Echtzeit

Nr. 5, Januar 2017

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. Dr. Wolfgang A. Halang FernUniversität in Hagen Lehrstuhl für Informationstechnik 58084 Hagen wolfgang.halang@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 pearl@fernuni-hagen.de
ISSN	2199-9244

Redaktionell abgeschlossen am 5. Januar 2017

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 2017: Call for Paper

Jutta Düring, Fachausschuss Echtzeitsysteme

Die Tagung „Echtzeit 2017“ findet am **16. und 17. November 2017** wie gewohnt in Boppard am Rhein statt. Unser diesjähriges Leitthema lautet:

Logistik und Echtzeit

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

- Echtzeitfähigkeit
- Funktionale Sicherheit
- Datensicherheit und Datenschutz
- Sichere Datenkommunikation
- Industrie 4.0 und Internet der Dinge
- Autonome Fahrzeuge und Transportsysteme
- Ressourcen- und Lieferkettenverwaltung
- Zuverlässigkeit
- Aktuelle Anwendungen
- Ausbildung

Stichtag für die Vortragsanmeldung ist **Montag, der 24. April 2017**. Nähere Informationen finden Sie unter `http://www.real-time.de/CfP.html`.

2 Änderungen in der Sprachdefinition in OpenPEARL in Bezug auf PEARL90 – Teil 3

Rainer Müller, Hochschule Furtwangen, mueller@hs-furtwangen.de
Marcel Schaible, FernUniversität in Hagen, marcel.schaible@fernuni-hagen.de

Projekt-Homepage: `http://sourceforge.net/projects/openpearl/`

Wie schon in [1] und [2] dargestellt ist die Sprachdefinition von PEARL90 [3] weder widerspruchsfrei noch eindeutig formuliert. In Ergänzung zu den dort bereits vorgestellten Konstrukten wurden auf der Sitzung des Arbeitskreises OpenPEARL am 17. November 2016 in Boppard am Rande der Tagung „Echtzeit“ folgende weitere Änderungen festgehalten.

2.1 GLOBAL-Schlüsselwort

In PEARL90 hat die Angabe eines Modulnamens beim GLOBAL-Schlüsselwort nur dokumentarischen Charakter. Zur Dokumentation gibt es in allen Programmiersprachen die Kommentaranweisungen.

Für OpenPEARL wurde entschieden, dass die Angabe eines Modulnamens beim GLOBAL-Schlüsselwort nun für den Import aus anderen Modulen verpflichtend wird. Dieser Modulname trennt dann auch Namensräume, wie dies z.B. in C++ bekannt ist.

Der Modulname kann entfallen, wenn ein Element aus dem Systemteil des gleichen Moduls importiert wird. Dies bedeutet, dass der eigene Modulname der Defaultwert für die Modulbezeichnung bei GLOBAL ist.

2.2 Optionale Formatangabe bei PUT und GET

Die Sprachdefinition wird dahingehend ergänzt, dass bei einer fehlenden Formatangabe bei PUT oder GET automatisch das LIST-Format zur Anwendung kommt und ein SKIP angefügt wird.

Beispiel:

```
DCL x FIXED(15);
DCL y FIXED(31);
PUT 'x=',x,' y=',y TO terminal;
```

ist gleichbedeutend mit

```
DCL x FIXED(15);
DCL y FIXED(31);
PUT 'x=',x,' y=',y TO terminal BY A, F(4) , A, F(11), SKIP;
```

2.3 Eingabe mit F-Format

In PEAL90 ist als einzige Ausnahme beim F-Format geregelt, dass ein leeres Eingabefeld als Wert 0 interpretiert wird. Dieser Punkt wird zu Gunsten der Homogenisierung aufgegeben.

Ein leeres Eingabefeld beim F-Format liefert in Zukunft eine Fehlermeldung per SIGNAL (`NoDataInFieldSignal`).

2.4 TFU

In PEARL90 hat TFU bei der DATION-Vereinbarung keine Funktion. Die Angabe wird dort vom Compiler ignoriert und wird nur zur Wahrung der Kompatibilität zu älteren Programmquellen toleriert.

In OpenPEARL soll nun folgendes bzgl. TFU gelten:

1. Neben der Angabe von MAX ist auch die Angabe einer positiven ganzen Zahl zulässig.
2. Bei fehlender numerischen Angabe wird die Defaultgröße 128 Byte eingesetzt.
3. Bei Eingaben sind Zeileneditierfunktionen (backspace, delete) nur innerhalb dieser Transferunit möglich. Sobald diese Zeichenzahl eingeben wurde, wird der Puffer von der Eingabeformatierung interpretiert.

4. Für die Ausgabe wird ein Ausgabepuffer dieser Größe von der Formatierung genutzt und austransferiert, wenn der Puffer voll ist oder eine Positionieranweisung (SKIP, X, POS, ...) erfolgt.
5. TFU hat nur bei der formatierten Eingabe mit PUT und GET eine Bedeutung. Bei READ/WRITE wird nicht separat gepuffert.

2.5 Benutzerdefinierte Signale

Es werden in OpenPEARL benutzerdefinierte Signale eingefügt.

```
UserSignalDeclaration ::=
   DCL OneIdentifierOrList SIGNAL
   RST( ConstantFIXEDExpression$ErrorNumber
     [ , ConstantFIXEDExpression$ErrorNumber ] ... ) GLOBAL;

UserSignalSpecification ::=\\
  SPC OneIdentifierOrList SIGNAL GLOBAL ( IdentifierOfTheModule );
```

Alle benutzerdefinierten Signale müssen per RST eine Fehlernummer zugeordnet bekommen.

Im Buildprozess wird überprüft, ob die Signalnummer nicht schon anderweitig in Benutzung ist.

2.6 Referenzen auf SIGNALS, INTERRUPTS und DATIONS

Diese Elemente bleiben erhalten. Der Sprachreport wird um entsprechende Beispiele ergänzt. Der Zugriff über nicht definierte Referenzen muss sicher erkannt und mit einem Signal gemeldet werden.

2.7 Zeitumstellung

Die PEARL-typischen Einplanungen mit absoluten Zeiten erfordern eine anwendungsspezifische Behandlung der Zeitumstellung bei Sommer-/Winterzeitumstellung oder auch Zeitkorrekturen per ntp.

Zur Behandlung dieser Fälle wird ein Systemelement eingeführt, welches anhand geeigneter Parameter einen *timeChanged* Interrupt auslöst.

Beispiel:

```
MODULE(demo);
SYSTEM;
   ! monitor the system time all 60 seconds and
   ! trigger the timeChange interrupt, if the
   ! deviation is larger than 10 milli seconds.
   timeChanged: MonitorTimeChange(60 SEC, 0.01 SEC);
PROBLEM;
   SPC timeChanged INTERRUPT GLOBAL(demo);
```

2.8 Weitere Änderungen

1. `RESIDENT` wurde eliminiert, da das Speichermanagement bei aktuellen Betriebssystemen von diesen übernommen wird und nur bedingt beeinflussbar ist.

2. `REENTRANT` wurde eliminiert, da dies in den aktuellen stackorientierten Programmiermodellen standardmäßig verfügbar ist.

3. Rekursionen sollten nach wie vor vermieden werden, da diese über einen Stacküberlauf leicht zu unsicheren Programmen führen können.

4. `OPERATOR`, `BY TYPE`, `VOID` und `REF STRUCT[]` gestrichen, da dies entweder zu reduzierter Transparenz des Programms oder zu unsicheren Programmen führt.

5. Nested Functions wurden eliminiert, da bei guter Modularisierung und vernünftiger Namenswahl der gleiche Effekt mit Prozeduren auf Modulebene erreicht werden kann.

6. Formatelement `EOF` (End-Of-File) als relative Positionsangabe wurde neu eingeführt. Es positioniert die Datei auf deren Ende, sodass sie einfach verlängert werden kann.

7. `EXCLUSIVE` als Openparameter eingeführt, damit eine Dation zeitgleich nur einmalig geöffnet werden kann – die Sprachnorm lässt zu, dass eine Dation mehrfach geöffnet und geschlossen wird.

8. `CONTROL()` und `S(...)`-Format bei TAKE/SEND gestrichen, da diese Anweisungen nach Sprachreport implementationsabhängig sind. Eine konkrete Funktion ist in PEARL90 hierzu nicht definiert. Das Verhalten eines PEARL-Programms sollte nicht von der Implementierung des Sprachstandards abhängen.

9. Signalbehandlung wurde ergänzt um `INDUCE` und `SignalFinalStatement`. Ein implizites `RETURN/TERMINATE` ist gestrichen, damit das Programmverhalten in diesen Situationen deutlich ersichtlich ist und auch Funktionsprozeduren einen korrekten Rückgabewert liefern.

10. Syntax im Systemteil (Parameterangabe, Konfigurationselemente und Assoziationen) wurde überarbeitet und wieder an frühere Schreibweisen angeglichen. Felder sind im Systemteil nicht zulässig. Die Parameter von Systemteilelementen werden funktionsartig geschrieben.

11. Schlüsselwort `SYSTEM` in DATION-Spezifikation zur Kennzeichnung einer Systemdatenstation.

12. Die Umwandlung zwischen FIXED und BIT wurde geändert:

    - `TOBIT(FIXED(n))` liefert jetzt `BIT(n+1)` bzw.
    - `TOFIXED(BIT(n))` liefert jetzt `FIXED(n-1)`.

    Als Konsequenz ist nun auch `FIXED(0)` in OpenPEARL zulässig, damit die Umwandlung zwischen BIT und FIXED inklusive Vorzeichenbit funktioniert.

Anmerkung:
In PEARL90 gilt: `TOBIT(FIXED(n))` liefert `BIT(n)` bzw. `TOFIXED(BIT(n))` liefert `FIXED(n)`. Dies bedeutet, dass ein Bit der FIXED-Zahl bei der Umwandlung nach BIT verloren geht – vermutlich das Vorzeichenbit. Es ist nicht geklärt, wie dies bei negativen Zahlen gehandhabt werden soll. Bei der Umwandlung einer BIT(n) nach FIXED wird auf jeden Fall eine positive FIXED-Zahl geliefert. – Eine bitweise Kopie zwischen BIT und FIXED ist in PEARL90 nicht möglich, wenn ein Bereichsüberlauf zu Fehlermeldungen führt:

```
DCL A FIXED (15);
DCL D BIT (16) ;
d:= 'FF'B4;
A := (TOFIXED D) FIT A ;
```

Der Wert von `TOFIXED D` ist 65535 mit dem Datentyp FIXED(16). Bei der Reduktion auf FIXED(15) entsteht ein Bereichsüberlauf, da FIXED(15) einen Wertebereich von −32768 bis 32767 hat.

Literatur

[1] R. Müller, M. Schaible: Änderungen in der Sprachdefinition in OpenPEARL in Bezug auf PEARL90. In GI/GMA/ITG-Fachausschuss Echtzeitsysteme (Hg.): Echtzeit. 2015, 3, S. 8–13.

[2] R. Müller, M. Schaible: Änderungen in der Sprachdefinition in OpenPEARL in Bezug auf PEARL90 – Teil 2. In GI/GMA/ITG-Fachausschuss Echtzeitsysteme (Hg.): Echtzeit. 2016, 4, S. 8–12.

[3] GI Fachgruppe 4.4.2: PEARL90 – Sprachreport, Version 2.0, 1995.

3 Vorwort der Tagung „Echtzeit 2016“ – Internet der Dinge

Wie bereits vor zwei Jahren mit dem Leitthema „Industrie 4.0“ nimmt diese Tagungsreihe erneut einen Modebegriff auf, um ihn vor dem Hintergrund jahrzehntelanger Erfahrung auf den Gebieten Echtzeit- und eingebettete Systeme sowie Prozessautomatisierung und -leittechnik auf seine Substanz und seinen Neuheitsgehalt hin abzuklopfen. Eine Definition[1] besagt, das „Internet der Dinge bezeichne die Vernetzung von Gegenständen mit dem Internet, damit diese Gegenstände selbstständig über das Internet kommunizieren und so verschiedene Aufgaben für den Besitzer erledigen könnten. Der Anwendungsbereich erstrecke sich dabei von einer allgemeinen Informationsversorgung über automatische Bestellungen bis hin zu Warn- und Notfallfunktionen.“

[1] Springer Verlag (Hg.), Gabler Wirtschaftslexikon, Stichwort: Internet der Dinge, http://wirtschaftslexikon.gabler.de/Archiv/1057741/internet-der-dinge-v4.html

Häufig wird über lange Zeit Hergebrachtes in neue Moden übernommen[2]. Als Beispiele dafür seien bzgl. des Internets der Dinge hier nur die im letzten halben Jahrhundert evolutionär entwickelten Technologien eingebetteter Systeme und zustandsabhängiger vorbeugender Anlageninstandhaltung genannt. Allein in der Verwendung von mehr Prozessoren und Sensoren, weiterer Miniaturisierung sowie erheblich gesteigertem Datenaustausch erkennt der Automatisierungstechniker jedoch genauso wie im Falle von Industrie 4.0 nichts prinzipiell Neues.

In Bezug auf den Einsatz der Radiofrequenzidentifikation 1999 von Ashton als Begriff geprägt, befindet sich das Thema Internet der Dinge nach dem 2015 veröffentlichten „New Gartner Hype Cycle for Emergent Technologies“[3] derzeit auf dem Gipfel der Euphorie, worauf in der Regel der Fall in das Tal der Enttäuschungen folgt. Dies erscheint unausweichlich, denn wie so Vieles ist das Internet der Dinge ein zweischneidiges Schwert: Einerseits soll es Effizienz und Wettbewerbsfähigkeit erhöhen, öffnet aber andererseits Werksspionage und Sabotage Tür und Tor. Gemäß der Netzseite „Sicherheit für das Internet der Dinge“[4] stellt Sicherheit die größte Herausforderung des Internets der Dinge dar.

Dabei ergibt sich mangelnde funktionale Sicherheit in vernetzten Systemen als Folge der Gefahren durch Abhören, Verfälschen oder Abfangen von Daten bei deren Übertragung sowie der Möglichkeiten des Einschleusens von Malware und der Überlastung von Knoten, die der Einsatz ungeeigneter Artefakte, nämlich im Bürobereich verbreiteter Informationstechnik und des Internets als Übertragungsmedium, mit sich bringt. Weil dieses Gefahrenpotenzial bei klassischen Technologien der Automatisierungstechnik viel geringer – oder konstruktionsbedingt überhaupt nicht vorhanden – ist, haben wir es hier bei Licht betrachtet mit einem Rückschritt hinsichtlich der technischen Qualität zu tun.

Aufgrund der oben umrissenen Problemlage ist die erste Sitzung der Tagung der Sicherheit im Internet der Dinge gewidmet. Sie will für das Thema Datensicherheit sensibilisieren, indem Möglichkeiten zur Absicherung von Kommunikationsnetzen diskutiert und die zur Abschirmung von Code und Daten für verschiedene Isolationskonzepte entstehenden Kosten betrachtet werden. Als konkrete Problemlösung wird ein zum Patent angemeldetes Verfahren zur sicheren anonymen Aufwertung und Belastung elektronischer Geldbörsen vorgestellt.

Das Internet der Dinge ist ohne den Einsatz von Echtzeitbetriebssystemen an den verschiedensten Stellen undenkbar. Darum beschäftigt sich eine eigene Sitzung mit den bei der Entwicklung eines dezidierten Betriebssystemkerns zusammengestellten spezifischen kombinierten Echtzeit- und Sicherheitsanforderungen, einer modular geschichteten Systemarchitektur zur Verbindung heterogener Gerätetreiber mit gängigen Übertragungsprotokollen sowie mit einem Betriebssystemkern für mit Umgebungsenergie betriebene Komponenten, das bei seinen Einplanungsentscheidungen deren schwankende Verfügbarkeit berücksichtigt.

Wegen der entscheidenden Bedeutung von Planung wird in der dritten Sitzung eine Software-Bibliothek für Ausbildungszwecke präsentiert, die die Planung des Echtzeitverhaltens unterstützt. Der Ausbildung dient auch die in Entwicklung befindliche Programmierumgebung OpenPEARL, über deren Konsistenzprüfungskomponente berichtet wird. Für die sicherheitsgerichtete Echtzeitprogrammierung wurden auf der Basis von PEARL eine Grundsprache sowie für jede der

[2] P. Mertens und D. Barbian: Digitalisierung und Industrie 4.0 – Trend mit modischer Überhöhung? *Informatik Spektrum* 39, 4, 301–309, 2016

[3] http://www.gartner.com/newsroom/id/3114217

[4] http://www.iot-sicherheit.ch

vier Sicherheitsstufen nach IEC 61508 eine inhärent sichere Teilsprache definiert, deren Syntax die Einhaltung der jeweiligen Einschränkungen erzwingt.

Im Rahmen der Tagung beschäftigt sich schon immer eine Sitzung mit aktuellen Echtzeitanwendungen. Gemäß dem Fokus Internet der Dinge werden Kontrollverfahren für die dynamische Kommunikation zwischen mobilen Hausautomatisierungsgeräten vorgestellt, eine Steuereinheit für Bewegungen mit sechs Freiheitsgraden entwickelt und in ein System virtueller Realität eingebunden sowie Methoden der erweiterten Realität zur Navigation in Gebäuden genutzt.

In der abschließenden Sitzung wird zunächst am Beispiel der Fernsteuerung von Heizungen gezeigt, dass konsequente Trennung von Anwendungsbereichen mit und ohne Echtzeitanforderungen und Verwendung jeweils geeigneter Programmiersprachen zu besseren Lösungen führt. Weiterhin werden ein Testbett für in der Raumfahrt eingesetzte und auf NAND-Flash-Speichern residierende Dateisysteme mit Redundanz sowie ein hierarchisch-asynchrones Zuteilungsverfahren für Mehrkernprozessoren vorgestellt, das die Software eingebetteter Systeme zur Laufzeit phasenabhängig rekonfiguriert.

Frau Dipl.-Ing. Jutta Düring gebührt unser herzlicher Dank dafür, dass sie zum wiederholten Male die Einreichungen mit größter Sorgfalt redigiert und den vorliegenden Band konsistent und ansprechend gestaltet hat.

Hagen, im August 2016

Wolfgang A. Halang
Herwig Unger

4 Graduiertenwettbewerb 2016 und Best Paper Award 2016

Jutta Düring, Fachausschuss Echtzeitsysteme

Die Preisträger des Graduiertenwettbewerbs 2016 sind

- **Pascal Pieper**: Umgebung für automatisierte Tests von Dateisystemen auf NAND-Flash-Speichern (http://dx.doi.org/10.1007/978-3-662-53443-4_14)
- **Phillip Raffeck**: Entwurf und Implementierung eines energieneutralen Echtzeit-Betriebssystems (http://dx.doi.org/10.1007/978-3-662-53443-4_6)
- **Sebastian Thomeczek**: Echtzeitanforderungen an Virtual Reality Systeme – Interaktive Anwendungen mit sechs Freiheitsgraden (http://dx.doi.org/10.1007/978-3-662-53443-4_11)

Das Programmkomitee der Tagung „Echtzeit 2016“ hat den Beitrag

Real-time and Security Requirements for Internet-of-Things Operating Systems

von Maja Malenko und Marcel Baunach mit dem Best Paper Award ausgezeichnet. (http://dx.doi.org/10.1007/978-3-662-53443-4_4)

5 Conference Autonomous Systems 2017

Jutta Düring, Fachausschuss Echtzeitsysteme

Der GI-Arbeitskreis *Echtzeitkommunikation* veranstaltet im nächsten Jahr bereits die „10th GI Conference on Autonomous Systems“, welche vom **22. bis 27. Oktober 2017** in Cala Millor auf Mallorca stattfinden wird.
Homepage: http://www.fernuni-hagen.de/kn/autsys/

Der Konferenzband 2016 erschien in den „Fortschritt-Berichten“ des VDI-Verlages [Reihe 10 Informatik/Kommunikation, Nr. 848] und beinhaltet folgende Beiträge.

Safety-related and Real-time Systems

- D. Koß: A Comparative Survey of System Specification Techniques for Safety-related Environments
- S. Widmann: Requirements for Safe Computer Architectures
- F. Saglietti: Systematic and Probabilistic Testing of Autonomous Mobile Robots
- M. Schaible: On the Contruction of a Crowd-verifiable Microprocessor
- M. Kirchhoff et al.: Affordable High-bandwith Real-time Mass-storage Architecture for Distributed Sensor Nodes

Networks and Routing

- S. Sodsee et al.: Applying Centrality Measures in a Multi-criteria Routing Strategy for Wireless Sensor Networks
- T.T. Naing et al.: Routing-based Topological Analysis on the Road Network in Myanmar
- F. Nassermostofi: Toward Authentication between familiar Peers in P2P Networking Systems

Suppression of Disturbances

- J. Niu et al.: Feasibility Study of Applying Chaotic Carrier Frequency Modulation in Switching-mode Power Supply
- N. Alain et al.: Identification of Filter Types by Bifurcation Analysis: Mathematical Modelling and Numerical Simulation

Neural and Evolutionary Computing

- M. Sodanil et al.: Retinopathy of Prematurity Classification based on Image Analysis
- A. Haj Mosa et al.: Dual Echo State Networks-based Generalized Matrix Inversion with Applications in Stochastic Time-varying Systems
- D. Suthiwong et al.: Cardinality-constrained Portfolio Optimization using an improved quick Artificial Bee Colony Algorithm

Natural Language Processing

- M.M. Kubek et al.: Centroid Terms and their Use in Natural Language Processing
- C. Tapsai et al.: Thai Language Segmentation by Automatic Ranking Trie

Education

- T. Tempelmeier: Didactics in Computer Science Education at Universities of Applied Sciences
- H. Coltzau et al.: Towards Live Feedback in Online Exercise Systems

Das Programm beinhaltet auch Kurzvorträge ohne Veröffentlichung im Konferenzband. Außerhalb der Veranstaltungen bleibt ausreichend Zeit für Diskussionen bei einer Tasse Kaffee oder einem Spaziergang am Strand.

Die Teilnahme an der Konferenz basiert ausschließlich auf Einladung. Bei Interesse können Sie sich gerne an Professor Unger unter herwig.unger@fernuni-hagen.de wenden.

6 Arbeitskreis OpenPEARL

Rainer Müller, Hochschule Furtwangen, mueller@hs-furtwangen.de
Marcel Schaible, FernUniversität in Hagen, marcel.schaible@fernuni-hagen.de

Projekt-Homepage: http://sourceforge.net/projects/openpearl/

6.1 Status der Implementierung von OpenPEARL

Die Implementierung von OpenPEARL geht langsam aber stetig voran. In diesem Artikel wird der aktuelle Stand dargestellt. Die Darstellung erfolgt in einer Positivliste. Sofern eine Sprachkomponente nur teilweise implementiert ist, ist angegeben, ob diese im Compiler (C) oder Laufzeitsystem (L) funktioniert. Die Entwicklung für Mikrocontroller ist stets hinter der Entwicklung für Linux, sodass hier nur die Linuxportierung betrachtet wird.

Die Sprachänderungen, welche auf der Sitzung des Arbeitskreises am 17.11.2016 besprochen wurden (siehe Beitrag 2), werden hier nicht berücksichtigt, da sie entweder noch offene Punkte betreffen oder nur kleine Anpassungen nach sich ziehen.

Sprachelement	**Unterstützung**	**Bemerkung**
ein Modul	CL	für mehrere Module fehlt die Unterstützung von GLOBAL
Systemteil	CL	
Problemteil	CL	Struktur funktioniert — es fehlen noch einige Sprachelemente
Datentyp FIXED		
DCL	CL	INIT nur mit Literalen
Ausdrücke	CL	DIV Operator fehlt noch
E/A	CL	

Sprachelement	Unterstützung	Bemerkung
Datentyp FLOAT		
DCL	CL	INIT nur mit Literalen
Ausdrücke	CL	
E/A	CL	
Datentyp BIT		
DCL	CL	INIT nur mit Literalen
Ausdrücke	L	
E/A	CL	
Datentyp CHAR		
DCL	CL	INIT nur mit Literalen
Ausdrücke	L	
E/A	CL	
Datentyp CLOCK		
DCL	CL	INIT nur mit Literalen
Ausdrücke	L	
E/A	CL	
Datentyp DURATION		
DCL	CL	INIT nur mit Literalen
Ausdrücke	L	
E/A	CL	
Datentyp SEMA		
DCL	CL	INIT nur mit Literalen
Ausdrücke	CL	
E/A	CL	
INV	C	für bestehende Datentypen; betrifft nur den Compiler
IF	C	betrifft nur den Compiler
CASE	C	beide Varianten; betrifft nur den Compiler
REPEAT	C	betrifft nur den Compiler
GOTO	C	betrifft nur den Compiler
EXIT	C	betrifft nur den Compiler
PROC	C	Deklaration und CALL gehen; betrifft nur den Compiler
TASK	CL	Definition und Tasksteuerung funktionieren. Ausdrücke sind bei Tasksteuerung noch nicht unterstützt
SEMA	CL	
INTERRUPT	L	Compiler erkennt die Elemente, die Codeerzeugung fehlt noch
DATION	CL	

Zur Vollständigkeit sei erwähnt, dass die Bereiche STRUCT, ARRAY, TYPE, LENGTH, BOLT und SIGNAL noch nicht in Bearbeitung sind. Es gibt Vorüberlegungen, die allerdings noch nicht umgesetzt wurden.

6.2 Offene Arbeitspakete im OpenPEARL-Projekt

6.2.1 Einleitung

Es gibt eine Reihe von Arbeitspaketen, die im Rahmen von Semester- oder Thesisarbeiten gut bearbeitet werden können. Die Umsetzung sollte im Laufe des Jahres 2017 erfolgen. Für die fachliche Unterstützung der Studierenden stehen die Autoren bereit. Die Bearbeitung der Arbeitspakete sollte so erfolgen, dass diese später bei Bedarf angepasst und erweitert werden können. Alle Arbeitspakete können unter Linux bearbeitet werden.

6.2.2 Testanwendungen für OpenPEARL

Die korrekte Umsetzung der Sprachdefinition muss durch Testanwendungen belegt werden. Hierzu können einzelne Testprogramme erstellt werden oder das neue Testframework von OpenPEARL benutzt werden. Dieser Bereich lässt sich z.B. unterteilen in

- Unterstützung des einzelnen Datentypen (FIXED, FLOAT, BIT, CHAR, CLOCK, DURATION)
- Ein-/Ausgabe mit PUT/GET und WRITE/READ
- Tasking Funktionalitäten
- Fehlererkennung im Compiler
- ...

Anwendungen in OpenPEARL sind natürlich auch eine gute Möglichkeit, das Sprachsystem zu überprüfen. Diese sehen die Autoren frühestens für das Sommersemester 2017 als realistisch an, da bis dahin das Sprachsystem weitgehend fertig gestellt sein sollte.

6.2.3 Performance-Tests

Die Entwicklung im Sprachsystem war bislang auf korrekte Funktionserfüllung ausgerichtet. Die Performanz zur Laufzeit stand noch nicht im Fokus. Ziel in diesem Thema sollten Performanzmessungen für verschiedene Typen von Operationen sein. Dieses Thema lässt sich unterteilen in z.B.

- Berechnungen in den einzelnen Datentypen
- Ein-/Ausgabe
- Fehlerbehandlung

Die Messungen könnten z.B. mit nativem C-Code und JAVA verglichen werden.

6.2.4 Benutzerhandbuch

Ein Benutzerhandbuch zur Erstellung von Programmen mit OpenPEARL wäre hilfreich. Hier sollten die Ein-/Ausgabe sowie üblichen Muster wie Ringpuffer behandelt werden. Die Synchronisationsdetails von SEMAPHORE und BOLT sind sicherlich auch von Interesse. Dieses Handbuch sollte mit LaTeX erstellt werden.

6.2.5 IDE als Eclipse Plug-In

Die Entwicklungsumgebung Eclipse ist recht beliebt. Ein Plug-In für die Entwicklung von Programmen mit OpenPEARL würde die Akzeptanz bei Anfängern sicherlich steigern. Als minimaler Funktionsumfang sehen wir Syntaxhighlighting, Compileraufruf und Auswertung der Fehlermeldungen. Sinnvolle Erweiterungen wären Onlinehilfe und Debuggingunterstützung.

6.2.6 Autoinstall

Aus lizenzrechtlichen Gründen wird OpenPEARL in Quellform verteilt und vom Systembetreiber compiliert und installiert. Hierzu werden eine Reihe von Werkzeugen benötigt, die alle entweder aus den entsprechenden Projektrepositories oder von den jeweiligen Paketquellen installiert werden. Eine Automatisierung dieses Vorgangs wäre hilfreich, sodass für die üblichen Linux-Distributionen (Suse, Debian, Ubuntu, ...) die benötigten Pakete bei Bedarf automatisiert installiert werden. Der Support für die Crossentwicklung für Mikrocontroller wäre schön, ist aber für die erste Ausbaustufe nicht essentiell.

6.3 Aktive Projektbeteiligung

Die aktive Beteiligung von weiteren Mitstreitern, sei es bei der Erstellung von Testanwendungen, der Umsetzung weiterer OpenPEARL-Sprachkonstrukte oder der allgemeinen Verbesserung des Übersetzers, sind dringend notwendig und jederzeit willkommen.

7 Zusammenfassungen von Abschlussarbeiten

Jutta Düring, Fachausschuss Echtzeitsysteme

7.1 Kontrollverfahren für mobile Echtzeitkommunikation

Sven Biermann, M.Sc., FernUniversität in Hagen, Lehrgebiet Kommunikationsnetze
Betreuer: mario.kubek@fernuni-hagen.de

Heutige Lösungen für die Hausautomatisierung bestehen oftmals aus geschlossenen Systemen eines Anbieters, um verschiedene Geräte wie z. B. elektrische Rollläden, Thermostate an Heizkörpern oder Multimediaanlagen zu steuern. Dabei wird auf ein Steuerungselement gesetzt, mit

dem die meist statischen Geräte bedient werden. Ziel dieser Masterthesis war es daher, eine flexiblere Lösung für die Hausautomatisierung, basierend auf dem Betriebssystem Android, zu entwickeln, um feststehende automatisierte Komponenten mit mobilen Geräten wie Tablets und Smartphones zu verbinden. Zentrale Anforderungen für die Umsetzung waren die Bereitstellung einer dynamischen Umschaltung zwischen verschiedenen Funkmedien, die als Grundlage für die Gerätevernetzung eingesetzt werden sollte, die Unterstützung eines Routingverfahrens, um mehrere Geräte in die Vernetzung mit einzubeziehen und die Schaffung einer sicheren Kommunikationsumgebung durch die Kanalumschaltung.

Das auf diesen Anforderungen erstellte Konzept verwendet die Protokolle Bluetooth und WifiDirect für eine Ad-Hoc Vernetzung der Androidgeräte auf eine für Heimautomatisierungsgeräte begrenzte Reichweite von bis zu 10 m zwischen den Verbindungspartnern. Dabei ermöglicht es den Geräten, sich auf beiden oder nur einem Kanal zu vernetzen. Bestehende Lösungen wie SPAN oder the Serval Network unterscheiden sich in der Anwendung dahingehend, dass sie die Vernetzung von Geräten mit WifiDirect über große Entfernungen hinweg ermöglichen und nur für ausgewählte Geräte zur Verfügung stehen. Eine dynamische Kanalumschaltung als Sicherheitsaspekt sehen diese Lösungen nicht vor. Hinsichtlich des Routings verbindet das Konzept von SmartConnect den reaktiven Ansatz des Ad-Hoc Protokolls AODV, das in einigen verfügbaren Lösungen eingesetzt wird, mit dem kontextsensitiven Ansatz von SCAR, das neben klassischen Identifikationsmerkmalen wie MAC und IP-Adressen auf umgebungsspezifische Informationen setzt. So ermöglicht dieses Konzept über eine WifiDirect Service Discovery Dienste, die von den Geräten angeboten werden, zu erkennen und selbst weiter zu verbreiten. Dabei werden die Gerätenamen der anbietenden Geräte verkettet und so als kontextsensitive Information für das Routing eingesetzt. Ein Dienstename beschreibt neben der Kennzeichnung einer Funktion gleichzeitig die Route zu seinem Anbieter. Alle Gerätenamen innerhalb dieser Verkettung stellen Gatewayknoten dar, die über den jeweiligen Gerätenamen die WifiDirect und Bluetooth Konfigurationen ihrer Kommunikationspartner ermitteln können. Diese Konfigurationen werden zuvor bei der WifiDirect Service Discovery und der Bluetooth Device Discovery ermittelt. Dadurch sind die Geräte in der Lage, Vorgänger- und Nachfolgeknoten auf der Route zu identifizieren und mit diesen Informationen ihre Routingtabellen zu befüllen. Gleichsam hat das Anbieten von Diensten den Vorteil, dass darüber verschiedene Funktionen der Anwendung, definiert durch Dienstetypen, angeboten werden können, darunter z. B. verschiedene Funktionen wie Rollladensteuerung und Thermostatsteuerung. Das Routing selbst erfolgt mit Hilfe eines klassischen Protokollheaders, welcher die Namen des Quell-, Nachfolge- und Zielgerätes sowie verschiedene Steuerdaten analog zu etablierten Routingverfahren enthält. Darüber hinaus überträgt es als Nutzdaten ein selbstdefiniertes, an das Diagnoseprotokoll UDS angelehntes Diensteprotokoll, das sogenannte SmartConnectProtokoll.

Diese Abschlussarbeit wurde auch auf der Tagung „Echtzeit 2016“ vorgestellt. Den Beitrag zum Tagungsband finden Sie unter http://dx.doi.org/10.1007/978-3-662-53443-4_10.

7.2 Indoor-Navigation mit Augmented Reality-Unterstützung

Andreas Hümmerich, M.Sc., FernUniversität in Hagen, Lehrgebiet Kommunikationsnetze
Betreuer: mario.kubek@fernuni-hagen.de

Befindet sich eine Person außerhalb eines Gebäudes, stehen ihr heutzutage solide und technisch ausgereifte Mittel zur Verfügung, um den Weg von ihrem aktuellen Standpunkt zu ei-

nem gewünschten Ziel ermitteln und eine entsprechende, interaktive und sich in Echtzeit aktualisierende Wegbeschreibung erhalten zu können. Anders stellt sich die Situation dar, wenn die Navigation innerhalb eines Gebäudes stattfinden soll. Trotz entsprechender Bemühungen namhafter Hersteller existiert hierzu noch kein übergreifender technischer oder konzeptioneller Standard. Ziel der zugrundeliegenden Masterarbeit war daher, zunächst die denkbaren Technologien zu vergleichen und darauf basierend eine theoretische Betrachtung der notwendigen technischen wie inhaltlichen Voraussetzungen für eine solche Indoor-Navigation zu erarbeiten. Abschließend erfolgte eine praktische Umsetzung der gewonnenen Erkenntnisse auf Basis der Android-Plattform. Insbesondere wurden dabei die Echtzeitfähigkeit der Anwendung und eine eigene Augmented Reality-Darstellung adressiert.

Eine unterliegende Basistechnologie für die Indoor-Navigation muss von gängigen Mobilgeräten (Smartphones, Tablets) unterstützt werden und sollte ferner keine hohen Infrastrukturkosten mit sich bringen. Die Nutzung globaler Navigationssatellitensysteme (GNSS) wie GPS wäre an dieser Stelle sicherlich die optimale Lösung, wäre deren Empfang innerhalb von Gebäuden in den meisten Fällen nicht unmöglich und nur mit großen technischem und finanziellen Aufwand nachrüstbar.

Es verbleiben somit WLAN und Bluetooth, die beide die Möglichkeit der etwaige Entfernungsbestimmung zu einem entsprechenden Sender mittels Auswertung des RSSI (Received Signal Strength Indicator)-Wertes erlauben als praktikable Basistechnologien. Bluetooth bietet dabei den Vorteil der kleineren und preiswerteren Sender („Baken“), die entsprechend häufig im Gebäude positioniert werden könnten. Daher erfolgt die weitere Betrachtung insbesondere auf diese Technologie hin ausgerichtet, eine analoge Umsetzung mit Hilfe von WLAN oder einer Kombination der Technologien wäre jedoch ebenfalls möglich.

Diese Abschlussarbeit wurde auch auf der Tagung „Echtzeit 2016“ vorgestellt. Den Beitrag zum Tagungsband finden Sie unter http://dx.doi.org/10.1007/978-3-662-53443-4_12.

7.3 Entwicklung einer Modultestumgebung für OpenPEARL

Martin Beßling, FernUniversität in Hagen, martin.bessling@studium.fernuni-hagen.de

Projekt-Homepage: http://sourceforge.net/projects/openpearl/

7.3.1 Einleitung

Im Rahmen der Bachelor-Arbeit mit dem Titel „Entwicklung einer Modultestumgebung für eine Echtzeitprogrammiersprache mit der Möglichkeit der automatischen Testausführung“ wurde eine Modultestumgebung für das OpenPEARL-Projekt entwickelt.

Die implementierte Modultestumgebung bietet die bisher fehlende Werkzeugunterstützung, um Modultests im OpenPEARL-Projekt einzusetzen. Als Basis wird das Testframework Google Test eingesetzt. Mit Hilfe von Erweiterungen an Google Test wird eine nahtlose Integration in das OpenPEARL-Projekt erreicht.

Die Komponenten des OpenPEARL-Systems (inkl. Modultestumgebung) sind in Abbildung 1 dargestellt.

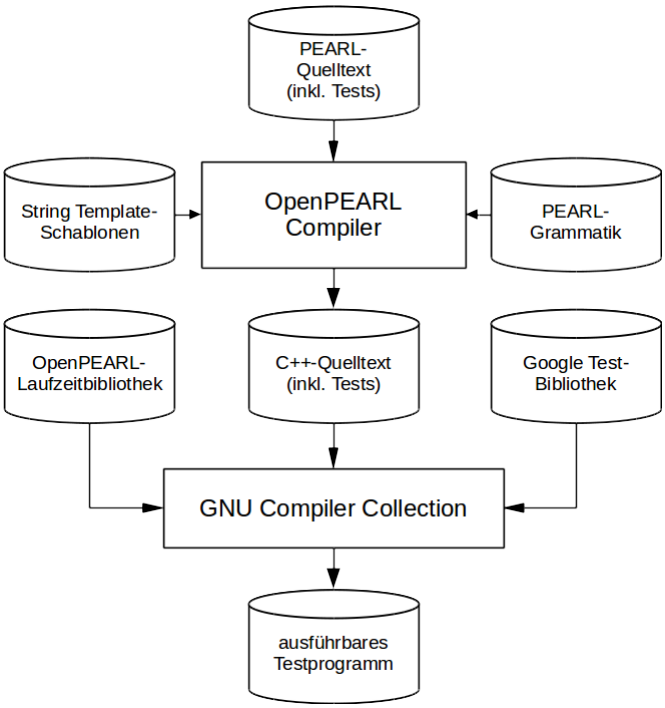

Abbildung 1: Komponenten des OpenPEARL-Systems

7.3.2 Beispiel

Im folgenden Test wird die Summen-Funktion `sum` getestet. Das Textmakro `EXPECT_EQ` erwartet, dass die beiden Ausdrücke den gleichen Wert ergeben. Das Testmakro `EXPECT_NE` erwartet, dass die beiden Ausdrücke verschiedene Werte ergeben.

```
TEST( sum, example )
{
  EXPECT_EQ( 20, sum(10,10) );
  EXPECT_NE( 20, sum(10,10) );
}
```

Die Ausführung des ersten Testmakros ist erfolgreich. Bei der Ausführung des zweiten Testmakros wird eine Fehlermeldung generiert. Das Ergebnis der Testausführung lautet folgendermaßen:

```
[==========] Running 1 test from 1 test case.
[ RUN      ] sum.example
OpenPEARL location, sum_gtest.prl: 15 Failure
Expected: 20 != sum(10,10), actual: 20 vs 20
[  FAILED  ] sum.example (0 ms)
[----------] 1 test from sum (0 ms total)
[==========] 1 test from 1 test case ran. (0 ms total)
[  PASSED  ] 0 tests.
[  FAILED  ] 1 test, listed below:
[  FAILED  ] sum.example
```

Anhand der Testausgabe kann man direkt die fehlerhafte OpenPEARL-Quelltextzeile ablesen und erhält Informationen über den fehlgeschlagenen Testausdruck angezeigt.

7.3.3 Zusammenfassung

Mit Hilfe der entwickelten Modultestumgebung können Modultests direkt im OpenPEARL-Quelltext definiert werden. Die Syntax der unterstützen Testausdrücke wurde von Google Test übernommen. Die Ausführung der Tests findet auf Basis des C++-Quelltexts mit Hilfe des Google Test-Frameworks statt. Die Ausgaben der Modultestumgebung enthalten aber die Informationen des originalen OpenPEARL-Quelltexts.

Zusätzlich zur Kommandozeilenausgabe besteht die Möglichkeit, die Testergebnisse in einer Datei im XML-Format zu speichern. Damit lassen sich die Ergebnisse langfristig speichern und sie können an externe Software-Produkte weitergegeben werden.

8 Trauer um Prof. Dr. Leberecht Frevert (1933 – 2016)

Dr. Peter Holleczek, Friedrich-Alexander Universität Erlangen-Nürnberg, RRZE

Professor Frevert war ein Mann der ersten Stunde bei der Entwicklung der Echtzeit-Programmiersprache PEARL. Vom Hahn-Meitner-Institut in Berlin aus steuerte er erste Ideen bei. 1972 wurde er zum Bevollmächtigten im Projekt „Prozesslenkung mit DV-Anlagen“ im Kernforschungszentrum Karlsruhe, dem Fördergremium für die PEARL-Sprachdefinition und -Pilotimplementierungen. 1975 wechselte er als Professor für Prozessdatenverarbeitung im Fachbereich Elektrotechnik an die Fachhochschule Bielefeld. Von dort aus beteiligte er sich weiterhin am PEARL-Geschehen und gehörte von 1982 bis 1991 dem Vorstand des PEARL-Vereins an. In dieser Zeit engagierte er sich hauptsächlich für die Öffentlichkeitsarbeit, als Betreuer der PEARL-Spalte in der Zeitschrift atp (Automatisierungstechnische Praxis) und als Herausgeber der PEARL-Mail. Nach dem Übergang des PEARL-Vereins in die GI-Fachgruppe Echtzeitprogrammierung in 1991 gab er bis 1998 den Rundbrief PEARL-News heraus. Sein Hauptwerk war indes die Herausgabe des Lehrbuchs „Echtzeitpraxis mit PEARL“ bei Teubner/Springer.

2013 wurde er in Anerkennung seiner Leistung auf dem Gebiet der Echtzeitsysteme und der Entwicklung der Programmiersprache PEARL zum Ehrenmitglied des GI-Fachausschusses Echtzeitsysteme ernannt.

Wir haben ihn als liebenswerten Menschen kennen und schätzen gelernt. Technisch sattelfest konnte er sich akribisch mit Detailfragen auseinander setzen und die Lösungen verständlich publizieren. Im persönlichen Gespräch freute es ihn – oft umgeben von Zigarrenrauch – Dinge auf den Punkt zu bringen und Pointen zu präsentieren, notfalls garniert mit Anekdoten aus seiner Lippeschen Heimat. Seine zurückhaltende und ausgleichende Art wird uns fehlen.