

# Echt Zeit

Nr. 4, Februar 2016

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](mailto:wolfgang.halang@fernuni-hagen.de)

**Stellvertreter** Prof. Dr. Dieter Zöbel  
Universität Koblenz-Landau  
Institut für Softwaretechnik  
56016 Koblenz  
[zoebel@uni-koblenz.de](mailto:zoebel@uni-koblenz.de)

**Redaktion** Prof. Dr.-Ing. habil. Herwig Unger  
Dipl.-Ing. Jutta Düring  
FernUniversität in Hagen  
58084 Hagen  
[pearl@fernuni-hagen.de](mailto:pearl@fernuni-hagen.de)

**ISSN** 2199-9244

Redaktionell abgeschlossen am 3. Februar 2016

### 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:

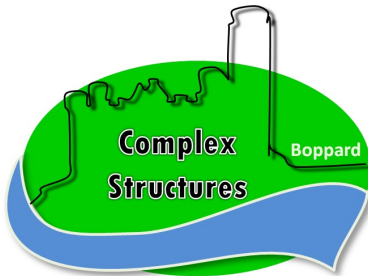
- 1 Veranstaltungen mit Unterstützung des Fachausschusses
- 2 Conference Autonomous Systems 2016
- 3 Tagung Echtzeit 2016: Call for Paper
- 4 Best Paper Award und Graduiertenwettbewerb 2015
- 5 Fachartikel: OpenPEARL Übersetzer – Projektfortschritt
- 6 Fachartikel: Änderungen in der Sprachdefinition in OpenPEARL in Bezug auf PEARL90 – Teil 2
- 7 Zusammenfassungen von Abschlussarbeiten:  
    Aufruf zur Einreichung  
    Prozessorarchitektur zum Einsatz unter sicherheitsgerichteten Echtzeitbedingungen
- 8 Archivierung unseres Rundbriefs
- 9 Online-Aktivitäten unserer Tagungsbände bei Springer

# 1 Veranstaltungen mit Unterstützung des Fachausschusses

Jutta Düring, Fachausschuss Echtzeitsysteme

## 1st GI Workshop on Complex Structures

20. – 23. März 2016, Hotel Ebertor in Boppard am Rhein



Structures and structural aspects reflect, determine and represent in a manifold manner the functionality, behaviour and properties of various systems and inter-system communication. Resulting from a system's working history, they may occur as reaction or adaptation to its environment, or even by processes of self-organisation and emergence. Consequently, efficient analysis of (not only graph-based) structures, recognition of special patterns in them as well as understanding of building, transformation and evolution laws are gaining importance not only in computer science. With the size of practically relevant systems

and structures growing daily, this workshop is intended as a forum for presenting methods to master the growing amount of information by new — not only mathematical — methods and analyses, models and simulations as well as by algorithms to recognise, consider, predict or influence structural changes. Since methods seems to be similar in different areas of application and research, it is the goal of the workshop to bring practitioners and scientists from different areas together, discuss the same or (apparently) completely different problems and find solutions for their problems by interdisciplinary discussions.

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

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

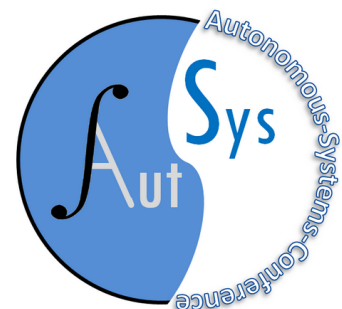
## 2 Conference Autonomous Systems 2016

Jutta Düring, Fachausschuss Echtzeitsysteme

Der GI-Arbeitskreis *Echtzeitkommunikation* veranstaltet in diesem Jahr die „9th GI Conference on Autonomous Systems“, welche vom **23. bis 28. Oktober 2016** in Cala Millor auf Mallorca stattfinden wird. Der Tagungsband wird in den „Fortschritt-Berichten“ des VDI-Verlages erscheinen.

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

Das Programm der diesjährigen Tagung wird besonders werden, da mehrere Jubiläen anstehen. Lassen Sie sich überraschen!



### 3 Tagung Echtzeit 2016: Call for Paper

Jutta Düring, Fachausschuss Echtzeitsysteme

Die Tagung „Echtzeit“ findet in diesem Jahr am **17. und 18. November 2016** wie gewohnt in Boppard am Rhein statt. Unser diesjähriges Leitthema lautet:

**„Internet der Dinge“.**

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
- Schnittstellen, Protokolle und Plattformen
- Energieeffizienz
- Diensterechnung und -bezahlung
- Erweiterte Realität (Augmented Reality)
- Altersgerechte Assistenzsysteme für selbstbestimmtes Leben (Ambient Assisted Living)
- Aktuelle Anwendungen
- Ausbildung

Stichtag für die Vortragsanmeldung ist **Mittwoch, der 20. April 2016**. Nähere Informationen finden Sie unter <http://www.real-time.de/CfP.html>.

### 4 Best Paper Award und Graduiertenwettbewerb 2015

Jutta Düring, Fachausschuss Echtzeitsysteme

Als besten Tagungsbeitrag der „Echtzeit 2015“ hat das Programmkomitee

**„Ein hierarchisches Scheduling-Modell für unbekannte Anwendungen mit schwankenden Ressourcenanforderungen“**

von Vladimir Nikolov, Franz J. Hauck und Lutz Schubert prämiert.

([http://dx.doi.org/10.1007/978-3-662-48611-5\\_6](http://dx.doi.org/10.1007/978-3-662-48611-5_6))

Die Beiträge der Gewinner des Graduiertenwettbewerbs 2015 lauten

- **Christian Ritzler:** Testanwendungen zur Überprüfung des PEARL-Systems auf Sprachkonformität ([http://dx.doi.org/10.1007/978-3-662-48611-5\\_13](http://dx.doi.org/10.1007/978-3-662-48611-5_13))
- **Peter Wägemann:** Assisting Analysis Based on Implicit Path Enumeration through Evolutionary Testing for Energy Awareness ([http://dx.doi.org/10.1007/978-3-662-48611-5\\_12](http://dx.doi.org/10.1007/978-3-662-48611-5_12))

## 5 OpenPEARL Übersetzer – Projektfortschritt

Rainer Müller, Hochschule Furtwangen, [mueller@hs-furtwangen.de](mailto:mueller@hs-furtwangen.de)

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

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

### 5.1 Einleitung

Der Übersetzer für die Programmiersprache OpenPEARL soll eine für Ausbildungszwecke frei verfügbare Programmierumgebung bereitstellen. Dabei werden OpenPEARL Quelltexte mittels eines ANTLR generierten Übersetzers in semantisch äquivalente C++ Quelltexte transformiert. In einem weiteren Übersetzungsschritt wird mit Hilfe der GNU Toolchain ein auf Linux ausführbares Programm erzeugt (siehe Abb. 1). Weitere Informationen befinden sich auf der Projektseite und in [1]. Im weiteren wird der aktuelle Projektfortschritt beschrieben und einen Ausblick auf Erweiterungen gegeben.

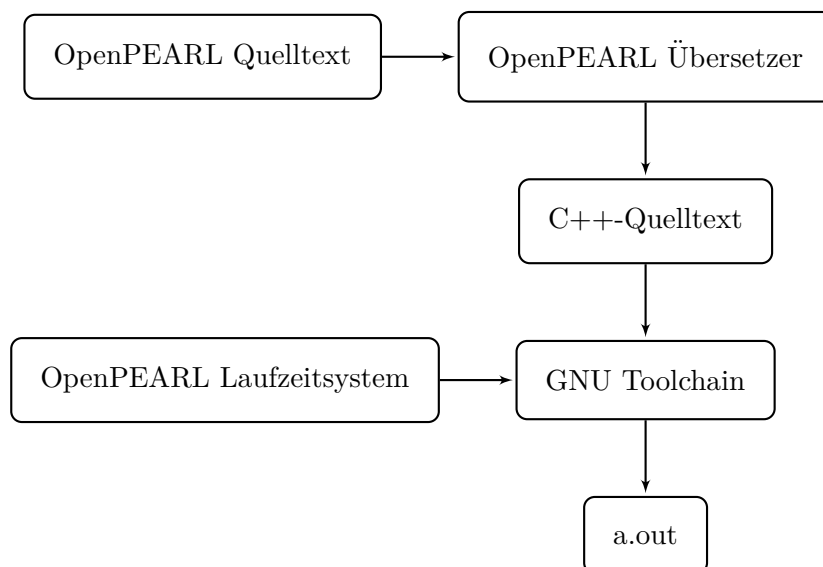


Abbildung 1: OpenPEARL Umgebung

### 5.2 Projektfortschritt

Das Projekt befindet sich nach wie vor in der Entwicklung, so dass hier der aktuelle Entwicklungsstand wiedergegeben wird. Die Autoren haben Ende des Jahres 2014, nachdem das ursprüngliche Inkubatorprojekt *SmallPEARL* nicht mehr den Projektstand, insbesondere bzgl. des angestrebten Sprachumfangs, widerspiegelte, das Projekt in **OpenPEARL** umbenannt. Diese Umbenennung hatte eine neue Projektheimat zur Folge. Die aktuellen Quellen sind nun unter der Internet-Adresse <http://sourceforge.net/projects/openpearl> frei verfügbar.

Auf der o.g. Projektseite werden alle Informationen inklusive einiger Diskussionforen rund um OpenPEARL bereitgestellt. Einen Übersetzer zu entwickeln ist kein leichtes Unterfangen und benötigt eine entsprechende Anzahl von Mitarbeitern. In den letzten Jahren wurden einige

Bachelor- und Masterarbeiten im Umfeld von OpenPEARL erfolgreich abgeschlossen oder werden zur Zeit bearbeitet. Die Weiterentwicklung, Fehlerbehebung und Betreuung der Studierenden erfordert einen immensen Einsatz der Autoren. Unter anderem deswegen wurde Anfang April 2015 an der FernUniversität in Hagen am Lehrstuhl für Informationstechnik von Prof. Halang eine halbe Mitarbeiterstelle geschaffen und damit eine Entspannung obiger Situation erzielt. Nichtsdestotrotz ist die Mitarbeit weiterer OpenPEARLianer dringend vonnöten. Ein aktualisierter Projektplan mit der Einplanung weiterer, offener PEARL Sprachkonstrukte findet sich auf der Projektseite.

### 5.2.1 Semantische Analyse

Ein erster Entwurf der semantischen Analyse wurde im Rahmen einer Bachelorarbeit an der FernUniversität in Hagen erfolgreich abgeschlossen. Hierbei betrachtet die Arbeit folgende semantische Prüfungen:

- Gültigkeitsbereiche von Variablen und Prozeduren
- Typkonformität in arithmetischen Ausdrücken
- Prüfung der Übergabeparameter von Prozeduren gegen ihre Signaturen

Die Implementierung wurde auf der Projektseite in einem eigenen Entwicklungszweig bereitgestellt. Die Anpassung, Ausbau und abschließende Integration der semantischen Analyse ist als eine der nächsten Aufgaben geplant.

### 5.2.2 Codegenerierung

Der OpenPEARL-Übersetzer generiert den C++-Zielcode mittels des Java Template Frameworks *StringTemplate* [3]. Dadurch wird eine strikte Trennung zwischen Analyse, semantischer Prüfung und der eigentlichen Codeerzeugung erreicht. Während der Entwicklung der Codeerzeugung wurde schnell festgestellt, dass sich viele Codeschablonen wiederholen. Deswegen wurde in einer Bachelorarbeit untersucht, inwieweit man unter Verwendung eines generischen Ansatzes mittels vordefinierter Codeschablonen hier eine übersichtlichere und flexiblere Darstellung erreichen kann.

### 5.2.3 Modultest

Neben der eigentlichen Entwicklung des Übersetzers ist eine umfangreiche Testsuite entstanden. Bei der Testentwicklung wurde sehr schnell deutlich, dass eine komfortablere Methode zur Testfalldefinition und -ausführung nützlich wäre. Aus diesem Grund entsteht zur Zeit in einer weiteren Abschlussarbeit ein solcher Testrahmen. Der Funktionsumfang ist an die Testumgebungen *CppUnit* bzw. *Google Test* angelehnt. Folgende Punkte werden hierbei unter anderem betrachtet:

- Ausführliche Dokumentation des Testfortschritts in einer Protokolldatei.
- Möglichkeit der gezielten Ausführung von einzelnen Tests der Testsuite.

- Bei einem Testausfall soll anwenderfreundlich direkt auf die entsprechende OpenPEARL-Quellcodezeile verwiesen werden.
- Möglichkeit der Definition von Negativtests.

### 5.3 Themen für zu vergebende Abschlussarbeiten

Momentan ist folgendes Thema zum OpenPEARL-Übersetzer für eine studentische Abschlussarbeit zu vergeben:

„Graphische Entwicklungsumgebung für eine Echtzeitprogrammiersprache mit der Möglichkeit der Visualisierung der Task-Synchronisation“

Weitere Themenstellungen werden im Wiki-Bereich der Projektplattform laufend aktualisiert. Interessenten wenden sich bitte direkt an die Autoren.

### 5.4 Ausblick

Folgende Punkte werden neben den bereits oben erwähnten Arbeitspaketen und der allgemeinen Fehlerbehebung bearbeitet:

- Definition von Codeblöcken (BEGIN ... END)
- Schleifen (FOR und REPEAT)
- Sprungmarken
- Prozeduren (PROCEDURE, RETURN und CALL)
- VOID Datentyp
- Unterbrechungen und Signale

#### 5.4.1 Aktive Projektbeteiligung

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

## Literatur

- [1] R. Müller, M. Schaible: Die Programmierumgebung OpenPEARL90. In: Industrie 4.0 und Echtzeit – Echtzeit 2014, Springer Vieweg, 2014, S. 31–40, DOI: 10.1007/978-3-662-45109-0.
- [2] GI Fachgruppe 4.4.2: PEARL90 – Sprachreport, Version 2.0, 1995.
- [3] T. Parr: The Definitive ANTLR Reference: Building Domain-Specific Languages; Pramatic Bookshelf; 2007.

## 6 Änderungen in der Sprachdefinition in OpenPEARL in Bezug auf PEARL90 – Teil 2

Rainer Müller, Hochschule Furtwangen, [mueller@hs-furtwangen.de](mailto:mueller@hs-furtwangen.de)

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

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

### 6.1 Einleitung

Wie schon in [1] dargestellt ist die Sprachdefinition von PEARL90 [2] weder widerspruchsfrei noch eindeutig formuliert. In Ergänzung zu den dort bereits vorgestellten Konstrukten wurden auf der Sitzung des Arbeitskreises OpenPEARL am Rande der Tagung Echtzeit 2015 in Boppard folgende Änderungen festgehalten.

### 6.2 Restriktivität der Sprache

Die Sprache wird so weiterentwickelt, dass sich ein sehr restriktives Verhalten bei Extremsituationen ergibt. Es werden bei allen problematischen Punkten in der Programmausführung SIGNALS erzeugt. Dies sind:

1. Insbesondere die „Fehlermeldung“ durch Sternchenausgabe in PUT-Anweisungen im F-Format wird durch ein Signal ersetzt.
2. Ebenso ist die Erzeugung von Signalen bei Bereichsüberschreitungen von FIXED, FLOAT und DURATION verpflichtend. Bei CLOCK ist derzeit festgelegt, dass alle Ergebnisse modulo 24 Stunden gerechnet werden, d.h. hier gibt es kein Signal bei einer Bereichsüberschreitung.  
*Ob nicht auch hier ein Signal erzeugt werden muss sollte noch eingehend diskutiert werden.*
3. Bei CHARACTER sieht die aktuelle Sprachnorm keine Regelung vor, wie bei Zuweisungen auf einen kürzeren String verfahren werden soll. Entsprechend des restriktiven Verhaltens soll in diesem Fall ein Signal erzeugt werden. Dies gilt ebenso für Ausdrücke vom Typ BIT.

### 6.3 Berechnung des Zeitpunktes von Einplanungen

Die Sprachnorm legt fest, dass zeitliche Einplanungen mit relativen und absoluten Zeitangaben erfolgen können. Es wurde bislang nicht festgelegt, wie mit Zeitumstellungen und Schaltsekunden umzugehen ist.

Hier gibt es nun die Festlegung, dass eine Änderung der Uhrzeit eine Neuberechnung der Einplanungszeitpunkte verursacht. Bei näherer Betrachtung dieser Festlegung kann man die Vorgehensweise ableiten, wie mit Änderungen der Systemzeit im Detail umzugehen ist bei übersprungenen Einplanungszeitpunkten.

#### Beispiel:

- AT 12:00 ACTIVATE T2;
- um 11:50 wird die Systemzeit um 20 Minuten vorgestellt — also auf 12:10



Man könnte argumentieren, dass die Einplanung dann sofort wirksam wird. Wie ist dann zu verfahren, wenn um 11:50 die Systemzeit um 23 HRS 40 MIN zurückgestellt wird? Soll dann ebenfalls die Einplanung sofort wirksam werden, da der Einplanungszeitpunkt überstrichen wurde?

Dies bedarf noch detaillierter Betrachtung.

Eventuell ist der in OpenPEARL bislang gewählte Ansatz, die absoluten Zeiten zum Zeitpunkt der Einplanung in relative Zeiten umzurechnen, nicht so ungünstig, wie dies zunächst erscheinen mag. Die Probleme mit der Zeitumstellung muss in diesem Fall der Anwendungsprogrammierer selbst lösen, indem er z.B. die Einplanungen mit PREVENT löscht und neu setzt. Hierzu könnte ein softwarebasierter Interrupt vom Laufzeitsystem bereitgestellt werden, der der Anwendung anzeigt, dass eine Zeitumstellung erfolgt ist.

*Eine Festlegung für die Sprachnorm in diesem Punkt sollte erst nach ausführlicher Erörterung und Diskussion erfolgen.*

## **6.4 Ein-/Ausgabe**

### **6.4.1 TFU und TFU(MAX)**

Der Sprachstandard legt fest, dass die Angabe einer Transfergröße aus Kompatibilitätsgründen zu früheren PEARL-Systemen zulässig ist. Eine konkrete Funktion dieser Elemente ist nicht fixiert.

Diese Elemente wurden nun aus der Sprachnorm entfernt, so dass die Vereinbarung von Datenstationen ohne Altlasten erfolgt.

### **6.4.2 Ein-/Ausgabeformat S**

Dieses Format erlaubt es, bei der Texteingabe die Länge des aktuell eingegebenen Textes zu ermitteln. Dies mag sinnvoll sein, da PEARL bei der Eingabe mit dem A-Format den eingegebenen Text mit Leerzeichen auffüllt, bis die CHAR(x)-Eingabevariable komplett gefüllt ist.

Dieses Eingabeformat wurde selten benutzt. Dieses Verhalten lässt sich auch selbst nachbauen, indem die Eingabevariable im Anwendungsprogramm entsprechend untersucht wird. Dieses Format wird aus Gründen der Sprachvereinfachung aus der Sprachnorm entfernt.

### **6.4.3 Ein-/Ausgabeformat L**

Dieses Format erlaubt eine generische Ausgabe von Werten beliebigen Typs in einem typkonformen Format.

Dieses Format wird ebenfalls aus der Sprachnorm entfernt. Der Anwendungsprogrammierer soll das geeignete Ausgabeformat selbst auswählen.

#### 6.4.4 Skalierungsparameter im F-Format

Das Ein-/Ausgabeformat erlaubt derzeit die Eingabe und Ausgabe von Werten mit den Datentypen FIXED und FLOAT. Damit bei Systemen ohne Gleitpunktarithmetik dennoch im Bereich der Ein- und Ausgabe mit Dezimalzahlen gearbeitet werden kann, stellt das F-Format als dritten Parameter einen Skalierungsfaktor zur Verfügung.

Der Arbeitskreis war der Meinung, dass dieses Element nicht mehr zeitgemäß sei. Daher wird der Skalierungsfaktor im F-Format aus der Sprachnorm gestrichen.

In Konsequenz ergibt sich, dass das F-Format für Ein- und Ausgabe von FIXED-Werten nur den Parameter für die Feldbreite benötigt. Die Form des F-Formats mit zwei Parametern ist damit nur noch für die Ein- und Ausgabe von FLOAT Werten sinnvoll. Die Regelung, dass bei fehlendem zweiten Parameter dieser mit 0 angenommen wird, bleibt davon unberührt. FLOAT-Werte können auf diese Weise auch ganzzahlig ein- und ausgegeben werden.

### 6.5 Objektorientierte Ansätze

#### 6.5.1 Overloading

Die aktuelle Sprachnorm erlaubt Overloading von Datenstrukturen. Es wird schon in der Sprachnorm dargestellt, dass dies das strenge Typkonzept durchbricht.

Dies ist ein Widerspruch zu dem geforderten strikten Verhalten der Sprache. Daher wird Overloading nicht weiter unterstützt.

#### 6.5.2 Operator-Funktionen

Die aktuelle Sprachnorm erlaubt die Definition von Operatorfunktionen in ähnlicher Weise, wie dies in C++ möglich ist, um Operationen mit den gebräuchlichen Operatoren beschreiben zu können.

Dies kann sehr schnell zu schwer verständlichen Programmen führen. Daher wird auf dieses Konzept verzichtet und darauf verwiesen, dass diese Funktionalitäten mit Prozeduren (PROC) nachbildbar sind.

#### 6.5.3 Referenzen auf TASK und PROC

Zeiger erfreuen sich in der C-Programmierung einer großen Beliebtheit. Die Probleme im Umgang mit Zeigern sind allgemein bekannt.

Daher wird auf REF, TASK und REF PROC im neuen Sprachumfang verzichtet.

## 6.6 Systemteil

Der Systemteil in PEARL90 erlaubt die Angabe von bis zu vier FIXED-Parametern für die Eigenschaften von Systemgeräten. Weiter sagt der Sprachreport aus, dass der Systemteil weitgehend durch die jeweilige Rechnerplattform definiert ist.

Diese Einschränkung ist z.B. für Anwendungen auf aktuellen Rechnersystemen unhandlich, da diese meist eine textuelle Beschreibung der Systemgeräte und deren Parameter nutzen. Weiterhin sind Bussysteme in heutigen Rechnersystemen allgegenwärtig.

Um diesen geänderten Anforderungen gerecht zu werden und dennoch eine einfache Syntax zu erhalten, wurde folgende Syntax im Systemteil von OpenPEARL umgesetzt. Die Sprachnorm soll die bisherige Freiheit bzgl. der jeweiligen Rechnerplattform beibehalten.

**Beispiel:** Systemteil und Spezifikation im Problemteil

```
MODULE ( newSyntax ) ;
SYSTEM;
  i2cbus : INOUT : i2cbus_1 ( '/dev/i2c1' , 100000 ) ;
  tempsensor1 : IN : i2cbus LM75( '43'B4 ) ;
PROBLEM;
SPC tempsensor1 DATION IN SYSTEM BASIC FIXED(15);
```

Die Änderungen sind:

- Zwischen Benutzernamen und Systemnamen wird per Doppelpunkt ein Parameterfeld für die Transferrichtung eingefügt.
- So genannte Busgeräte erlauben die Verwendung des Benutzernamens als Platzhalter der Verbindung zum Systemgerät an diesem Bus. Die Trennung erfolgt durch ---.
- Jedes Systemgerät kann eine vom Autor des Systemgerätetreibers definierte Anzahl von Parametern haben. Diese sind hier z.B. der Bezeichner der Schnittstelle im unterlagerten Betriebssystem sowie die Übertragungsgeschwindigkeit beim Gerät `i2cbus`.
- Systemgeräteparameter sind Konstanten aller einfachen Datentypen von PEARL.
- Der Import eines Benutzernamens für ein Systemgerät erfolgt im Problemteil mit der SPC-Anweisung, welche um die Angabe des Schlüsselwortes `SYSTEM` und eines Datentyps oder der Angabe von `ALL` ergänzt wurde. Damit kann der Compiler die Korrektheit der aktuellen Typen in der `TAKE`- und `SEND`-Anweisung überprüfen.

*Anmerkung:* Auf die Angabe eines Datentyps im Systemteil wurde verzichtet, da das Systemgerät selbst am besten weiß, welcher Datentyp korrekt ist. Dies wird in OpenPEARL durch einen separaten Übersetzungsschritt sichergestellt.

## Literatur

- [1] R. Müller, M. Schaible: Änderungen in der Sprachdefinition in OpenPEARL in Bezug auf PEARL90. In: GI/GMA/ITG-Fachausschuss Echtzeitsysteme (Hrsg.): EchtZeit. 2015, Nr. 3, S. 8–13.
- [2] GI Fachgruppe 4.4.2: PEARL90 – Sprachreport, Version 2.0, 1995.

## 7 Zusammenfassungen von Abschlussarbeiten

Jutta Düring, Fachausschuss Echtzeitsysteme

### 7.1 Aufruf zur Einreichung

Die Abstracts von Abschlussarbeiten und Dissertationen aus den Einrichtungen unserer Mitglieder mit starkem Bezug zu PEARL oder Echtzeitproblemen sollen als fester Bestandteil in den Rundbrief aufgenommen werden. Ziel ist die Darstellung der Breite der aktuellen Forschungsarbeiten in diesem spannenden Gebiet.

Die Leser werden gebeten, auf geeignete Arbeiten zu achten und die betreffenden Zusammenfassungen mit den ergänzenden Daten an die Redaktion ([pearl@fernuni-hagen.de](mailto:pearl@fernuni-hagen.de)) zu übermitteln.

Der Umfang der Zusammenfassung sollte ca. 15–20 Textzeilen umfassen. Die Einreichung sollte Angaben über Verfasser, Hochschule, Kontaktadresse (E-Mail) und evtl. eine URL für weitere Informationen enthalten.

### 7.2 Prozessorarchitektur zum Einsatz unter sicherheitsgerichteten Echtzeitbedingungen

Daniel Koß, M.Sc., FernUniversität in Hagen, Lehrgebiet Informationstechnik  
[daniel.koss@fernuni-hagen.de](mailto:daniel.koss@fernuni-hagen.de)

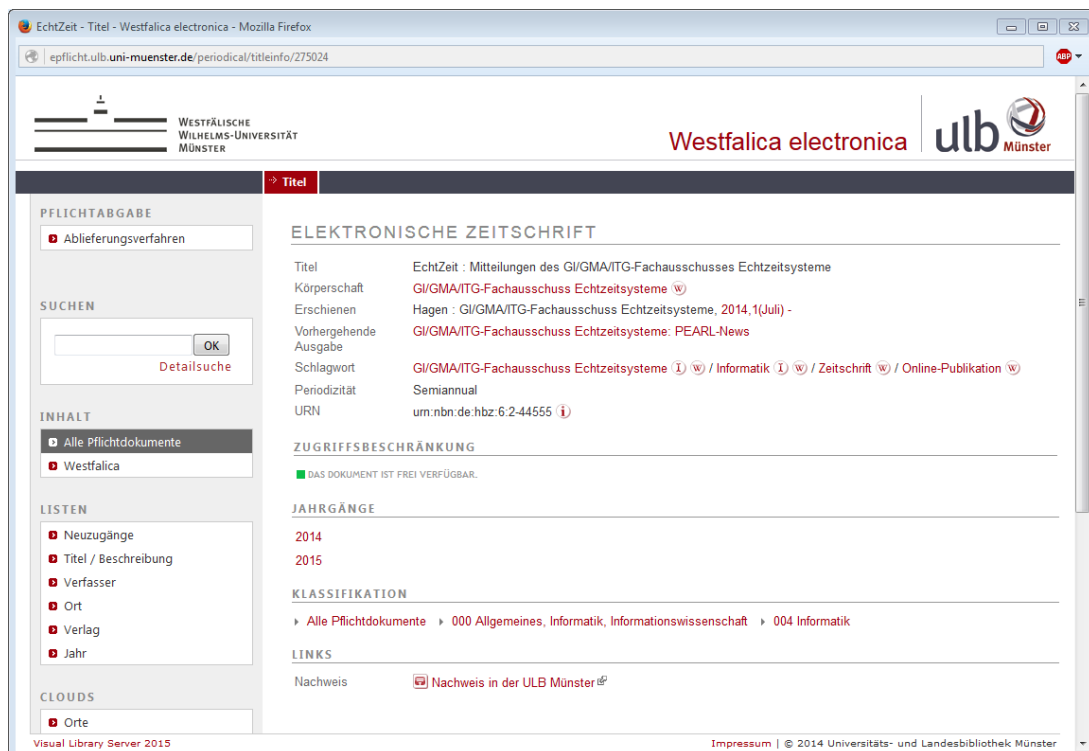
Moderne Prozessorarchitekturen, die auch in der sicherheitsgerichteten Echtzeitdatenverarbeitung eingesetzt werden, basieren auf Strukturen und Konzepten, die zum Teil aus den Anfängen der Informationstechnik hervorgegangen sind. Dies führte dazu, dass moderne Prozessorarchitekturen nicht geeignet sind, um sicherheitsgerichtete Echtzeitfunktionen auszuführen. Von der Anforderungserhebung an sicherheitsgerichtete Echtzeitsysteme über die Konzeptentwicklung bis zur abschließenden Bewertung wird eine Prozessorarchitektur entworfen, die ohne Altlasten und mit überraschend einfachem Aufbau selbst höchste Anforderungen an die Ausfallsicherheit erfüllt. Dabei wird gezeigt, dass nicht notwendigerweise Einbußen an Funktionsumfang oder Einsatzmöglichkeiten hingenommen werden müssen.

Diese Abschlussarbeit wurde auch auf der Tagung „Echtzeit 2015“ vorgestellt. Den Beitrag zum Tagungsband finden Sie unter [https://dx.doi.org/10.1007/978-3-662-48611-5\\_3](https://dx.doi.org/10.1007/978-3-662-48611-5_3).

## 8 Archivierung unseres Rundbriefs

Jutta Düring, Fachausschuss Echtzeitsysteme

Neben der Deutschen Nationalbibliothek, die unseren Rundbrief in Papierform archiviert, gibt es auch eine Pflichtexemplar-Sammlung der Bundesländer. Auswahlkriterium für das zuständige Bundesland ist der Verlagsort. Zur Zeit ist dies Hagen, also Nordrhein-Westfalen. Für den Regierungsbezirk Arnsberg ist die Universitäts- und Landesbibliothek Münster zuständig. Hier wird die elektronische Version unseres Rundbriefs „*EchtZeit*“ archiviert und frei zur Verfügung gestellt.

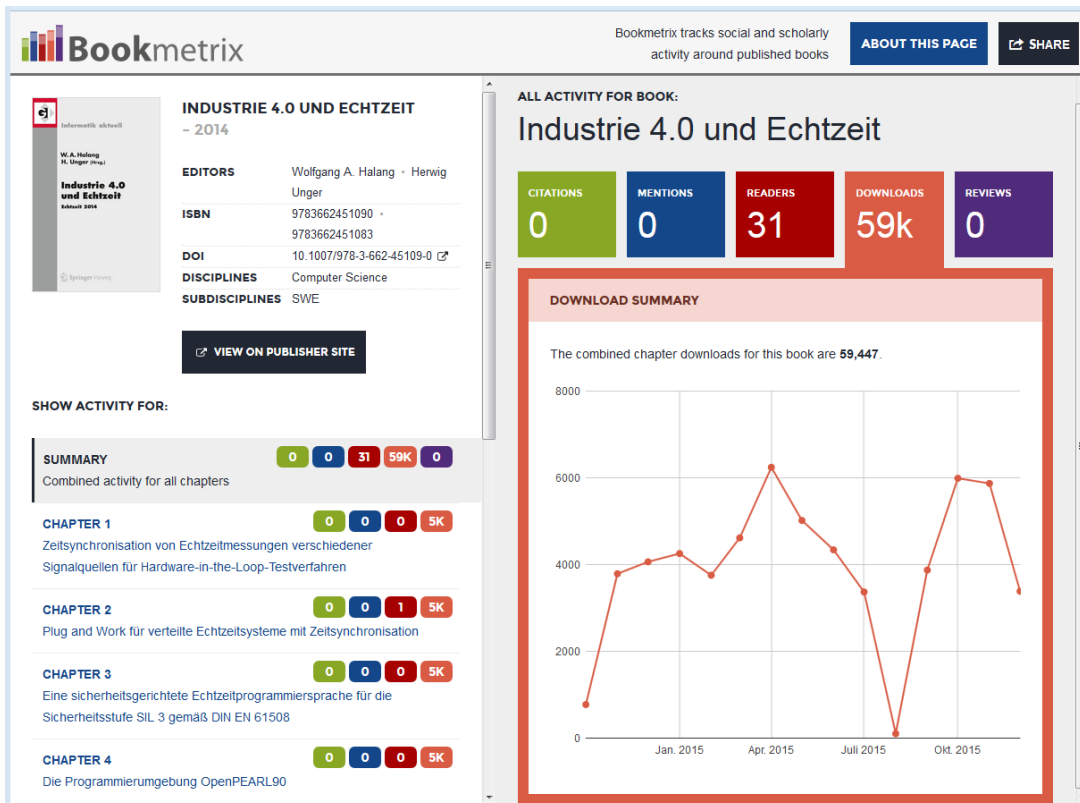


## 9 Online-Aktivitäten unserer Tagungsbände bei Springer

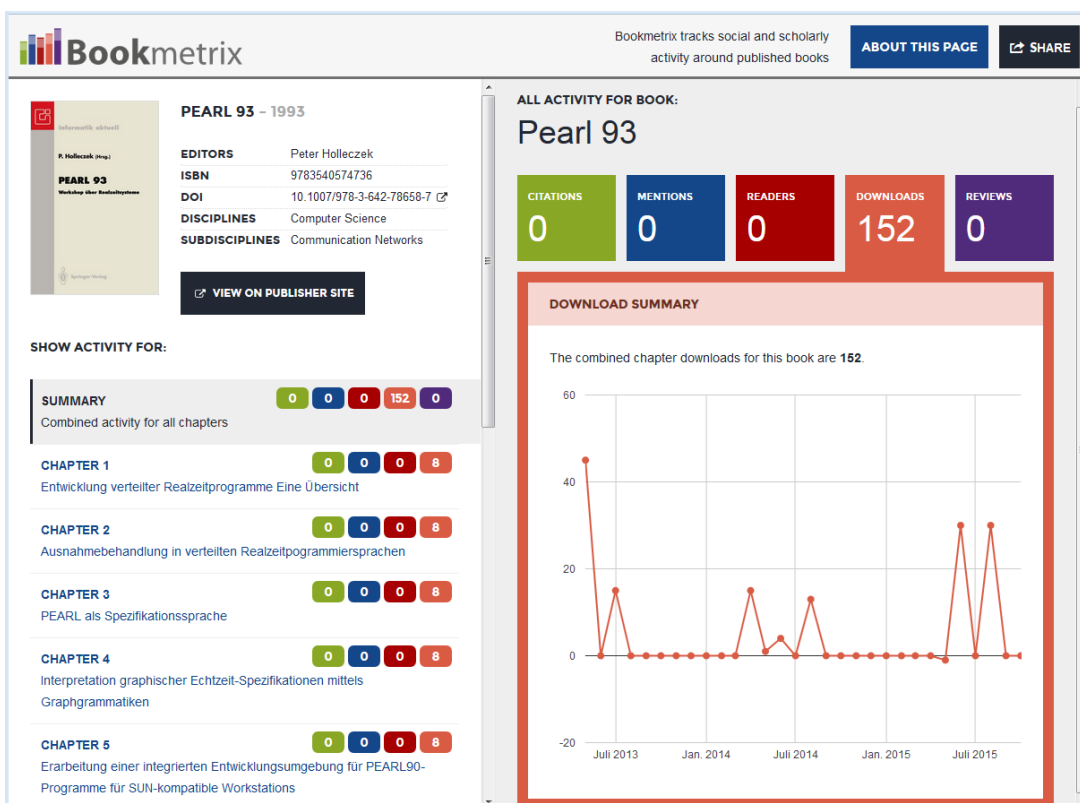
Jutta Düring, Fachausschuss Echtzeitsysteme

Der Springer-Verlag stellt auf seiner Plattform *SpringerLink* mit *Bookmetrix* verschiedene Kennzahlen zu Büchern und sogar einzelnen Buchkapiteln dar. Die Metriken geben Aufschluss darüber, wie oft ein Buch gelesen, heruntergeladen, rezensiert, zitiert und im Internet erwähnt oder geteilt wurde.

Die Downloads unseres Tagungsbandes *Echtzeit 2014* liegen bei knapp 60.000 innerhalb eines Jahres. Der Tagungsband *Echtzeit 2015* wurde in den zwei Monaten seit Veröffentlichung immerhin 1.700 mal herunter geladen. Und selbst bei den älteren, nachträglich digitalisierten Tagungsbänden liegen die Download-Zahlen bei ca. 150. Allerdings in gut 2,5 Jahren... ☺



Echtzeit 2014: Erfassung von Oktober 2014 bis Dezember 2015



PEARL 1993: Erfassung von Mai 2013 bis Oktober 2015