VDI

REIHE 08
MESS-,
STEUERUNGS-
UND REGELUNGS-
TECHNIK

Fortschritt-Berichte VDI

Mahyar Azarmipour, M. Sc.,
Aachen

NR. 1275

Virtualisierung prozessnaher Steuerungen in der Prozessautomatisierung

BAND
1|1

VOLUME
1|1

Lehrstuhl für
Prozessleittechnik
der RWTH Aachen

Virtualisierung prozessnaher Steuerungen in der Prozessautomatisierung

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

Mahyar Azarmipour, M. Sc.

Berichter: Univ.-Prof. Dr.-Ing. Ulrich Epple
Univ.-Prof. Dr.-Ing. Stefan Kowalewski
Univ.-Prof. Dr.-Ing. Tobias Kleinert

Tag der mündlichen Prüfung: 25.01.2022

VDI

REIHE 08
MESS-,
STEUERUNGS-
UND REGELUNGS-
TECHNIK

Fortschritt-Berichte VDI

Mahyar Azarmipour, M. Sc.,
Aachen

NR. 1275

Virtualisierung prozessnaher Steuerungen in der Prozessautomatisierung

BAND
1|1

VOLUME
1|1

ACPLT
AACHENER
PROZESSLEITTECHNIK

Lehrstuhl für
Prozessleittechnik
der RWTH Aachen

VDI verlag

Azarmipour, Mahyar

Virtualisierung prozessnaher Steuerungen in der Prozessautomatisierung

Fortschritt-Berichte VDI, Reihe 08, Nr. 1275. Düsseldorf: VDI Verlag 2022.

126 Seiten, 62 Bilder, 4 Tabellen.

ISBN 978-3-18-527508-1, E-ISBN 978-3-18-627508-0, ISSN 0178-9546

48,00 EUR/VDI-Mitgliederpreis: 43,20 EUR

Für die Dokumentation: Virtualisierung – Speicherprogrammierbare Steuerung – Informationsdiode – Industrie 4.0 – Dynamisches Deployment – Automatisierungstechnik – IT/OT-Konvergenz

Keywords: Virtualization – Programmable logic controller – Information diode – Industry 4.0 – Dynamic deployment – Automation – IT/OT-Convergence

Die vorliegende Arbeit wendet sich an Ingenieur*innen und Wissenschaftler*innen aus der Prozessautomatisierung. Ziel dieser Arbeit ist ein Architekturentwurf für die Steuerungsgeräte der prozessnahen Komponenten, um diese mit einer höheren Vernetzung und Wandelbarkeit auszustatten. Die Architektur setzt Hypervisor-Virtualisierung ein, um eine Trennung der Anwendungen mit unterschiedlichen Anforderungen auf der gleichen Hardware zu ermöglichen. Die Anwendungen werden in vorkonfigurierten Partitionen gekapselt und betrieben. Um die Modularisierung der Anwendungen zu erhöhen, werden Container als zusätzliche Virtualisierungskomponenten eingesetzt. Für die Verwaltung der gesamten Komponentenhierarchie ist ein Verwaltungssystem vorgesehen, das die erforderlichen Dienste zur Komponentenverwaltung zur Verfügung stellt.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter www.dnb.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 www.dnb.de.

D82 (Diss. RWTH Aachen University, 2022)

ISBN 978-3-18-527508-1, E-ISBN 978-3-18-627508-0, ISSN 0178-9546

Vorwort

Die vorliegende Arbeit ist während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Prozessleittechnik der RWTH Aachen University entstanden. An dieser Stelle möchte ich mich herzlich für die Unterstützung und die ermöglichten Chancen bedanken.

An erster Stelle gilt mein großer Dank Herrn Prof. Dr.-Ing. Ulrich Epple für die Unterstützung meines Promotionsvorhabens. Die angenehme und konstruktive Arbeitsatmosphäre sowie der ausgezeichnete fachliche Austausch mit ihm haben zum erfolgreichen Abschluss meiner Arbeit beigetragen.

Bei Herrn Prof. Dr.-Ing. Stefan Kowalewski, Inhaber des Lehrstuhls Informatik 11 – Embedded Software an der RWTH Aachen University, möchte ich mich für die Übernahme der Rolle des Zweitgutachters bedanken.

Ich bedanke mich bei Herrn Prof. Dr.-Ing. Tobias Kleinert für seine Unterstützung und die Fachdiskussionen, welche ebenfalls zum Gelingen der Arbeit beigetragen haben.

Ich danke meinen Kollegen und den studentischen Hilfskräften für die gute Zusammenarbeit und die interessanten Gespräche. Besonders hervorheben möchte ich (in alphabetischer Reihenfolge) Haitham Elfaham, Julian Grothoff, Daniel Jakob, Lars Nothdurft und Christian von Trotha. Bei Frau Milescu bedanke ich mich für die gute Zusammenarbeit und organisatorische Hilfe.

Ein weiterer Dank gilt Afrooz Nazari für die Unterstützung und Motivation in den vergangenen Jahren.

Besonderer Dank gilt auch meinen Eltern und meiner Schwester für die Unterstützung während meiner gesamten Promotionszeit.

Warstein, im März 2022 Mahyar Azarmipour

Wissenschaft: Es ist nicht ihr Ziel, der unendlichen Weisheit eine Tür zu öffnen, sondern eine Grenze zu setzen dem unendlichen Irrtum.

Bertolt Brecht

Inhaltsverzeichnis

List of Symbols

ASIL Automotive Safety Integrity Level

AUTOSAR AUTomotive Open System ARchitecture

CFC Continuous Function Charts

CPC Core Process Control

CPU Central Processing Unit

CPS Cyber Physical Systems

SFC Sequential Function Chart

DT Digital Twin

ESE Einzelsteuereinheiten

ECU Electronic Control Units

ERP Enterprise Resource Planning

EAL Evaluation Assurance Levels

FS File System

FB Funktionsbaustein

FIFO First In First Out

GSE Gruppensteuereinheiten

HMI Human Machine Interface

I40 Industrie 4.0

IIC Industrial Internet Consortium

IoT Internet of Things

IP Internet Protocol

IT Informationstechnik

IACS Industrial Automation and Control System

IPC Industrie-PC

KAS komponentenbasierte Architektur für Automatisierungssysteme

LWIP Light Weight Internet Protocol

M+O Monitoring and Optimization

MMU Memory Management Unit

MES Manufacturing Execution System

NOA NAMUR Open Architecture

OT Operational Technology

OPC UA Open Platform Communications Unified Architecture

OO Object Oriented

OS Betriebssystem

POSIX Portable Operating System Interface

PNK Prozessnahe Komponente

QoS Quality of Service

RTOS Real Time Operating System

ROM Read Only Memory

SSC Sequential State Charts

SFC Sequential Function Charts

SAP Service Access Points

SIL Safety Integrity Level

SPS Speicherprogrammierbare Steuerung

SOA Service Oriented Architecture

SCADA Supervisory control and data acquisition

TCP Transmission Control Protocol

UDP User Datagram Protocol

VM Virtuelle Maschine

VMM Virtual Machine Monitor

VoR Verification of Request

Kurzfassung

Industrie 4.0 ist ein neues Paradigma, das eine zentrale Rolle in der Entwicklung der zukünftigen Automatisierungssysteme spielt. Die neue Generation der industriellen Automatisierungstechnik zielt auf die Erhöhung der Wandelbarkeit der Automatisierungssysteme ab. Dabei ist die Vernetzung und die Kooperation mit der IT-Welt eine wichtige Voraussetzung, um die angeforderte Wandelbarkeit zu erreichen. Daher müssen neue Architekturen und Lösungen eingesetzt werden, um eine Kooperation zwischen den Automatisierungssystemen und der IT zu realisieren. Ziel dieser Arbeit ist ein Architekturentwurf für die Steuerungsgeräte der prozessnahen Komponenten, um diese mit einer höheren Vernetzung und Wandelbarkeit auszustatten. Die Hauptanforderungen, welche von der Architektur erfüllt werden, sind:

- Der parallele Betrieb von Anwendungen unterschiedlicher Kritikalität
- Das dynamische Deployment von neuen Anwendungen zur Laufzeit
- Die Realisierung eines sicheren Gateways für die Kommunikation zwischen Systemen der Automatisierungsebene und übergeordneten IT-Systemen
- Die offene Kommunikation mit der IT-Welt
- Die Realisierung eines lokalen Software- und Zugriffsverwaltungssystems

Die vorgeschlagene Architektur besteht aus einem Mehrebenen-Komponentenmodell und wird als komponentenbasierte Architektur für Automatisierungssysteme (KAS) bezeichnet. Die unterste Ebene der KAS-Architektur ist die Ebene der Partitionen. Die KAS-Architektur setzt Hypervisor-Virtualisierung ein, um eine Trennung der Anwendungen mit unterschiedlichen Anforderungen auf der gleichen Hardware zu ermöglichen. Die Anwendungen werden in vorkonfigurierten Partitionen gekapselt und betrieben. Um die Modularisierung der Anwendungen zu erhöhen werden Container als zusätzliche Virtualisierungskomponenten eingesetzt. Containertechnologien ermöglichen die Kapselung und Verwaltung der Anwendungen in unterschiedlichen Containern innerhalb einer Partition. Dadurch können beispielsweise unterschiedliche Versionen der Anwendungen in einer Partition verwaltet werden. Die Container bilden die zweite Komponentenebene in der KAS-Architektur. Die letzte Komponentenebene stellt die Kapselung in die Funktionsbausteine dar. Für die Verwaltung der gesamten Komponentenhierarchie ist in der KAS-Architektur ein Verwaltungssystem vorgesehen, das die erforderlichen Dienste zur Komponentenverwaltung zur Verfügung stellt. Das Verwaltungssystem ist eine Systemfunktionalität der KAS-Architektur und in einer eigenen Partition gekapselt. Eine weitere Systemfunktion der KAS-Architektur ist das Interface. Dieses wird ebenfalls in einer eigenen Partition gekapselt. Diese Interface-Partition ist die einzige Partition, die mit externen Komponenten außerhalb der Kernautomatisierung kommunizieren darf. In der Arbeit werden für die Validierung der KAS-Architektur beispielhaft Anwendungspartitionen für die Prozessführung

und die Simulation entwickelt. Mit Hilfe dieser Anwendungen können realistische Szenarien der Automatisierungsebene prototypisch implementiert und getestet werden. Die Ergebnisse zeigen, dass die KAS-Architektur eine leistungsfähige und übersichtlich verwaltbare Systemumgebung darstellt, um für neue Anforderungen eine hohe Flexibilität zu bieten, sowie der durchgängigen Interoperabilität der Automatisierungsebene zu genügen, ohne die Integrität der Kernautomation zu gefährden.

Abstract

Industry 4.0 is a new paradigm that plays a central role in the development of future automation systems. The new generation of industrial automation aims to increase the agility of the automation system. In this context, cooperation with the IT world is an important prerequisite to achieve the requested agility. Therefore, new architectures and solutions have to be developed to realize a cooperation between the automation systems and the IT. The goal of this work is an architecture design for the control devices in order to provide them with a higher level of connectivity and agility. The main features, which are fulfilled by the architecture, are:

- The parallel operation of applications of different criticality
- The dynamic deployment of new applications at runtime
- The realization of a secure gateway for communication between automation level systems and higher level IT systems
- The open communication with the IT world
- The realization of a local software and access management system

The proposed architecture consists of a multi-level component model and is referred to as a component-based architecture for automation systems (KAS). The lowest level of the KAS architecture is the partition level. The KAS architecture employs hypervisor virtualization to enable separation of applications with different requirements on the same hardware. Applications are encapsulated in preconfigured partitions. Containers are used as additional virtualization components to increase modularization of applications. Container technologies enable the encapsulation and management of applications in different containers within a partition. This means, for example, that different versions of the applications can be managed in one partition. The containers are the second component level in the KAS architecture. The last component level represents the encapsulation in the function blocks. For the management of the entire component hierarchy in the KAS architecture a management system is developed. The management system is a system functionality of the KAS architecture and is encapsulated in its own partition. Another system function of the KAS architecture is the interface. This is also encapsulated in its own partition. The interface partition is the only partition that is allowed to communicate with outside of the core automation domain. In this work, application partitions for process control and simulation are developed as examples for the validation of the KAS architecture. These applications can be used for a prototype implementation of automation level scenarios. The results show that the KAS architecture provides a powerful and clearly manageable system environment to meet the new requirements for agility as well as continuous interoperability of the automation level without compromising the integrity of the core automation.