

# Echt Zeit

Nr. 9, März 2019

Mitteilungen  
des GI/GMA/ITG-Fachausschusses  
Echtzeitsysteme



GESELLSCHAFT FÜR INFORMATIK E.V.



**VDE**

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](mailto:herwig.unger@fernuni-hagen.de)
- Stellvertreter** Dipl.-Inform. (Univ.) Marcel Schaible  
FernUniversität in Hagen  
Fakultät für Mathematik und Informatik  
58084 Hagen  
[marcel.schaible@fernuni-hagen.de](mailto:marcel.schaible@fernuni-hagen.de)
- Redaktion** PD Dr.-Ing. habil. Mario Kubek  
Dipl.-Ing. Jutta Düring  
FernUniversität in Hagen  
Lehrstuhl für Kommunikationsnetze  
58084 Hagen  
[echtzeit@fernuni-hagen.de](mailto:echtzeit@fernuni-hagen.de)
- ISSN** 2199-9244

Redaktionell abgeschlossen am 20. März 2019

### 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 Jubiläumsjahr 2019
- 2 Tagung Echtzeit 2019 – Call for Papers
- 3 Einreichung von Vorschlägen zum Graduiertenwettbewerb 2019
- 4 Neuwahl des Stellvertretenden Sprechers des Fachausschusses
- 5 12th Conference on Autonomous Systems 2019
- 6 Veranstaltungen mit Unterstützung des Fachausschusses
- 7 50 Jahre PEARL in eingebetteten Systemen

# 1 Jubiläumsjahr 2019

Jutta Düring, Fachausschuss Echtzeitsysteme

In diesem Jahr haben wir gleich zwei Jubiläen zu verzeichnen:

## 50 Jahre Echtzeitprogrammiersprache PEARL



1968 förderte das Bundesministerium für Bildung und Wissenschaft das „Projekt NDV13“ – die Ausarbeitung einer Sprachvorlage unter dem Arbeitstitel „EXOS“ (EXperimentOrientierte Sprache).

Im Sommer 1970 erschien die erste Veröffentlichung „PEARL – The Concept of a Process and Experiment Oriented Programming Language“. Ausführliche Informationen folgen im Beitrag „50 Jahre PEARL in eingebetteten Systemen“.

## 40 Jahre Tagung in Boppard

Der PEARL-Verein veranstaltete die erste Tagung im November 1980 in Boppard. Nach der Umgründung des PEARL-Vereins in die GI-Fachgruppe 4.4.2 „Echtzeitprogrammierung, PEARL“ fand im Dezember 1992 die erste PEARL-Tagung unter Leitung der GI-Fachgruppe statt.

Die ersten Tagungsbände von 1989 bis 1997 erschienen unter dem Titel „PEARL – Workshop über Realzeitsysteme“, von 1998 bis 2007 unter „PEARL“ und Angabe des Leitthemas. Im Jahr 2008 erfolgte die Umbenennung in „Echtzeit“.



## 2 Tagung Echtzeit 2019 – Call for Papers

Jutta Düring, Fachausschuss Echtzeitsysteme

Homepage: <http://www.real-time.de/CfP.html>

Die Tagung „Echtzeit 2019“ findet wie gewohnt im Hotel Ebertor in Boppard am Rhein statt. Das Leitthema im Jubiläumsjahr lautet:

**„Autonome Systeme – 50 Jahre PEARL“  
21. und 22. November 2019**

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

- Perspektiven von EZ-Systemen
- Modellierung und Simulation
- Koordination und Vernetzung
- Bilderkennung und -verarbeitung
- Funktionale und IKT-Sicherheit
- KI unter Echtzeitbedingungen
- Automatische Sprachverarbeitung
- Ambient Assisted Living
- Aktuelle Anwendungen
- Ausbildung

Stichtag für die Vortragsanmeldung ist **Sonntag, der 21. April 2019.**

E-Mail: [tagung@real-time.de](mailto:tagung@real-time.de).

### 3 Einreichung von Vorschlägen zum Graduiertenwettbewerb 2019

Jutta Düring, Fachausschuss Echtzeitsysteme

Homepage: <http://www.real-time.de/preise/grad-bedingung.html>

Bis zu drei herausragende Diplom-, Bachelor- und Masterarbeiten im Bereich Echtzeitsysteme werden mit je 500,- € prämiert. Seit letztem Jahr können auch **Doktorarbeiten** vorgeschlagen werden.

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 14. April 2019.**

Vorschläge sind zu richten an Frau Prof. Dr. Juliane Benra ([benra@jade-hs.de](mailto:benra@jade-hs.de)).

### 4 Neuwahl des Stellvertretenden Sprechers des Fachausschusses

Jutta Düring, Fachausschuss Echtzeitsysteme

In der Mitgliederversammlung vom 15.11.2018 wurde **Marcel Schaible** einstimmig ohne Enthaltungen zum neuen Stellvertretenden Sprecher des Fachausschusses Echtzeitsysteme gewählt.

Der Sprecher des Fachausschusses Prof. Unger dankte dem bisherigen Stellvertretenden Sprecher Prof. Dr. Dieter Zöbel für seine geleistete Arbeit.



Verabschiedung von Prof. Dr. Dieter Zöbel (Mitte)  
durch Prof. Unger (rechts) und Prof. Halang (links)

## 5 12th Conference on Autonomous Systems 2019

Jutta Düring, Fachausschuss Echtzeitsysteme  
Homepage: <http://www.confautsys.org>

Der GI-Arbeitskreis Echtzeitkommunikation beschäftigt sich mit Echtzeitfragen des Internet und sozialer Netze. Die von ihm veranstaltete Konferenz „Autonome Systeme“ findet von

**Donnerstag, den 24. Oktober bis**  
**Dienstag, den 29. Oktober 2019**

in Cala Millor auf Mallorca statt. Der Jahresband erscheint in den „Fortschritt-Berichten“ des VDI-Verlages.



## 6 Veranstaltungen mit Unterstützung des Fachausschusses

Jutta Düring, Fachausschuss Echtzeitsysteme

**10. Jahreskolloquium „KommA“**  
Homepage: <http://www.jk-komma.de>

**Termin: 20./21. November 2019**  
**Ort: Magdeburg**

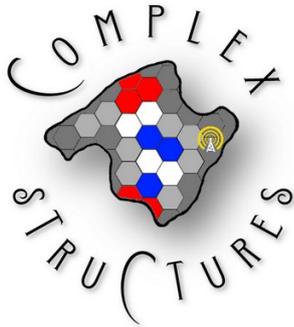


Das Jahreskolloquium „KommA“ des Instituts für industrielle Informationstechnik (inIT) der Hochschule OWL und des Instituts für Automation und Kommunikation (ifak) e.V. findet 2019 in

Magdeburg statt. Das Kolloquium ist ein Forum für Wissenschaft und Industrie im deutschsprachigen Raum für alle technisch/wissenschaftlichen Fragestellungen rund um die industrielle Kommunikation.

## 4th Workshop on Complex Structures 2019

Homepage: <http://www.fernuni-hagen.de/kn/complex-structures/>



**Termin: 31. März – 4. April 2019**

**Ort: Cala Millor, Mallorca**

Der Workshop ist gedacht als ein Forum zur Präsentation von neuen – nicht nur mathematischen – Methoden und Analysen, welche die wachsende Flut an Informationen bewältigen. Ziel ist es, Wissenschaftler und Praktiker aus verschiedenen Bereichen zusammen zu bringen, um die selben oder (anscheinend) völlig verschiedene Probleme zu besprechen und Lösungen durch interdisziplinäre Diskussionen zu finden.

Der Workshop ist offen für eine Teilnahme von Hochschullehrern und Doktoranden.

## 7 50 Jahre PEARL in eingebetteten Systemen

Wolfgang A. Halang, FernUniversität in Hagen, [wolfgang.halang@fernuni-hagen.de](mailto:wolfgang.halang@fernuni-hagen.de)

Marcel Schaible, FernUniversität in Hagen, [marcel.schaible@fernuni-hagen.de](mailto:marcel.schaible@fernuni-hagen.de)

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 gebaut und getestet. Ebenfalls begann 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 – eine Sprache, die durch konzeptionelle Klarheit und Eleganz besticht und ein von anderen bisher unerreichtes Muster zur Strukturierung und Programmierung verteilter Systeme darstellt.

PEARL erfüllt die allgemeinen Anforderungen an eine strukturierte Hochsprache für den Bereich eingebetteter Echtzeitsysteme: Einerseits bietet es die üblichen Konstrukte technisch-wissenschaftlich orientierter Sprachen mit besonders strenger Typprüfung an und andererseits stellt es 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 Problemtteilen 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 Problemtteilen 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 z.B. 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 bspw. 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. Wie das folgende Beispiel verdeutlicht, zeigt PEARL hier seine besondere Stärke. In der für die Programmierung eingebetteter Systeme z.Zt. vorherrschenden nicht echtzeitfähigen Sprache C benötigt man im Zusammenwirken mit einem Betriebssystem – hier QNX – das folgende, völlig unverständliche Codesegment

```
#include <stdio.h>
#include <time.h>
#include <sys/timers.h>
#include <sys/proxy.h>
#include <sys/kernel.h>
#include <signal.h>
void main()
{
    pid_t timer_proxy, end_proxy, task_pid;
    timer_t id, id_2, i;
    struct itimerspec timer, timer_end;
    struct itimercb timercb, timercb_end;
    timer_proxy=qnx_proxy_attach(0,0,0,0);
    if(timer_proxy==-1){printf("Unable to attach timer_proxy"); return;}
    timercb.itcb_event.evt_value=timer_proxy;
    id=mktimer(TIMEOFDAY, _TNOTIFY_PROXY,&timercb);
    if(id==-1){printf("Unable to attach timer\n");}
    timer.it_value.tv_sec=4L;
    timer.it_value.tv_nsec=0L;
    timer.it_interval.tv_sec=1L;
    timer.it_interval.tv_nsec=0L;
```

```

reltimer(id,&timer,NULL);
end_proxy=qnx_proxy_attach(0,0,0,0);
if(end_proxy==-1){printf("Unable to attach end_proxy"); return;}
timercb_end.itcb_event.evt_value=end_proxy;
id_2=mktimer(TIMEOFDAY,_TNOTIFY_PROXY,&timercb_end);
if(id_2==-1){printf("Unable to attach timer\n");}
timer_end.it_value.tv_sec=time(NULL)+10;
timer_end.it_value.tv_nsec=0L;
timer_end.it_interval.tv_sec=0L;
timer_end.it_interval.tv_nsec=0L;
abstimer(id_2,&timer_end,NULL);
for(;;)
    {Receive(timer_proxy,0,0);
      spawn1(P_NOWAIT,"task1","task1",0,0,0,);
      printf("Tick\n");
      if((task_pid=Creceive(end_proxy,0,0))==end_proxy)
{rmtimer(id); rmtimer(id_2); return;}}
}

```

um die periodische Aktivierung einer Task einzuplanen, wozu in PEARL nur eine einzige, auch für Nichtfachleute unmittelbar verständliche und selbstdokumentierende Anweisung ausreicht:

```
AT 12:00:00 ALL 1 SEC UNTIL 12:00:10 ACTIVATE Task1 PRIORITY 5;
```

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 am leichtesten vermitteln und verstehen. Das hat positive Konsequenzen auf Qualität und Sicherheit der später von den Absolventen erstellten Systeme. Direkt mit der „C-Tür ins Haus zu fallen“ (siehe oben), wäre deshalb didaktisch unsinnig. Derzeit wird PEARL an rund 50 Universitäten, Fachhochschulen und Berufsakademien in Deutschland und Europa für die Ingenieurausbildung eingesetzt.

Die Spracherweiterung Mehrrechner-PEARL hieße nach aktueller Nomenklatur besser „PEARL für verteilte Systeme“, denn es handelt sich um die weltweit einzige genormte Sprache für solche Systeme. Sie stellt Möglichkeiten zur Kommunikation zwischen Rechnerknoten mittels Nachrichtenaustausch sowie zur dynamischen Rekonfiguration, aber auch, ohne dabei ausführbaren Code zu erzeugen, zur Beschreibung von Konfigurationen und zur Parameterversorgung von Übersetzern, Bindern und Ladern zur Verfügung. Die Architekturbeschreibung eines verteilten Systems umfaßt Angaben über seine konstituierenden Rechnerknoten, 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 PEARL-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.

Mit PEARL hatte die deutsche Informationstechnik auf dem Gebiet der Echtzeitsysteme die international führende Stellung erreicht, denn die Sprache ist ihren Konkurrenten wegen der Klarheit und geringen Komplexität ihrer Konstrukte sowie ihrer die funktionale Sicherheit fördernden leichten Lesbarkeit und Verständlichkeit technisch weit überlegen, weshalb ihre Konzepte auch bei der Standardisierung eines Satzes von Unterprogrammen zur Erweiterung von FORTRAN um Echtzeitfähigkeit zu „Industrial Real-Time FORTRAN“ durch den „International Purdue Workshop on Industrial Computer Systems“ Pate stand. Allein das französische 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 30 Jahren gemachten Erfahrungen fanden 1998 in der Norm DIN 66253-2 „PEARL90“ ihren Niederschlag, mit der die Teile 1 und 2 der DIN 66253 abgelöst wurden.

PEARL90-Übersetzer sind für Linux und eine Reihe z.T. echtzeitfähiger Betriebssysteme verfügbar. Einer dieser Übersetzer generiert nicht Maschinen-, sondern C-Code, der dann auf beliebigen Plattformen weiterverwendet werden kann. Für den Einsatz in eingebetteten Systemen mit hohen Verlässlichkeitsanforderungen besonders hervorzuheben ist das konzeptionell auf unmittelbare Unterstützung der Echtzeitkonstrukte von PEARL zugeschnittene Realzeitbetriebssystem RTOS-UH des Instituts für Regelungstechnik der Leibniz-Universität Hannover, für das der Status der „Betriebsbewährtheit“ durch Nachweis von über 9 Millionen Stunden fehlerfreien Betriebs 18 exemplarischer Anwendungen erlangt werden konnte. Die Gesamtzahl industrieller Einsatzfälle der vielfältigsten Art dürfte im oberen fünfstelligen Bereich liegen. Vorwiegend eingesetzt werden PEARL und auch RTOS-UH von auf Automatisierungstechnik spezialisierten mittelständischen Firmen und Ingenieurbüros, die mit Hilfe dieser sehr effizienten Werkzeuge ihren Wettbewerbs- und Produktivitätsvorsprung realisieren. Um auch das für die Automatisierungstechnik so wichtige Gebiet der speicherprogrammierbaren Steuerungen abzudecken, wurde mit PEARL-PLC eine Abbildung der Funktionsplansprache gemäß der Norm IEC 61131-3 nach PEARL realisiert.

Im Jahre 2012 begann eine Arbeitsgruppe des gemeinsamen Fachausschusses „Echtzeitsysteme“ der Gesellschaft für Informatik (GI), der VDI/VDE-Gesellschaft für Meß- und Automatisierungstechnik (GMA) sowie der Informationstechnischen Gesellschaft (ITG) im VDE, einen auf der PEARL90-Sprachnorm basierenden Übersetzer „OpenPEARL“ zu entwickeln, der zusammen mit einem passenden Laufzeitsystem für Linux-basierte Systeme frei erhältlich ist. Im Zuge dieser Arbeiten wurde die Norm kritisch gesichtet, was zur Aufdeckung von Widersprüchen und nicht eindeutigen Formulierungen sowie in der Folge zu zahlreichen Präzisierungen und Vereinfachungen der Sprache führte, die dann in OpenPEARL Einzug hielten. Erfahrungsgemäß höchst selten eingesetzte oder technisch obsolet gewordene Konstrukte wurden ebenso aus ihr entfernt wie solche, die zu fehlerträchtigen, unstrukturierten, schwer analysierbaren und nicht modular aufgebauten Programmen mit nicht-deterministischem Laufzeitverhalten führen und strenge Datentypisierung behindern.

Darauf aufbauend wurden in Fortführung der seit 1973 andauernden Praxis, PEARL mittels Auswertung im industriellen Einsatz gewonnener Erfahrungen immer weiter zu fokussieren und zu vereinfachen, unter dem Namen „SafePEARL“ durch Zusammenfügung und Modifikation

von OpenPEARL und Mehrrechner-PEARL eine sicherheitsgerichtete Grundsprache zur Echtzeitprogrammierung sowie darin für jede der vier Sicherheitsintegritätsstufen nach der internationalen Norm IEC 61508 eine wegen der Beschränkung auf das Notwendigste einfache, leicht verständliche und somit inhärent sichere Teilsprache definiert, deren Syntax die Einhaltung der jeweils geltenden Einschränkungen erzwingt. Die den beiden oberen Sicherheitsstufen SIL 4 und SIL 3 zugeordneten Teilsprachen für Ursache-/Wirkungstabellen bzw. Funktions- und Ablaufpläne auf der Grundlage verifizierter Bibliotheken sind die einzigen, die hohen Sicherheitsanforderungen genügende Programme derzeit einfach und wirtschaftlich zu verifizieren gestatten. Für die unteren Sicherheitsstufen wurden textuelle Teilsprachen definiert, und zwar für SIL 2 eine formaler Programmverifikation zugängliche Sprache und für SIL 1 eine statische Sprache mit sicheren Konstrukten zum Einsatz im asynchronen Mehrprozeßbetrieb.

Weniger sichere Sprachmerkmale werden schrittweise zu höheren Sicherheitsstufen hin zur Programmformulierung nicht mehr zugelassen, weshalb nicht für jede Stufe eine neue Sprache gelernt werden muß und Übersetzer prüfen können, ob Programme bestimmte Sicherheitsauflagen erfüllen. Das Prinzip, Teilmengen einer Sprache für kritische Anwendungen zu definieren, gestattet es, Programme nach bestimmten Sicherheitsanforderungen zu entwickeln und Codeabschnitte für sicherheitskritische und -unkritische Systemteile nahtlos miteinander zu verbinden. Je sicherheitskritischer ein System ist, desto restriktivere Methoden sind einzusetzen. Zur ihrer Durchsetzung wird mittels Übersetzerdirektiven das jeweils gewünschte Sicherheitsniveau vorgegeben, so daß alle notwendigen Prüfungen bereits zur Übersetzungszeit durchgeführt werden können. In einem Codeabschnitt sind nur Konstrukte der Sprachteilmenge der bezeichneten und der höheren Sicherheitsintegritätsstufen zulässig. Codeabschnitte mit unterschiedlichen Sicherheitsanforderungen sind schachtelbar, um so transparent miteinander verbunden werden zu können.

Mit SafePEARL und seinen ineinander geschachtelten, spezifischen Teilmengen für die vier Sicherheitsintegritätsstufen nach IEC 61508 wurde eine weltweit einzigartige Sprache zur Programmierung eingebetteter, verteilter Echtzeitsysteme geschaffen, die sämtliche bekannten Sprachmittel zur Förderung funktionaler Sicherheit in sich vereinigt und sich an der menschlichen Verständnissfähigkeit orientiert. Leitmotiv bei ihrer Formulierung war das Bestreben, größtmögliche Korrektheit, Konsistenz und Genauigkeit der Darstellung bei geringst möglichem Umfang zu erreichen. Aus diesem Grunde wurden die vier den Sicherheitsintegritätsstufen zugeordneten Teilsprachen nicht einzeln beschrieben, sondern durch folgende Tabelle definiert, die angibt, welche Konstrukte der Grundsprache (SIL 0) auf den Sicherheitsstufen SIL 1 bis SIL 4 jeweils verwendet werden dürfen. Gemäß Beschluß des DIN-Normenausschusses NA 043-01-22 AA „Programmiersprachen“ ersetzt SafePEARL seit 2018 die älteren Versionen nach DIN 66253 Teil 3 und DIN 66253-2.

Gepflegt wird PEARL durch den Fachausschuß „Echtzeitsysteme“, der u.a. zweimal jährlich die Zeitschrift „EchtZeit – der Rundbrief“ herausgibt und jährlich im November die Fachtagung „Echtzeit“ durchführt, deren Tagungsbände im Springer-Verlag in der Reihe „Informatik aktuell“ erscheinen.

Für die aktuellsten Informationen zu PEARL und RTOS-UH sei auf die Internet-Auftritte des Fachausschusses (<http://www.real-time.de>) und der Firma IEP GmbH (<http://www.pearl90.de/>, <http://rtos.iep.de/>) verwiesen. Auf ersterem findet sich zur freien Verfügung ein PEARL-Sprachreport in englischer Sprache für den internationalen Markt und ein Verweis auf den frei verfügbaren OpenPEARL-Übersetzer (<http://sourceforge.net/projects/openpearl/>).

<b>Anweisung/Klausel</b>	<b>SIL:</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
GOTO, EXIT		+	-	-	-	-
(bedingte) Ausdrücke und Zuweisungen		+	+	+	-	-
bedingte Anweisungen und Anweisungsauswahl		+	+	+	-	-
Angabe physikalischer Einheiten		+	+	+	+	+
Ursache-Wirkungstabellen		+	+	+	+	+
sequentielle Ablaufpläne		+	+	+	+	-
Synchronisierung mit SEMA- und BOLT-Variablen		+	-	-	-	-
Synchronisierung mit LOCK und TIMEOUT-Klausel		+	+	+	-	-
Verwendung interner Signale		+	+	+	-	-
Verwendung von Unterbrechungssignalen		+	+	+	-	-
Verwendung von Tasks		+	+	+	-	-
mit Prioritäten		+	+	-	-	-
mit Fristenangaben und Zeitüberwachung		+	+	+	-	-
(Funktions-) Prozeduraufrufe		+	+	+	+	-
Wiederholungen		+	+	-	-	-
mit MAXLOOP-Klausel		+	+	+	-	-
Verwendung von Zeigern und Referenzen		+	-	-	-	-
PUT/GET, WRITE/READ, CONVERT		+	+	+	-	-
TAKE/SEND		+	+	+	+	+
verteilte Systeme		+	+	+	+	+
dynamische Rekonfiguration		+	+	+	+	-
Botschaftenaustausch		+	+	+	-	-

**PEaRL**

**50 Jahre**

## 40. Tagung Echtzeit Autonome Systeme

Vorgestellt werden sollen Beiträge zu

- 50 Jahre Echtzeitprogrammiersprache PEaRL
- Perspektiven von EZ-Systemen
- Modellierung und Simulation
- Koordination und Vernetzung
- Bilderkennung und -verarbeitung
- Funktionale und IKT-Sicherheit
- KI unter Echtzeitbedingungen
- Aktuelle Anwendungen
- Ausbildung

**21.- 22. Nov.  
2019**

**Vortragsanmeldung : 21. April 2019**

Graduiertenwettbewerb  
Besonders erbeten sind aus **studentischen Abschlussarbeiten** sowie aus **Dissertationen** entstandene Beiträge, von denen bis zu drei der besten mit Preisen ausgezeichnet werden.

Veranstaltungsort  
Hotel Ebertor  
Boppard am Rhein

Tagungsorganisation und Vortragsanmeldung  
Dipl.-Ing. Jutta Düring  
Telefon: 02331 987-4529  
<https://www.real-time.de/echtzeit.html>  
E-Mail: [tagung@real-time.de](mailto:tagung@real-time.de)