

# Fortschritt-Berichte VDI

VDI

Reihe 8

Mess-,  
Steuerungs- und  
Regelungstechnik

Nr. 1254

Dipl.-Ing. Andreas Schüller,  
Aachen

## Ein Referenzmodell zur Beschreibung allgemeiner Prozeduren im leittechni- schen Umfeld

**ACPLT**  
AACHENER  
PROZESSLEITTECHNIK

Lehrstuhl für  
Prozessleittechnik  
der RWTH Aachen



# **Ein Referenzmodell zur Beschreibung allgemeiner Prozeduren im leittechnischen Umfeld**

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.-Ing. Andreas Heinrich Schüller**

aus Aachen

Berichter: Univ.-Prof. Dr.-Ing. Ulrich Epple  
Univ.-Prof. Dr.-Ing. Alexander Fay

Tag der mündlichen Prüfung: 09.11.2016



# Fortschritt-Berichte VDI

Reihe 8

Mess-, Steuerungs-  
und Regelungstechnik

Dipl.-Ing. Andreas Schüller,  
Aachen

Nr. 1254

Ein Referenzmodell zur  
Beschreibung allgemeiner  
Prozeduren im leittech-  
nischen Umfeld



Lehrstuhl für  
Prozessleittechnik  
der RWTH Aachen

Schüller, Andreas

## **Ein Referenzmodell zur Beschreibung allgemeiner Prozeduren im leitechnischen Umfeld**

Fortschr.-Ber. VDI Reihe 8 Nr. 1254. Düsseldorf: VDI Verlag 2016.

162 Seiten, 68 Bilder, 19 Tabellen.

ISBN 978-3-18-525308-6, ISSN 0178-9546,

€ 62,00/VDI-Mitgliederpreis € 55,80.

**Für die Dokumentation:** Prozessleitechnik – Prozedur – Modellierung – Ablaufbeschreibung – Referenzmodell – Beschreibungssprache – Dienste – Interaktion

Die vorliegende Arbeit wendet sich an Ingenieure und Wissenschaftler im Bereich der Automatisierungstechnik. Sie befasst sich mit der Erstellung eines Referenzmodells zur Prozedurbeschreibung. Das Modell baut auf einer umfangreichen Analyse verschiedener bereits existierender Prozedurbeschreibungssprachen auf. Das Referenzmodell bildet eine Abstraktionsebene, die den gemeinsamen Kern der verschiedenen Beschreibungssprachen enthält. Diese Abstraktionsebene ermöglicht eine semantische Verknüpfung von Steuerungsprozeduren für technische Prozesse und Geschäftsprozesse. Die Funktionen des Steuernden und die des Ausführenden können im Referenzmodell sowohl durch Menschen als auch durch Maschinen übernommen werden. Umgesetzt wird dies durch die Verwendung eines Dienstsysteams, das eine strikte Trennung zwischen den steuernden und den ausführenden Akteuren ermöglicht.

### **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>.

D 82 (Diss. RWTH Aachen University, 2016)

© VDI Verlag GmbH · Düsseldorf 2016

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9546

ISBN 978-3-18-525408-6

---

# Vorwort

Diese Dissertation habe ich während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Prozesseittechnik an der RWTH Aachen University erstellt. Während dieser Zeit konnte ich auf eine große Unterstützung durch das gesamte Team des Lehrstuhls bauen, für die ich mich an dieser Stelle bedanken möchte.

Zunächst gilt mein besonderer Dank Herrn Professor Dr.-Ing. Ulrich Epple. Die angenehme Arbeitsatmosphäre an seinem Lehrstuhl, viele intensive fachliche Diskussionen mit ihm und persönliche Gespräche waren die Basis zum Gelingen dieser Arbeit. Durch die vielfältigen Aufgaben am Lehrstuhl und die Möglichkeit der Teilnahme an Fachausschüssen und Arbeitskreisen hat er es mir ermöglicht den Blick auch über den Tellerrand meines Dissertationsthemas hinweg richten zu können.

Herr Professor Dr.-Ing. Alexander Fay, Inhaber der Professur für Automatisierungstechnik an der Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, danke ich für die Übernahme der Zweitbegutachtung. Seine konstruktiven Kommentare haben mir eine weitere Sicht auf das Thema gegeben. Bei Herrn Univ.-Prof. Dr.-Ing. Karl Nienhaus bedanke ich mich für die Übernahme des Prüfungsvorsitzes.

Ein großer Dank gilt auch meinen Kollegen am Lehrstuhl. Die intensiven Diskussionen mit ihnen haben mir oft geholfen, neue Lösungswege zu erkennen. Erwähnen möchte ich an dieser Stelle Herrn Dipl.-Inform. Markus Schlüter, der mich als langjähriger Betreuer während meiner Zeit als studentische Hilfskraft an das Thema Prozesseittechnik herangeführt hat. Ein Dank gilt auch meinen ehemaligen studentischen Hilfskräften und den Studierenden, die bei mir ihre Abschlussarbeit geschrieben haben.

Bedanken möchte ich mich auch bei den Mitgliedern des NAMUR Arbeitskreises 1.11 „Referenzmodelle der Leittechnik“. Im Rahmen des Arbeitskreises durfte ich meine Arbeiten zur Prozedurbeschreibung vorstellen. Die konstruktiven Diskussionen haben dafür gesorgt, dass meine Arbeit einen Bezug zur Praxis erhalten hat.

Herzlich bedanken möchte ich mich auch bei Frau Ursula Bey und bei Frau Margarete Milescu-Huber, die mir bei der Bewältigung der Hürden in der Hochschulbürokratie geholfen haben und mir während der Zeit am Lehrstuhl ebenso wie Frau Martina Uecker jederzeit mit Rat und Tat zur Seite standen. Für die abschließende Durchsicht möchte ich mich besonders bei Frau Bey und bei Frau Simone Claßen bedanken.

Auch meiner Verlobten Frau Dr.-Ing. Nadin Schlegel möchte ich ein großes Danke schön für ihre Unterstützung und ihre Geduld sagen. Schließlich möchte ich mich bei meiner Familie bedanken, insbesondere bei meinen Eltern Mathias und Anita Schüller, die mich zu diesem beruflichen und privaten Weg geführt haben. Danke!

Aachen, im November 2016

*Andreas Schüller*

---

*Nichts befähigt die Wissenschaft so wie der Schwatz mit Kollegen auf dem Flur.*  
Arno Penzias (\*1933), amerik. Physiker.

# Inhaltsverzeichnis

<b>Akronyme</b>	VII
<b>Kurzfassung</b>	XI
<b>Abstract</b>	XIII
<b>1 Einleitung</b>	1
1.1 Motivation . . . . .	1
1.2 Zielsetzung . . . . .	4
1.3 Gliederung . . . . .	5
<b>2 Definitionen und grundlegende Begriffswelt</b>	6
2.1 Prozess . . . . .	6
2.2 Das kybernetische Grundprinzip . . . . .	7
2.2.1 Produktsystem . . . . .	8
2.2.2 Anlagensystem . . . . .	8
2.2.3 Steuerungssystem . . . . .	10
2.3 Steuerungsfunktionen . . . . .	11
2.3.1 Die Prozedur als Steuerungsfunktion . . . . .	12
2.4 Funktionaler Leitsystemaufbau . . . . .	14
2.5 Kommunikation . . . . .	16
2.5.1 Dienstbasierte Kommunikation . . . . .	16
2.6 Prozedurbeschreibungsmittel . . . . .	18
2.6.1 Formale Sprachen . . . . .	19
2.6.2 Modelle . . . . .	20
<b>3 Analyse von Prozedurbeschreibungssprachen</b>	24
3.1 Elemente einer Prozedurbeschreibungssprache . . . . .	24
3.1.1 Aufbaumodell . . . . .	25
3.1.2 Hierarchie- und Vernetzungsmodell . . . . .	25
3.1.3 Abstraktions- und Zuordnungsmodell . . . . .	26
3.1.4 Aktions- und Aktivitätenmodell . . . . .	26
3.1.5 Ausführungssteuerungsmodell . . . . .	27
3.2 Auswahl der Prozedurbeschreibungssprachen . . . . .	27
3.3 Prozedurbeschreibungssprachen zur Steuerung von technischen Prozessen . . . . .	27
3.3.1 Endliche Automaten . . . . .	28
3.3.2 Petrinetze . . . . .	31
3.3.3 Grafacet . . . . .	34

3.3.4	Procedural Function Charts . . . . .	36
3.3.5	Grafchart . . . . .	38
3.3.6	Sequential Function Charts . . . . .	42
3.3.7	Zustandsdiagramme . . . . .	46
3.3.8	PLC Statecharts . . . . .	49
3.3.9	Sequential State Charts . . . . .	51
3.4	Prozedurbeschreibungssprachen zur Steuerung von Geschäftsprozessen . . . . .	54
3.4.1	Aktivitätsdiagramme . . . . .	54
3.4.2	Business Process Execution Language . . . . .	57
3.4.3	Business Process Model and Notation . . . . .	58
3.4.4	Koordination, Kooperation und Kommunikation . . . . .	61
3.4.5	Ereignisgesteuerte Prozessketten . . . . .	62
3.5	Vergleich der analysierten Sprachen . . . . .	64
<b>4</b>	<b>Referenzmodell</b>	<b>67</b>
4.1	Anforderungen an das Referenzmodell . . . . .	67
4.2	Modellbeschreibung . . . . .	68
4.2.1	Aufbaumodell . . . . .	69
4.2.2	Hierarchie- und Vernetzungsmodell . . . . .	70
4.2.3	Aktions- und Aktivitätenmodell . . . . .	74
4.2.4	Abstraktions- und Zuordnungsmodell . . . . .	76
4.2.5	Ausführungssteuerungsmodell . . . . .	78
4.3	Darstellungsformen des Referenzmodells . . . . .	82
4.3.1	Visualisierung . . . . .	82
4.3.2	XML-Darstellung . . . . .	84
4.4	Anforderungen an das Kommunikationssystem und an die Ausführungseinheiten . . . . .	85
4.5	Prototypische Implementierung . . . . .	87
<b>5</b>	<b>Anwendung des Referenzmodells</b>	<b>88</b>
5.1	Entwurfsprozess einer Prozedur . . . . .	88
5.1.1	Flexible Strukturen . . . . .	90
5.2	Steuerungsprozedur einer Pumpe . . . . .	93
5.2.1	Verwendung von Rollen . . . . .	97
5.3	Integration von Menschen in die Prozedurausführung . . . . .	97
5.3.1	Assistenzsysteme . . . . .	99
<b>6</b>	<b>Zusammenfassung und Diskussion</b>	<b>100</b>
6.1	Zusammenfassung . . . . .	100
6.2	Diskussion . . . . .	102
<b>A</b>	<b>Zusammenfassung der Analyse</b>	<b>105</b>
A.1	Aufbaumodell . . . . .	105
A.2	Hierarchie- und Vernetzungsmodell . . . . .	106
A.3	Abstraktions- und Zuordnungsmodell . . . . .	108
A.4	Aktions- und Aktivitätenmodell . . . . .	109
A.5	Ausführungssteuerungsmodell . . . . .	111

<b>B Prozeduren der Akteure bei einer Ventilwartung</b>	<b>114</b>
<b>C Kompatibilität zu bestehenden Beschreibungssprachen</b>	<b>119</b>
C.1 Umschreibung der Aktionsbestimmungszeichen . . . . .	119
C.2 Priorität der Alternativverzweigungen . . . . .	121
C.3 Modellierung von Do und Exit . . . . .	122
C.4 Verarbeitung von Messwerten und Setzen von Stellgrößen . . . . .	123
<b>D XML-Repräsentation des Referenzmodells</b>	<b>125</b>
<b>Literaturverzeichnis</b>	<b>134</b>

---

# Akronyme

ABK	Anzeige-/Bedienkomponente
AD	Aktivitätsdiagramme
AP	Automatisierungspyramide
ARIS	Architektur integrierter Informationssysteme
AWL	Anweisungsliste
Blob	Binary Large Object
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language (BPEL) for Web Services
BPMN	Business Process Modeling Notation
CAEX	Computer Aided Engineering Exchange
CPS	Cyber-Physical Systems
DEA	Deterministische Endliche Automaten
DPWS	Device Profiles for Web Services
EA	Endliche Automaten
EPK	Ereignisgesteuerte Prozessketten
ERP	Enterprise Resource Planning
ES	Engineering Station
FBS	Funktionsbaustein-Sprache
FDI	Field Device Integration
HMI	Human-Machine-Interface
IEC	International Electrotechnical Commission
ISA	International Society of Automation
ISO	International Organization for Standardization
K3	Koordination, Kooperation und Kommunikation
KOP	Kontaktplan
LIMS	Labor-Informations- und Managementsystem
MES	Manufacturing Execution System
MSR	Mess-, Steuerungs- und Regelungstechnik

NAMUR	Interessengemeinschaft Automatisierungstechnik der Prozessindustrie
NEA	Nichtdeterministische Endliche Automaten
OASIS	Organization for the Advancement of Structured Information Standards
OCEB	OMG Certified Expert in Business Process Management
OMG	Object Modeling Group
OPC UA	Open Platform Communications Unified Architecture
OS	Operator Station
OWL	Web Ontology Language
PCS	Process Control System
PFC	Procedural Function Charts
PID	Proportional-Integral-Differential
PIMS	Prozessdaten-Informations- und Managementsystem
PLC	Programmable Logic Controller
PLC SC	Programmable Logic Controller (PLC) Statecharts
plcML	PLC-Modeling Language
PLS	Prozessleitsystem
PLT	Prozessleittechnik
PN	Petrinetze
PNK	Prozessnahe Komponente
POE	Programm-Organisationseinheit
QoS	Quality of Service
R&I	Rohrleitungs- und Instrumentierungsdiagramm
RAMI	Referenzarchitekturmodell Industrie 4.0
RIO	Remote Input/Output
SC	Statecharts
SFC	Sequential Function Charts
SIPN	Steuerungstechnisch Interpretierbare Petrinetze
SOA	Service-oriented Architecture
SPS	Speicherprogrammierbare Steuerung
SSC	Sequential State Charts
ST	Strukturierter Text
SysML	Systems Modeling Language
UML	Unified Modeling Language
URI	Uniform Resource Identifier
WS-BPEL	Web Service BPEL

WSDL	Web Service Description Language
XML	Extensible Markup Language
XSD	Extensible Markup Language (XML) Schema Definition
XSLT	Extensible Stylesheet Language Transformation

---

# Kurzfassung

Prozeduren nehmen in der Automatisierung einen immer größeren Stellenwert ein. Prozeduren zum Start eines Motors werden ebenso wie Abläufe zur Steuerung von Geschäftsprozessen seit vielen Jahren betrachtet und in domänenspezifischen Sprachen beschrieben. Viele dieser domänenspezifischen Sprachen sind in der Praxis erprobt. Sie ähneln sich, besitzen jedoch keine gemeinsame Syntax und Semantik. Dies ist vor allem den unterschiedlichen Anforderungen der Domänen geschuldet. Während die Steuerungsprozedur einer Maschine formal definiert sein muss, damit die Steuerung sie eindeutig ausführen kann, liegt der Fokus einer Instandhaltungsprozedur auf einer Darstellung, die durch den menschlichen Akteur einfach verstanden und umgesetzt werden kann. Im Gegensatz zu Maschinen können Menschen auch informale Aufrufe interpretieren. Die bestehenden Prozedurbeschreibungssprachen werden in einer umfassenden Analyse betrachtet.

Die Vielfalt der domänenspezifischen Sprachen hat kein Problem dargestellt, solange die beschriebenen Prozeduren unabhängig voneinander betrachtet werden. Heutzutage wird allerdings unter den Schlagworten „horizontale Integration“ und „vertikale Integration“ eine ganzheitliche Betrachtung von Prozeduren angestrebt. Sowohl die horizontale als auch die vertikale Integration sind essentielle Bestandteile im Zukunftsprojekt Industrie 4.0. Zum einen sollen hierbei die Prozeduren zwischen den einzelnen Domänen ausgetauscht werden können, ohne dass jede Domäne die Sprachen der anderen Domänen verstehen muss. Zum anderen ist ein Zugriff auf die Informationen über den Prozedurzustand und die Beeinflussung von Prozeduren über die Ebenen der Automatisierungspyramide hinweg von Nöten.

In dieser Arbeit wird ein Referenzmodell zur Prozedurbeschreibung erarbeitet, das den gemeinsamen Kern der domänenspezifischen Sprachen beschreibt: Sie bestehen aus zwei Typen von Elementen, einer dieser Typen wirkt aktiv auf die Umgebung ein, der andere Typ reagiert auf Änderungen der Umgebung. Des Weiteren gibt es gemeinsame Konstrukte zur Modellierung von Hierarchien, Alternativverzweigungen und Nebenläufigkeiten. Die Interaktion mit der Umgebung ist einer der Hauptunterschiede zwischen den domänenspezifischen Sprachen. Damit auch in dieser Hinsicht ein universelles Modell erzeugt werden kann, muss die Interaktion mit der Umgebung über Dienstaufrufe und Zustandsabfragen erfolgen. Dies deckt das Setzen und Abfragen von Variablen, aber auch die informale Übermittlung eines Arbeitsauftrags des Chefs an seinen Mitarbeiter ab. Somit ist sichergestellt, dass das Prozedurmodell auf alle Komplexitätsgrade angewendet werden kann. Das Referenzmodell ist unabhängig von einer konkreten Visualisierung. Dies bietet den Vorteil, dass jeder Nutzer der Prozedurbeschreibung eine für ihn persönlich optimierte Darstellung auswählen kann, ohne dass eine Modifikation der Prozedurbeschreibung ist. Auch eine Kopplung mit Assistenzsystemen ist möglich.

Neben der Beschreibung der Prozedur wird ein Konzept benötigt, das die einheitliche und eindeutige Ausführung der Prozedur zulässt. Dies beinhaltet die operative Ausführung der Schrittfolge im Regelfall, aber auch ein eindeutig definiertes Verhalten im Fehlerfall. Nur auf diese Weise ist eine Übertragbarkeit der Prozedur zwischen verschiedenen Systemen

gewährleistet.

Die Verwendung von Dienstaufrufen stellt ein Werkzeug zur Erhöhung der Flexibilität einer Prozedurbeschreibung dar. Die Abkehr von fest projektierten Signalverbindungen bewirkt, dass eine Zuordnung von benötigten Ausführungseinheiten erst zur Laufzeit erfolgen kann. Hier werden die kognitiven Fähigkeiten des Menschen ausgenutzt, die es ihm ermöglichen situationsbedingte Entscheidungen zu treffen. Ein Rollenkonzept unterstützt die flexible Zuordnung. Auf diese Weise kann der Entwickler der Prozedur in den Rollen Anforderungen an die Ausführungseinheiten definieren. Basierend auf den Rollen ist es realisierbar zur Laufzeit eine Zuordnung zu konkreten Ausführungseinheiten vorzunehmen.

---

# Abstract

Procedures assume an increasingly important role within current automation technology. Procedures for starting an engine as well as sequences for controlling business processes have been examined and subsequently described in domain-specific languages for many years now. Many of these domain-specific languages are tried and tested in practice. Although the languages are fairly similar, they do not share a common syntax and semantics. This fact can be explained by the different requirements of the individual domains. While the control procedures of a machine must be formally described in order that the control system can execute them unequivocally. The focus of a maintenance procedure is firmly placed on a representation that can be easily understood and implemented by a human being. In contrast to machines, human beings can also interpret informal requests. The existing procedure description languages will be considered in a comprehensive analysis.

The variety of the domain-specific languages has never presented a problem as long as the described procedures were considered independently from each other. Nowadays, however, a holistic approach towards procedures is pursued under the headings of “horizontal integration” and “vertical integration”. Both, the horizontal- and the vertical integration are essential components of the future-oriented project “Industrie 4.0”. On the one hand, it is envisaged that procedures can be exchanged between individual domains without the necessity that each domain can understand the language of the other domain. On the other hand, it is required that information regarding a procedure’s state can be accessed, and that procedures can be influenced by the different levels of the automation pyramid.

In this work, a reference model for procedure descriptions is developed that describes the common core of the different domain-specific languages. Basically, the languages consist of two types of elements. One of these element types actively influences the environment, while the other type reacts to changes in the environment. Additionally, there are shared constructs for the modelling of hierarchies, alternative branches and concurrencies. The interaction with the environment is one of the central differences between the domain-specific languages. In order to also create a universal model in this respect, the interaction with the environment can only be realized via service requests and status inquiries. This covers the setting and querying of variables as well as the informal submission of a task by the department head to his employees. This ensures that the procedure model can be applied to all levels of complexity. Moreover, the procedure model is independent of a concrete visualization. It is advantageous that each user of the procedure description can thus select a personally optimized representation without the need of having to modify the actual procedure description. A coupling with assistance systems is also possible.

Along with the description of the procedure, a concept that facilitates a consistent and unambiguous execution of the procedure is required. This includes the operative execution of the step chain during normal operations, but also a clearly-defined behavior in case of faults. This is the only way in which the portability of procedures between different systems can be ensured.

The use of service invocations can be understood as a tool to increase the flexibility of

the procedure descriptions. The renunciation of pre-configured signal connections has the effect that the allocation of the required execution units cannot occur until runtime. In this context, the cognitive abilities of human beings that enable us to make situation-based decisions can be fully exploited. The flexible allocation is additionally supported by a roles concept. By this means, the developer of a procedure can define the requirements of an execution unit in roles. Based on these roles, an allocation of the concrete execution units can be performed at runtime.