Fortschritt-Berichte VDI

VDI

Reihe 8

Mess-, Steuerungs- und Regelungstechnik

Nr. 1257

Dipl.-Inform. Dipl.-Wirt.Inform. Sten Grüner, Ilvesheim

Ressourcenadaptive Anwendungen für die operative Prozessleittechnik

ACPLT AACHENER PROZESSLEITTECHNIK

Lehrstuhl für Prozessleittechnik der RWTH Aachen

Ressourcenadaptive Anwendungen für die operative Prozessleittechnik

Von der Fakultät für Georessourcen und Materialtechnik
der Rheinisch-Westfälischen Technischen Hochschule Aachen

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigte Dissertation

vorgelegt von **Dipl.-Inform. Dipl.-Wirt.Inform.**

Sten Grüner

aus Ekaterinburg, Russland

Berichter:	Univ.-Prof. Dr.-Ing. Ulrich Epple
	Univ.-Prof. Dr.-Ing. Stefan Kowalewski
Tag der mündlichen Prüfung:	23. Mai 2017

Fortschritt-Berichte VDI

Reihe 8

Mess-, Steuerungs- und Regelungstechnik

Dipl.-Inform. Dipl.-Wirt.Inform.
Sten Grüner, Ilvesheim

Nr. 1257

Ressourcenadaptive Anwendungen für die operative Prozessleittechnik

Lehrstuhl für
Prozessleittechnik
der RWTH Aachen

Grüner, Sten
Ressourcenadaptive Anwendungen für die operative Prozessleittechnik
Fortschr.-Ber. VDI Reihe 8 Nr. 1257. Düsseldorf: VDI Verlag 2017.
166 Seiten, 51 Bilder, 2 Tabellen.
ISBN 978-3-18-525708-7, ISSN 0178-9546,
€ 62,00/VDI-Mitgliederpreis € 55,80.

Für die Dokumentation: Prozessleittechnik – Softwarearchitektur – Ressourcenadaptive Anwendungen – IEC 61131 – Slackzeit – Scheduling – Laufzeitumgebung

Die aktuellen Entwicklungen der Prozessleittechnik und der Automatisierungstechnik fordern die Verlagerung zusätzlicher Funktionen auf die Prozessleitebene der Automatisierungspyramide. Die Ausführungszeit der anwenderspezifischen Logik innerhalb der Laufzeitsysteme unterliegt gewissen Fluktuationen. Die überschüssige Zeit, Slackzeit genannt, bleibt wegen der festen Zykluszeit häufig ungenutzt. Der Beitrag dieser Arbeit ist ein Rahmenwerk für die nahtlose Integration von ressourcenadaptiven Anwendungen in solche Laufzeitsysteme. Diese Anwendungen können für die Bereitstellung der zusätzlichen Funktionalität während der Slackzeit genutzt werden, ohne die Echtzeitanforderungen und den Funktionsumfang der existierenden Kernanwendung einzuschränken. Der Mehrwert des Rahmenwerks für die Prozessleittechnik wird an mehreren Use-Cases demonstriert. Dazu zählen Anwendungen mit und ohne Zugriff in den operativen Betrieb des Laufzeitsystems.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter http://dnb.ddb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek
(German National Library)
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at http://dnb.ddb.de.

D82 (Diss. RWTH Aachen University, 2017)

ISSN 0178-9546
ISBN 978-3-18-525708-7

Vorwort

Die vorliegende Arbeit wurde während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Prozessleittechnik der RWTH Aachen University angefertigt. Für die Förderung seitens der DFG im Rahmen des Graduiertenkollegs 1298 „AlgoSyn“ (Algorithmische Synthese reaktiver und diskret-kontinuierlicher Systeme) möchte ich mich an dieser Stelle gesondert bedanken.

Mein besonderer Dank gebührt meinem Doktorvater Herrn Univ.-Prof. Dr.-Ing. Ulrich Epple. Sein Gespür für aussichtsreiche Forschungsthemen an der Schnittstelle zwischen Automatisierungstechnik und Informatik, die Bereitschaft zum Ideenaustausch, die vertrauensvoll ermöglichten Freiräume und die durch ihn geschaffene einzigartige interdisziplinäre Atmosphäre am Lehrstuhl hat meine Arbeitsweise maßgeblich geprägt. Dies hat wesentlich zum Gelingen dieser Arbeit beigetragen.

Ebenso danke ich Herrn Univ.-Prof. Dr.-Ing. Stefan Kowalewski, Inhaber des Lehrstuhls für eingebettete Systeme der RWTH Aachen University, für die freundliche Übernahme der Zweitbegutachtung. Herrn Univ.-Prof. Dr.-Ing. Gerhard Hirt, Leiter des Instituts für Bildsame Formgebung der RWTH Aachen University, danke ich für die Übernahme des Prüfungsvorsitzes.

Für das geweckte Interesse am Graduiertenkolleg gilt mein Dank dessen Sprecher, Herrn Univ.-Prof. Dr. rer. nat. Dr. h.c. Wolfgang Thomas, Inhaber des Lehrstuhls für Logik und Theorie diskreter Systeme der RWTH Aachen University. Für die organisatorische Arbeit rund um das Graduiertenkolleg danke ich Frau Helen Bolke-Hermanns und Frau Silke Cormann.

Für viele fruchtbare Diskussionen möchte ich mich bei meinen Kollegen und Kolleginnen am Lehrstuhl und den Mitgliedern des Graduiertenkollegs bedanken. Besonders erwähnen möchte ich (in alphabetischer Reihenfolge) Herrn Lars Evertz, Herrn David Kampert, Herrn Florian Palm, Herrn Andreas Schüller und Herrn Constantin Wagner. Weiterhin danke ich Herrn Dr. techn. Alois Zoitl, Herrn Univ.-Prof. Dr.-Ing. habil. Leon Urbas und Herrn Julius Pfrommer für das frühe Feedback zu den Kernideen dieser Arbeit.

Frau Margarete Milescu-Huber und Frau Ursula Bey danke ich für die Unterstützung bei den diversen organisatorischen Tätigkeiten. Ebenso danke ich Frau Margarete Milescu-Huber und Frau Julia Botov für die Durchsicht dieser Arbeit. Des Weiteren möchte ich mich bei den studentischen Hilfskräften des Lehrstuhls bedanken.

Meiner Ehefrau Natalia und unseren Kindern Konstantin und Alexandra danke ich von ganzem Herzen für die ständige Unterstützung, Geduld und Motivation während der gesamten Promotion.

Schließlich möchte ich meiner Mutter Renate für die liebevolle Unterstützung bei jeder beruflichen und privaten Entscheidung danken.

Ilvesheim, im Mai 2017

Sten Grüner

“Simplicity is prerequisite for reliability.”
—Edsger W. Dijkstra, 1975

Inhaltsverzeichnis

Abkürzungsverzeichnis

ABK Anzeige- und Bedienkomponente

AdA Automatisierung der Automatisierung

API Application Programming Interface

BCET Best Case Execution Time

CFC Continuous Function Chart

CPPS Cyber-Physical Production Systems

CPS Cyber-Physical Systems

CPU Central Processing Unit

CTL Computational Tree Logic

DSL Domain-Specific Language

ECC Execution Control Chart

EDF Earliest Deadline First

ERP Enterprise Resource Planning

EWS Engineering Workstation

FBD Funktionsbaustein oder Funktionsbausteinsprache (engl. Function Block Diagram)

FBN Funktionsbausteinnetzwerk

GAMS General Algebraic Modeling System

HMI Human-Machine Interface

HTTP Hypertext Transfer Protocol

I/O Input-Output

IDE Integrated Development Environment

ILP Integer Linear Program

ISSC In-cycle Sequential State Chart

kgV kleinstes gemeinsames Vielfaches

LP Linear Program

MILP Mixed Integer Linear Program

MINLP Mixed Integer Nonlinearly Constrained Program

MIQP Mixed Integer Quadratic Program

OO Objektorientierung

OPC UA Open Platform Communications Unified Architecture

PLC Programmable Logic Controller (deutsch SPS)

PLS Prozessleitsystem

PNK prozessnahe Komponente

POU Program Organization Unit

QoC Quality of Control

R&I Fließschema Rohrleitungs- und Instrumentenfließschema

RA Resource-Aware oder ressourcenadaptiv

RTE Runtime Environment

SDK Software Development Kit

SFC Sequential Function Chart (deutsch Ablaufsprache)

SK selbstständige Komponente

SOA serviceorientierte Architektur

SPS speicherprogrammierbare Steuerung (engl. PLC)

SSC Sequential State Chart

ST Structured Text (deutsch strukturierter Text)

SysML Systems Modeling Language

TA Timed Automata

TCP Transmission Control Protocol

TCTL Timed Computation Tree Logic

UML Unified Modeling Language

WCET Worst Case Execution Time

XML-RPC Extensible Markup Language Remote Procedure Call

Kurzfassung

Die aktuellen Entwicklungen der Prozessleittechnik und der Automatisierungstechnik, die unter den Oberbegriffen „Industrie 4.0“ und „Cyber-Physical Production Systems“ subsumiert werden, fordern die Verlagerung bzw. die Bereitstellung zusätzlicher Funktionen auf die Prozessleitebene der Automatisierungspyramide. Diese Funktionen umfassen beispielsweise Self-X Funktionalitäten wie Selbstoptimierung und Selbstdiagnose einzelner Komponenten sowie die Bereitstellung zusätzlicher nicht-echtzeitrelevanter Daten wie die Beschreibung der Fähigkeiten und Merkmale des Systems. Diese zusätzlichen Funktionen machen den Unterschied zwischen herkömmlichen Systemen und smarten Industrie 4.0-Systemen aus.

Die Bereitstellung der zusätzlichen Funktionalität erfordert überplanmäßige Rechen- und Kommunikationsressourcen, was insbesondere im Hinblick auf die echtzeitkritischen Laufzeitumgebungen nichttrivial ist. Zum einen werden die verfügbaren Ressourcen einzelner Systeme bereits vollständig genutzt bzw. reserviert, zum anderen könnten die Betreiber den Aufwand der notwendigen Rekonfiguration des Systems unter anderem aus Gründen der langen Betreibs- und Lebenszyklen der Systeme scheuen.

Die Laufzeitsysteme der Prozessleitebene werden in den meisten Fällen in einem konstanten Zyklus betrieben, der dem zu kontrollierenden physischen System angepasst ist. Da die Ausführungszeit der anwenderspezifischen Logik gewissen Fluktuationen sowie Überabschätzungen in Bezug auf die maximale Laufzeit unterliegt, variiert die tatsächliche Ausführungszeit innerhalb des Zyklus. Die überschüssige Zeit, Slackzeit genannt, bleibt wegen der festen Zykluszeit häufig ungenutzt.

Die dynamische Anpassung der benötigten (Rechen-)Ressourcen ist eine Dimension der Flexibilität leittechnischer Anwendungen, die in der Domäne der Automatisierungstechnik bislang unbeachtet blieb. In den Bereichen der Echtzeitsysteme und des Schedulings existieren dagegen bereits Konzepte, die den Ausgangspunkt für diese Arbeit darstellen. Die Analyse dieser Ansätze, unter Berücksichtigung der aufgestellten spezifischen Anforderungen der Leittechnik, bildet die Grundlage dieser Dissertation.

Die Zielsetzung dieser Arbeit ist ein Rahmenwerk für die nahtlose Integration von ressourcenadaptiven Anwendungen in die zyklischen Laufzeitsysteme. Diese Anwendungsklasse kann für die Bereitstellung der zusätzlichen Funktionalität während der Slackzeit genutzt werden, ohne die Echtzeitanforderungen und den Funktionsumfang der existierenden Kernanwendung einzuschränken.

Ein Beitrag der Arbeit ist eine Softwarearchitektur, die die Koexistenz unterschiedlicher Ausführungsparadigmen innerhalb eines Laufzeitsystems ermöglicht. Die Paradigmen umfassen die tasklistengesteuerte Ausführung nach IEC 61131-3, die ereignisgesteuerte Ausführung nach IEC 61499 sowie die Einbettung weiterer Ausführungsvorschriften, wie z. B. der eingeführten ressourcenadaptiven Ausführung. Dieses ist durch die konsequente Kapselung der Daten und der Ausführungsvorschrift innerhalb der Komponenten sowie des Prinzips des hierarchischen Schedulings möglich.

Eine mögliche Ausführungsvorschrift wird durch das zusätzlich eingeführte Meta-Modell zur Beschreibung der Ausführungszeitsensitivität für Funktionsbaustein-Anwendungen innerhalb des Laufzeitsystems definiert. Dazu wird die Semantik der Prozedurbeschreibungssprache Sequential State Chart angepasst, um die Auswertung des Charts innerhalb eines Zyklus des Laufzeitsystems zu ermöglichen. Die Syntax der Sprache ist den meisten Nutzern bekannt, was positiv zu der Akzeptanz der Sprache beiträgt. Die Semantik der Prozedur wird formal mithilfe des UPPAAL-Toolkits modelliert, das neben der Eindeutigkeit auch zusätzliche Möglichkeiten für das Engineering, wie z. B. die formale Validierung und Simulation, eröffnet.

Anschließend wird eine Referenzarchitektur für den systemweiten Komponentenscheduler vorgestellt, der die Überwachung und die dynamische Zuteilung der Slackzeit an die einzelnen ressourcenadaptiven Komponenten sicherstellt. Für diesen Zweck wird eine Kombination aus offline und online Scheduling verwendet. Die Berechnung des offline Schedules beinhaltet das Lösen eines NP-harten Problems, das mithilfe eines gemischt-ganzzahligen linearen Programms und eines passenden Solvers aufgestellt bzw. gelöst wird. Die Kombination aus einem offline und online Verfahren ermöglicht die Ausführung der ressourcenadaptiven Anwendungen sowie weitere Möglichkeiten der Flexibilisierung des Schedulings, wie z. B. die Möglichkeit des dynamischen Austauschs des Schedules zur Laufzeit bei gleichzeitiger Sicherstellung der Echtzeitschranken.

Das eingeführte Rahmenwerk inklusive einer Engineering-Umgebung wurde als Erweiterung der quelloffenen Laufzeitumgebung ACPLT/RTE prototypisch implementiert. Der Mehrwert der ressourcenadaptiven Anwendungen für die Prozessleittechnik wird an mehreren Use-Cases demonstriert. Dazu zählen Anwendungen mit und ohne Zugriff in den operativen Betrieb des Laufzeitsystems. Zu der ersten Kategorie gehören die prozessbegleitende Simulation mit variabler Simulationsgenauigkeit und die mehrstufige Messwertvalidierung. In die zweite Kategorie fallen die nicht-echtzeitfähige Kommunikation mittels OPC UA und die Transaktionskontrolle für regelbasiertes Engineering im Rahmen der Automatisierung der Automatisierung.

Abstract

Resource-Aware Applications for Operative Process Control Engineering

Current developments in process control engineering and industrial automation that can be subsumed under the umbrella terms “Industrie 4.0” and “Cyber-Physical Production Systems”, require the provision of additional functionalities to the process control layer of the automatization pyramid. These functionalities include, for example, Self-X functionalities like self-optimization and self-diagnosis of single automation components as well as the provision of additional non real-time information like the description of system capabilities and attributes. These additional functionalities make all the difference between conventional and smart Industrie 4.0 production systems.

The deployment of these additional services requires supplementary computation- and communication-resources. Providing these resources is non-trivial due to the hard real-time requirements of industrial runtime environments. On one hand, the available resources may already be completely utilized or reserved. On the other hand, operators may shy away from the required costs of the system reconfiguration due to the long service- and life-cycles of the utilized equipment.

Industrial runtime systems are usually operated with a fixed cycle time that is fitted to the controlled physical system. The effectively utilized processing time of the whole system varies due to fluctuations in the actual execution times of the application-specific control logic as well as overestimations of its worst-case execution time. The unused processing time at the end of a cycle, the so called slack time, is usually not utilized by current runtime environments.

A dynamic adaptation of required (computational) resources is a dimension of flexibility of industrial automation applications that has not been focused upon in process control engineering research. However, some approaches for resource-awareness of applications exist in the research communities of real-time systems and scheduling theory. A review of these approaches constitutes a starting point for this dissertation. The analysis of the available approaches and frameworks has to be performed under the aspects of derived domain-specific functional and non-functional requirements.

The goal of this work is to develop a framework for a seamless integration of resource-aware applications into cyclic runtime environments. This class of industrial automation applications can be used for the provision of additional functionality during the slack time which per definition cannot violate the real-time requirements and the functionality of the existing runtime’s core-application.

A first contribution of this work is a software architecture which allows the coexistence of different execution control paradigms within one runtime environment. These paradigms comprise a task list-based execution according to IEC 61131-3, an event-based execution of IEC 61499 as well as embedding further execution control rules such as the introduced resource-aware execution mechanisms. This embedment is possible due to a rigorous

encapsulation of dataflow and execution control flows within a program organization unit and the utilization of mechanisms of hierarchical scheduling.

One possibility of realizing the resource-aware execution control is represented by the introduced meta-model for describing the execution-time sensitivity of function block applications inside the runtime environment. The meta-model is built upon a procedure description language called Sequential State Chars of which the semantics are adopted so that they can be evaluated during the cycle of the runtime system. The syntax of the language is familiar to most users in the industrial automation domain thus allowing a higher acceptance of the introduced framework. The semantics of the procedure description language is formalized by using a transformation to timed automata of the UPPAAL-toolkit. This transformation not only allows an unambiguous semantics of the introduced meta-model, but also adds additional possibilities for the engineering, e.g. formal validation and simulation of modelled procedures.

Subsequently, a reference architecture for a resource-aware system level component scheduler is introduced. This architecture allows the monitoring and dynamic assignment of slack time to single resource-aware components at runtime. The presented scheduler uses a combination of offline and online scheduling. The computation of an offline scheduling table requires solving an NP-hard problem. This task is accomplished by modelling the scheduling problem as a mixed-integer program and solving it with available solvers. The combination of offline and online scheduling techniques not only allows the execution of resource-aware applications, but also the use of additional features like the dynamic exchange of offline scheduling tables at runtime and the execution of sporadic tasks.

The introduced resource-aware framework and the appendant engineering environment were prototypically implemented as an extension of an open source industrial runtime environment ACPLT/RTE. The additional value of resource-aware applications is demonstrated in different use cases with and without operative process intervention. The first category includes the process accompanying simulation with variable simulation precision and multistage validation of measured values. The second category contains non real-time communication with OPC UA and transaction control that is used for rule based engineering systems in the domain of automation of automation.