



PEARL-News

Ausgabe 1/2010

Mitteilungen
der GI-Fachgruppe Echtzeitsysteme

ISSN 1437-5966

Impressum

Herausgeber	GI-Fachgruppe Echtzeitsysteme (RT) URL: http://www.real-time.de
Sprecher	Dr. P. Holleczeck Universität Erlangen-Nürnberg, Regionales Rechenzentrum Martensstraße 1, D-91058 Erlangen Telefon: 09131/85-27817 Telefax: 09131/30 29 41 E-Mail: peter.holleczeck@rrze.uni-erlangen.de
Stellvertreter	Prof. Dr. Dr. W. Halang FernUniversität in Hagen Universitätsstraße 27 - PRG D-58084 Hagen Telefon: 02331/987-372 Telefax: 02331/987-375 E-Mail: wolfgang.halang@fernuni-hagen.de
Redaktion	Prof. Dr. R. Müller FH Furtwangen, Fachbereich Computer- & Electrical Engineering Robert-Gerwig-Platz 1, 78120 Furtwangen Telefon: 07723/920-2416 Telefax: 07723/920-2610 E-Mail: mueller@hs-furtwangen.de
ISSN	1437-5966

Redaktionell abgeschlossen am 6. Juli 2010

Einreichung von Beiträgen

Diese Zeitschrift soll nicht nur Mitteilungsblatt sein, sondern auch eine Plattform für den Informations- und Meinungsaustausch zwischen allen an den Fragen der Echtzeitprogrammierung Interessierten bilden. Diskussionsstoff bzw. offene Fragen gibt es auf unserem Gebiet reichlich.

Wir möchten Sie, liebe Leserinnen und Leser, daher ausdrücklich ermuntern, auch in Zukunft die PEARL-News durch Ihre Beiträge mit zu gestalten. Für ein ausgewogenes Bild der News sollten Beiträge nicht länger als 5 Seiten sein.

Rainer Müller (Furtwangen)

Inhaltsverzeichnis

1 Programm des Workshops PEARL 2009	3
2 Neuer Arbeitskreis PEARL-Compiler?	4
3 Workshop zu Lehrsprachen in der Informatik	5
4 Rezension in der FKT	6

1 Programm des Workshops PEARL 2009

Der diesjährige Workshop steht unter dem Leitthema „Echtzeit 2010 - Eingebettete Systeme“. Er wird wieder im Hotel Ebertor in Boppard am Rhein stattfinden. Das Programmkomitee hat auf seiner Sitzung am 12. Mai 2010 in Frankfurt/Main das nachstehende Programm zusammengestellt. Der Tagungsband erscheint - wie in den vergangenen Jahren - in der Reihe „Informatik aktuell“ des Springer-Verlages.

Erster Workshop-Tag: Donnerstag, der 18. November 2010

11:00 Treffen der Arbeitskreise

13:00 Begrüßung

13:15 Sitzung 1: *Konzepte* (Sitzungsleitung: Prof. Kaltenhäuser)

Ressourcen-Effiziente Software Systeme: Modellierung, Vorhersage und Optimierung des Energiebedarfs

(C.Bunse, Universität Mannheim)

Notwendigkeit von Metriken für neue Programmiermethoden automatisierungstechnischer Anlagen

(S.Braun, Technische Universität München)

Entwicklung einer echtzeitfähigen CLI-Laufzeitumgebung für den Einsatz in der Automatisierungstechnik

(A.Schepeljanski, Technische Universität Chemnitz)

14:45 Pause

15:15 Sitzung 2: *Virtualisierung* (Sitzungsleitung: Prof. Heitmann)

Echtzeitfähigkeit virtueller Maschinen

(R.Kaiser, Fachhochschule Bingen)

Design eines Virtual Machine Monitors aufbauend auf einem Mikrokern

(T.Stumpf, Hochschule Furtwangen)

16:15 Pause

16:45 Sitzung 3: *Graduierungsvorträge* (Sitzungsleitung: Prof. Benra)

Superblock-basierte High-Level WCET-Optimierungen

(T.Kelter, Technische Universität Dortmund)

Entwurf und Realisierung eines Adapters für UniLoG zur Lastgenerierung an IP-basierten Schnittstellen

(M.Kulas, Universität Hamburg)

GPS Navigation on Ski Slopes

(Schmid, Eidgenössische Technische Hochschule Zürich)

18:00 Abendessen

19:15 Abendprogramm

Mitgliederversammlung der Fachgruppe

Zweiter Workshop-Tag: Freitag, der 19. November 2010

9:00 Sitzung 4: *Anwendungen* (Sitzungsleitung: Prof. Schiedermeier)

Rapid Prototyping eingebetteter Systeme zur echtzeitfähigen Sortierung von Schüttgütern

(R.Heintz, Fraunhofer-Institut für Optronik Karlsruhe)

Echtzeitverarbeitung von RFID Daten durch dezentrale Systemarchitekturen

(P.Sukjit, FernUniversität Hagen)

Ein Echtzeitregler zur Kollisionsvermeidung von Flugrobotern
(Eckert, Universität Erlangen)

10:30 Pause

11:00 Sitzung 5: *Hardwarenahe Systeme* (Sitzungsleitung: Prof. Müller)

IEEE 1588-Leistungstester mit Emulation von Umwelteinflüssen
(C.Hallas, Hochschule Ostwestfalen-Lippe)

Smart Metering: Entwicklung eines Intelligenten Stromzählers
(D. Benyoucef, Fachhochschule Furtwangen)

FPGA-basierte System on Chip Plattform in der Echtzeitbildverarbeitung
(B.Schwarz, Hochschule für Angewandte Wissenschaften Hamburg)

12:30 Verabschiedung

12:45 Imbiss

2 Neuer Arbeitskreis PEARL-Compiler?

PEARL wäre eine wunderbare Programmiersprache in der Ausbildung, sofern ein für Studierende gut zugänglicher Compiler verfügbar wären. Derzeit gibt es das RTOS-UH System und einen PEARL-Compiler der Fa. Werum.

Der "Werum-Compiler" war lange Zeit eine gute Basis für die Ausbildung an den Hochschulen, nachdem die Fa. Werum freundlicherweise die Linuxversion für nichtkommerzielle Anwendungen freigegeben hatte. Leider haben diverse Änderungen an den Linuxdistributionen dazu geführt, dass das Laufzeitsystem nur noch auf einigen Linuxdistributionen funktioniert.

Das RTOS-UH System ist an spezielle Hardware gebunden oder unter einem ATARI-Emulator unter Windows mit den "Echtzeiteigenschaften von Windows" verfügbar.

Meine Beobachtung in den Fakultäten Informatik und Computer- & Electrical Engineering der Hochschule Furtwangen zeigt, dass Linux sich auch den Rechnern der Studenten gegenüber Windows durchgesetzt hat.

Aus dieser Situation rührt die Idee einen PEARL-Compiler zu erstellen, der

- auf Linux läuft und für Linux übersetzt
- native Linuxressourcen benutzt
- evtl. sogar real-time-Linux mit berücksichtigt
- unter GPL o.ä. steht

Dieses Projekt ist für eine Einzelperson - auch mit einer Reihe von Studenten im Hintergrund - nicht zu bewältigen. Ich fände es toll, wenn sich in der *Fachgruppe Echtzeitsysteme* Mitstreiter fänden, die dieses Projekt aktiv unterstützen.

Ohne große Analyse ist klar, dass

- ein Compiler mit syntaktischer und semantischer Analyse zu erstellen ist
- ein Treiberkonzept für Standard und Prozessperipherie zu entwickeln und zumindest rudimentär für die Standardperipherie zu implementieren ist
- ein Laufzeitsystem mit den PEARL-Standardfunktionen zu entwickeln ist
- die Abbildung der PEARL-Taskinganweisungen auf das unterlagerte Betriebssystem zu konzipieren und umzusetzen sind

Ich rufe daher alle Interessierten auf sich bei mir zu melden. Je nach Zuspruch und Know-How-Verteilung könnte am Rande der Tagung in Boppard ein Kick-Off-Meeting für einen neuen Arbeitskreis "PEARL-Compiler" stattfinden.

R.Müller
HS Furtwangen
mueller@hs-furtwangen.de

3 Workshop zu Lehrsprachen in der Informatik

Am Freitag, den 12.3.2010, fand von 10:00 bis 16.00 Uhr beim DIN e.V. in Berlin unter der Organisation des Unterausschusses NIA-01-22 „Programmiersprachen“ ein Workshop über Lehrsprachen in der Informatik statt.

Aktuell wird eine breite Palette von Programmiersprachen in der Lehre an fortführenden Schulen und im Studium eingesetzt. Sprachen wie C++, Java, C#, Ada, Basic, Logo, Pascal, Python, Scheme, PEARL und viele andere werden aktuell in Deutschland genutzt, um Schüler und Studierende an die Programmierung heranzuführen. Viele oft scheinbar widersprüchliche Anforderungen werden an Lehrsprachen gestellt:

- Verständlichkeit und Praxisorientierung
- Einfachheit und umfassende Einsatzbarkeit
- Unterstützung objektorientierter, prozeduraler und funktionaler Paradigmen
- Inhärente Sicherheit und Effizienz
- Intuitivität und logische Strenge
- Maschinennähe und hohes Abstraktionsniveau
- Klare Sprachkonzepte und breite Unterstützung durch Entwicklungswerkzeuge

Im Workshop wurden Python, Small Basic, Ada, C, C++, C#, Eiffel, Java, PEARL, Pascal und graphische Funktionsblocksprachen kurz vorgestellt und unter obigen Aspekten diskutiert. Einig waren sich alle, dass Programmierkonzepte auf einem hohen Niveau unterrichtet werden müssen. Während Small Basic als Zielgruppe für Schüler und Laien gedacht ist, zielen andere Programmiersprachen auf Studierende und professionelle Softwareentwickler ab. Sprachen, die in der Schule eingesetzt werden, sollten Konzepte vermitteln, auf denen die Hochschule aufsetzen kann. Der Fokus sollte auf Algorithmen liegen, das algorithmische Denken sollte in der Schule geweckt werden. Eine ideale Sprache für die Lehre muss nicht eine ideale Sprache für die Praxis sein. Da die Hochschulen aber für die Praxis ausbilden, war der Praxisbezug ein wichtiger Aspekt, den eine Sprache für die Lehre zu erfüllen hat. Als wesentlich wurde erachtet, dass eine Lehrsprache klare Strukturen enthält, sowie einen kleinen, leicht zu lernenden Kern, der durch diverse Bibliotheken ergänzt werden kann. Als weiter wichtig erachtet wurden Portierbarkeit, Plattformunabhängigkeit, Modularisierung und Skalierung. Die Sprache sollte Sprachmittel für Nebenläufigkeit, Parallelisierung bieten, sie sollte multiparadigmatisch sein, d.h. Sprachelemente sowohl für Objektorientierung als auch funktionale Elemente sowie imperative Elemente besitzen. Kontrovers diskutiert wurde die Notwendigkeit von Entwicklungsumgebungen. Während anfangs auf die Unterstützung durch IDEs verzichtet werden sollte, sind sie für professionelle Softwareentwicklung unerlässlich. Einig waren sich die Teilnehmenden, dass man unterscheiden sollte zwischen Einsteigersprachen und Spezialsprachen für bestimmte Anwendungsbereiche. An Programmiersprachen für sogenannte Bindestrich Informatik Studiengänge sind andere Anforderungen zu stellen, als an Programmiersprachen in Studiengängen für Kerninformatiker.

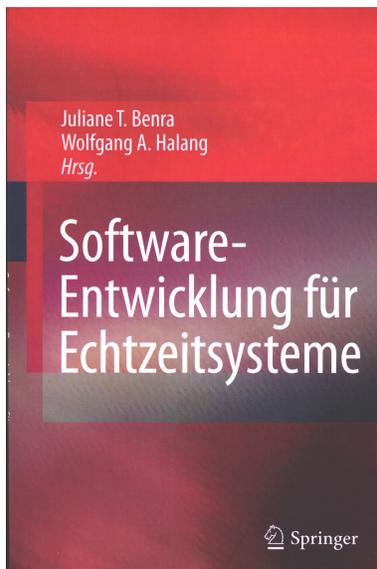
Prof. Gudrun Schiedermeier
Hochschule Landshut
gschied@fh-landshut.de

4 Rezension in der FKT

In der Fachzeitschrift für Fernsehen, Film und elektronische Medien (FKT) erscheint folgende Rezension:

Software-Entwicklung für Echtzeitsysteme

Juliane T. Benra, Wolfgang A. Halang (Hrsg.)



Warum dieses Buch als Besprechung in der FKT?

Programme in der Broadcasttechnik sind in der Regel Echtzeitsysteme. Viele Hersteller verwenden allerdings nicht echtzeitfähige Betriebssysteme und Programmiersprachen. Nach der Lektüre werden sie die Probleme verstehen, die sie mit Software-Herstellern haben und warum die Echtzeittechnik besonderer Aufmerksamkeit bedarf.

Die Entwicklung von Echtzeitsoftware ist als Nische im Überangebot von Softwaretools für Wald- und Wiesen-Programme leider nicht sehr oft in der Öffentlichkeit präsent. Mit dem Buch von Benra und Halang als Herausgeber haben sich 9 Autoren ein hohes Ziel gesetzt. Das Vorwort weist auf die stabile Quelle in den jährlichen Workshops in Boppard am Rhein hin. Dort treffen sich die „Echtzeitler“ und diskutieren die hehren Ziele der aus der Umgebung an den Interrupt gebundenen Programme.

Im Vorwort wird auch schon auf die Entwicklung in Deutschland mit PEARL hingewiesen. Diese Programmiersprache hat

die Echtzeit-Informatik eigentlich revolutioniert, kam aber durch die Zögerlichkeit der Hersteller von Prozessrechnern nicht überzeugend zum Zug. Das Buch geht in 8 Kapiteln auf 253 Seiten auf alle Belange der Prozessinformatik ein. Ein umfangreiches Literaturverzeichnis folgt danach.

Die Verfasser der einzelnen Kapitel sind ausnahmslos ausgewiesene Echtzeitler und sind auch immer noch mit der Weiterentwicklung der Echtzeitgedanken beschäftigt.

Die grundlegenden Begriffe aus den Echtzeitsystemen werden auch für Nichtinformatiker verständlich erläutert. Während es dann jedoch Informatikkenntnisse bedarf, um die übrigen Kapitel zu verstehen. Dafür ist es auch ein Fachbuch, das den Praktiker und den werdenden Informatiker alle Tücken erläutert. Der Stand der Technik bei verteilten Systemen sind ebenso wie die Entwurfsmethoden mit modernen Mitteln voll erschöpfend beschrieben. Der Unterschied in der Programmierung vor allem der beschränkte Umfang der Statements zeigen, wie auch heute noch PEARL deutlich übersichtlicher programmiert werden kann. Vorbildlich die Beschreibung der Synchronisation von Tasks. Betriebssysteme, die Unterschiede zu den allgemeinen Anforderungen werden mit Blick auf das Echtzeitverhalten an den 11 Systemdiensten erläutert. Multitasking, Mehrprozessor und Mehrbenutzersysteme werden anschaulich und kompetent durchleuchtet. Die Umsetzung Mehrprozessorbetrieb und die Unterbrechungsbehandlung, sowie die Möglichkeiten der Prozesszuteilung werden sehr anschaulich und verständlich beschrieben. Bei der Echtzeitsynchronisation wird ausführlich auf die Uhrensynchronisation quasi als Basis der Echtzeit eingegangen. Das Kapitel zur Programmierung geht auf PEARL, Ada und JAVA ein. Die Listings zu PEARL sind schwer zu lesen, liegt an der Schriftart. Die Qualitäten der drei Sprachen werden fair verglichen. Die Qualitätssicherung von Echtzeitsystemen ergeht sich in allgemeinen Betrachtungen zur Qualität von Software. Es entsteht der Eindruck, dass die Qualitätssicherung in der Echtzeitprogrammierung noch nicht vollständig implementiert ist. Die Darstellung als solche ist sehr verständlich.

Das Buch ist als Fachbuch für den Praktiker in den Firmen und auch die höheren Semester der Informatik konzipiert und wird in fast allen Punkten diesem Anspruch gerecht. Das Konzept PEARL wird immer wieder gelobt. Warum wird es dann nicht öfter und konsequenter eingesetzt? Lassen wir uns da nicht zu sehr von anderen ablenken?

D. Sauter
sauter@beenen.de, sauter@irt.de