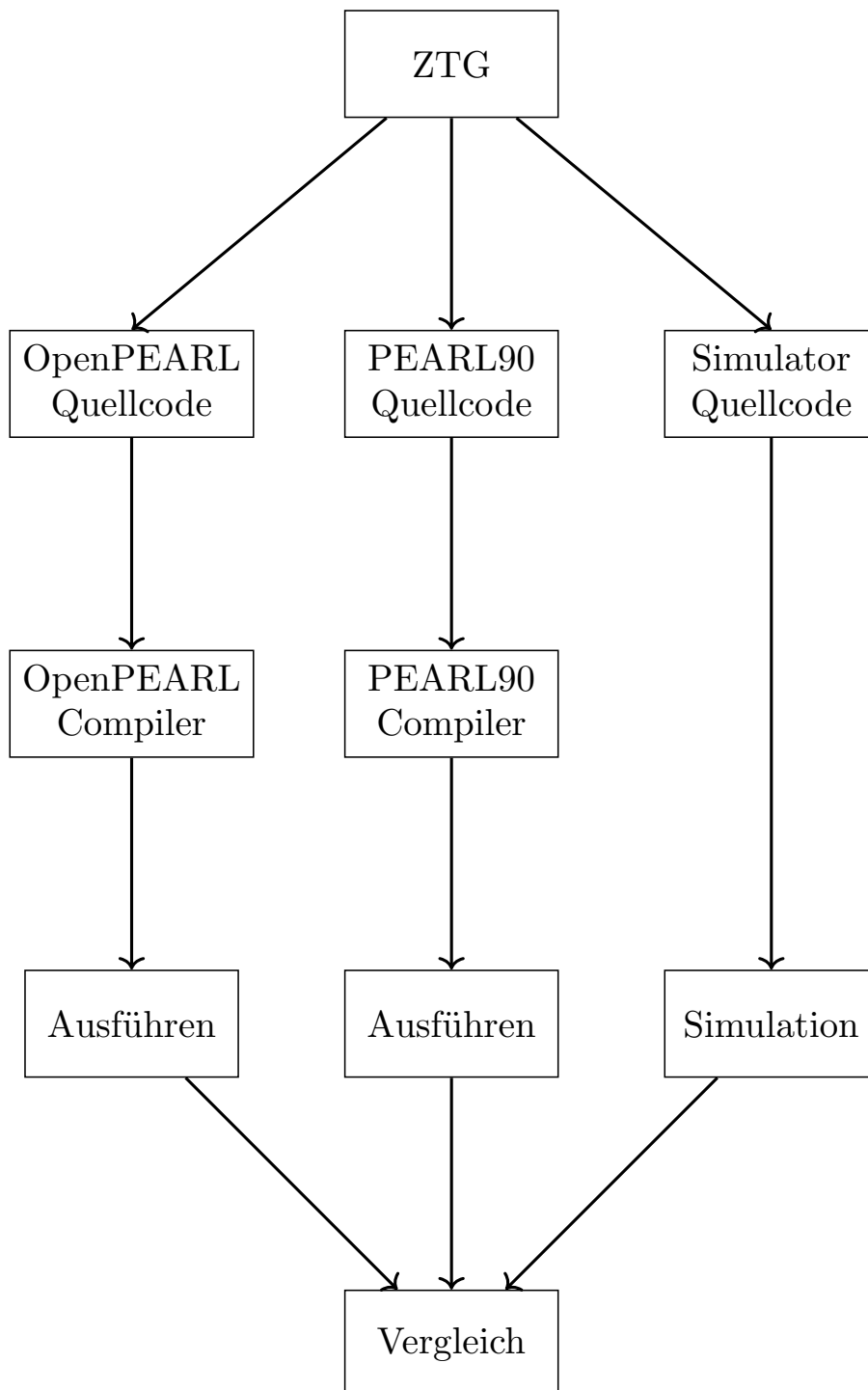


Abbildung 1: Zufallsgesteuerter Testfallgenerator

Literatur

- [1] R. Müller, M. Schaible : Die Programmierumgebung OpenPEARL90. ; Industrie 4.0 und Echtzeit - Echtzeit 2014, Fachtagung des gemeinsamen Fachausschusses Echtzeitsysteme von Gesellschaft für Informatik e.V.(GI), VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und Informationstechnischer Gesellschaft im VDE (ITG), Boppard, 20. und 21. November 2014, Seiten 31-40
- [2] GI Fachgruppe 4.4.2: PEARL90 – Sprachreport, Version 2.0, 1995.
- [3] X. Yang, Y. Chen, E. Eide, J. Regehr : Finding and Understanding Bugs in C Compilers ; ACM, SIGPLAN Not., 2011, Seiten 283-294.

6 Bekanntmachung auf WikiCFP

Jutta Düring, Fachausschuss Echtzeitsysteme

Erstmals habe ich die Tagung „Echtzeit 2018“ unter *WikiCFP – A Wiki for Calls for Papers* (<http://www.wikicfp.com/cfp/>) eingetragen. Mit dem Stichtag zur Beitragsanmeldung am 22. April 2018 gab es immerhin 1.352 Besucher; bis Mitte September 2018 sogar 2.200 Besucher. Auswirkungen auf den Bekanntheitsgrad der Tagung in Form von Anmeldungen etc. habe ich noch nicht erkennen können, aber die Hoffnung stirbt zuletzt . . .



The screenshot shows the WikiCFP website interface. At the top, there is a search bar and a navigation menu. The main content area displays a call for papers for 'Echtzeit 2018 : Echtzeit 2018 - Echtzeit und Sicherheit'. The page includes a table with details about the workshop, such as dates, location, and deadlines. Below the table, there is a 'Call For Papers' section with a list of topics and a brief description of the workshop.

WikiCFP
A Wiki for Calls For Papers

posted by user: [Echtzeit](#) | 2202 views | tracked by 1 users: [\[display\]](#)

[Add to My List](#) [I'm Organizer](#) [Modify](#)

Echtzeit 2018 : Echtzeit 2018 - Echtzeit und Sicherheit

Link: <http://www.real-time.de/workshop.html>

When	Nov 15, 2018 - Nov 16, 2018
Where	Boppard, Germany
Submission Deadline	Apr 22, 2018
Notification Due	May 31, 2018
Final Version Due	Jul 15, 2018

[Categories](#) [real-time](#) [safety](#) [security](#) [information security](#)

Call For Papers

ECHTZEIT is the annual workshop of the technical committee "Real-time Systems" of the German Informatics Society, held since 1980 (in the beginning as PEARL Workshop). The proceedings will be published by Springer Verlag in the series 'Informatik aktuell'.

CALL FOR PAPER:
Mit ihrer Fachtagung bietet der GI/GMA/ITG-Fachausschuss Echtzeitsysteme Wissenschaftlern, Nutzern und Herstellern ein Forum an, auf dem neue Trends und Entwicklungen zu Echtzeit und Sicherheit vorgestellt werden sollen, insbesondere hinsichtlich

- Echtzeitfähigkeit
- Funktionaler Sicherheit
- Datensicherheit und Datenschutz
- Sichere Datenkommunikation
- Industrie 4.0 und Internet der Dinge
- Zuverlässigkeit
- Ambient Assisted Living
- Autonomes Fahren
- Smart Grids
- Aktuelle Anwendungen
- Ausbildung

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

Abbildung 2: 2.200 Besucher bis Mitte September 2018 (Quelle: <http://www.wikicfp.com/cfp/>)