

Klaus Bender (Hrsg.)

**DEKOS-Engineering-Modell für intelligente  
Automatisierungskomponenten**

Klaus Bender, Rolf Birkhofer, Markus Bregulla,  
Florian Kuttig, Heiko Meyer

Herausgegeben von

**Univ.-Prof. Dr.-Ing. Klaus Bender  
Technische Universität München**

in der Reihe

**Informationstechnik im Maschinenwesen**



Herbert Utz Verlag · Wissenschaft  
München 2002

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist  
bei Der Deutschen Bibliothek erhältlich

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf photomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben – auch bei nur auszugsweiser Verwendung – vorbehalten.

Copyright © Herbert Utz Verlag GmbH 2002

ISBN 3-8316-0169-0

Printed in Germany

Herbert Utz Verlag GmbH, München

Tel.: 089/277791-00 – Fax: 089/277791-01

## **Vorwort**

Die hier vorgestellten Ergebnisse beruhen im wesentlichen auf den Ergebnissen des vom Bundesministerium für Bildung und Forschung (bmb+f) geförderten Verbundprojekt DeKOS (*Dezentrale Kooperierende Offene MikroSysteme*, Förderkennzeichen 16SV894/1). Der Lehrstuhl für Informationstechnik im Maschinenwesen (itm) wurde 1998 mit der Leitung des Projektes betraut.

Technische Innovationen der letzten Jahre wie beispielsweise die FDT-Technologie (Field Device Tool) haben unmittelbaren Einfluss auf den Projektverlauf genommen. Einer notwendigen Projektverlängerung bis 2002 wurde mit großer Mehrheit durch den Projektträger und der Verbundpartner zugestimmt. So wurde es ermöglicht, aktuelle Trends und Technologien bei den diversen prototypischen Realisierungen zu berücksichtigen. Für den weiteren industriellen Nutzen der Ergebnisse war dies von wesentlicher Bedeutung.

Aufgrund der Komplexität und Aktualität des Projektes konnten im Rahmen dessen am itm mehrere hoch interessante wissenschaftliche Arbeiten entstehen. Das Ziel des vorliegenden Buches besteht unter anderem darin, über dem eines Abschlussberichtes hinaus, detaillierte Einblicke in die wissenschaftlichen Modelle und Methodiken zu geben, die Grundlage für die zahlreichen prototypischen Implementierungen im Projekt waren.

Gerade die vielen prototypischen Realisierungen und Softwaretools die während des DeKOS-Projektes entstanden sind, waren nur machbar durch die tatkräftige Unterstützung interessierter Studenten die sowohl in Form von diversen Diplom- und Semesterarbeiten, als auch über längeren Zeitraum als wissenschaftliche Hilfskraft zum Erreichen der Projektziele nicht unerheblich beigetragen haben. An dieser Stelle möchten wir uns für das außerordentlich große Engagement bei der Verwirklichung der Ideen und Konzepte bei allen beteiligten Studenten bedanken.

Die projektbegleitenden Veröffentlichungen in Fachzeitschriften und auf Kongressen hatten in der Fachwelt zu großer Resonanz und Nachfrage geführt. Dies hat die beteiligten Mitarbeiter ermutigt, die Projektergebnisse in der vorliegenden Buchreihe einer breiteren Öffentlichkeit zur Verfügung zu stellen.

Mein besonderer Dank gebührt noch den genannten Autoren sowie allen Mitarbeitern des Lehrstuhls itm der Technischen Universität München, die direkt oder auch indirekt zur erfolgreichen Realisierung des Projektes beigetragen haben.



## **Inhaltsverzeichnis**

1	Einleitung .....	7
2	Problematik der neuen Geschlossenheit.....	11
3	Das Verbundprojekt DeKOS .....	15
3.1	Projektstruktur und –verlauf .....	15
3.2	Projektpartner.....	15
3.3	Aufgaben des Verbundprojektes .....	16
3.3.1	Gesamtkonzept .....	16
3.3.2	Arbeitspakete.....	17
3.4	Phasenplan .....	25
3.5	Erreichte Ziele und Ergebnisse .....	26
3.5.1	Spezifikationsschrift des DeKOS-Bedienmodells .....	26
3.5.2	Bibliothek der DeKOS-Funktionen .....	26
3.5.3	Tool zur Erzeugung von DeKOS-konformen Gerätebeschreibungen ..	27
3.5.4	Demonstrationsanlagen zur Validierung des DeKOS-Bedienmodells und Verbreitung der Projektergebnisse.....	28
3.6	Zusammenfassung und Ausblick .....	33
3.6.1	Zusammenfassung.....	33
3.6.2	Ausblick .....	34
4	Vereinheitlichung der Bedienfunktionalität .....	37
4.1	Anforderungsanalyse .....	38
4.2	Modellbildung.....	42
4.3	Bedienmodell .....	47
4.4	Einheitliche, wiederverwendbare Funktionen .....	54
4.5	Ordnungssystematik für die Gerätefunktionen .....	57
4.5.1	Automatisierungsfunktion.....	58
4.5.2	Geräteunabhängigkeit .....	61
4.5.3	Zusammenfassung.....	63
4.6	Atomare Bedienfunktionen.....	65
4.6.1	Setzen und Lesen des Messbereichsanfangswerts .....	66
4.6.2	Alarminformationen lesen.....	67
4.6.3	Bestellangaben lesen .....	69
4.6.4	Übertragungsverhalten setzen .....	72
4.6.5	Einheit setzen .....	73
4.7	Komfortfunktionen .....	74
4.8	Bedienobjekte.....	77
4.9	Zusammenfassung.....	81

5	Formale Beschreibung der DeKOS-konformen Gerätefunktionalität .....	83
5.1	Festlegung des Modellierungskonzeptes für die DeKOS-Referenzarchitektur (DRA).....	83
5.2	Klassenmodell der DRA .....	88
5.2.1	Bedienfunktionen .....	88
5.2.2	Gesamtgerät.....	90
5.2.3	Generische Parameter.....	91
5.2.4	Funktionsabläufe .....	91
5.3	Real Parameter Erweiterung (RPE) .....	93
5.4	Schnittstelle .....	93
5.5	Analyse der Eigenschaften realer Parameter .....	94
5.5.1	Analyse der Electronic Device Description Language (EDDL).....	96
5.5.2	Profibus-GSD .....	100
5.5.3	CANOpen-EDS .....	102
5.6	Modellierungskonzept der RPE .....	104
5.7	Kopplung der RPE mit der DRA .....	107
5.8	Formale Beschreibung der Architekturelemente .....	109
5.9	Anforderungen an die formale Beschreibung .....	109
5.9.1	Spezielle Anforderungen.....	109
5.9.2	Allgemeine Anforderungen.....	110
5.9.3	Diskussion der Anforderungen .....	112
5.10	Die deklarative Metasprache XML.....	113
5.10.1	Grundelemente der XML.....	114
5.10.2	Erfüllung der speziellen Anforderungen .....	116
5.10.3	Erfüllung der allgemeinen Anforderungen.....	120
5.10.4	Zusammenfassung und weitere Vorgehensweise.....	123
5.11	XML Schema für das Partialmodell DRA.....	123
5.11.1	Namensraum-Deklarationen .....	123
5.11.2	Abbildung der DRA-Klassen.....	125
5.11.3	Festlegung der Namen .....	127
5.11.4	Schnittstelle zur RPE .....	128
5.11.5	Wurzelement .....	129
5.12	XML Schema für das Partialmodell RPE .....	130
5.12.1	Namensraum-Deklarationen .....	131
5.12.2	Klassen und Typen der RPE.....	132
5.12.3	Datentypen von ParamObject .....	133
5.12.4	Wurzelement .....	135
6	Softwarekomponenten für die Gerätebedienung.....	137
6.1	JINI-Technologie .....	137
6.2	FDT-Technologie .....	139
6.3	Architektur des DeKOS-DTMs .....	140

---

6.4	DeKOS-Prototypen .....	143
6.4.1	Stellantrieb, Firma AUMA Werner Riester .....	143
6.4.2	Industriewaage, Firma Bizerba .....	144
6.4.3	Magnetisch-Induktiver Durchflussmesser, Firma E&H .....	146
6.4.4	Positioniersteuerung, Firma Kuhnke.....	149
6.4.5	Bilderkennungssystem, Firma Siemens AG .....	152
6.4.6	Druckmessumformer, Firma WIKA .....	154
6.4.7	Positioniersteuerung, Firma SEW-Eurodrive .....	155
7	Verwertbarkeit und industrielle Einführung der Projektergebnisse .....	161
7.1	Nutzen und Verwertbarkeit .....	161
7.1.1	Zeiteinsparung .....	161
7.1.2	Qualitätssteigerung .....	162
7.1.3	Komplexitätsbeherrschung.....	163
7.1.4	Kostenreduzierung .....	163
7.1.5	Verbesserung der Wettbewerbssituation.....	164
7.2	Industrielle Einführung .....	166
7.2.1	Pepperl+Fuchs.....	166
7.2.2	Bosch-Rexroth.....	168
7.2.3	PROFIDrive-DTM-Konsortium.....	169
7.2.4	Arbeitskreis PNO .....	169
8	Schlussbemerkung.....	170
9	Literaturverzeichnis.....	171
10	Anhang .....	175
10.1	Geräte der fertigungstechnischen Demonstrationsanlage.....	175
10.2	Geräte der verfahrenstechnischen Demonstrationsanlage .....	176
10.3	Fragebogen zur Funktionsanalyse der Feldgeräte im gesamten Lifecycle.....	178





## 1 Einleitung

Die ständige Leistungszunahme und Miniaturisierung im Bereich der Mikroelektronik hat in den letzten Jahren nicht nur unmittelbaren Einfluss auf den PC-Markt genommen, sondern ebenso eine starke Veränderung in der Automatisierungstechnik bewirkt. So werden Feldgeräte, aufgrund der vorhandenen Rechenleistung in der Komponente, zunehmend „intelligenter“ – hierdurch wurde es ermöglicht einen Großteil der Funktionalität von der Leitebene in die Feldgeräteebene zu verlagern. Die umfangreiche und komplexe Funktionalität heutiger hochintelligenter Feldgeräte führt einerseits zu enormen Vorteilen hinsichtlich der Flexibilität und der zahlreichen Einsatzbereiche, andererseits lassen die umfangreichen Möglichkeiten die Komplexität des Gesamtsystems erheblich ansteigen. Der enorme Umfang an Funktionalität ist für den Anwender nur schwer beherrschbar. Die Gerätehersteller bieten zwar zur Bedienung, Parametrierung, Konfiguration, Wartung, etc. ihrer Komponenten eigene Softwaretools an, diese unterscheiden sich jedoch von einander nicht nur durch die Bedienphilosophie der Tools selbst. Vielmehr bieten gerade die Feldgeräte einen auf zufällige Art gewachsenen Funktionsvorrat an, der sich hauptsächlich durch die Namensgebung und die Schnittstellen unterscheidet, und folglich von Hersteller zu Hersteller variiert. Hieraus resultiert eine neue Geschlossenheit bezüglich der Projektierung, Inbetriebnahme und Wartung intelligenter Feldgeräte. Die enormen Vorteile werden deshalb, in Form von hohem Aufwand und den daraus resultierenden Kosten für das Engineering automatisierungstechnischer Anlagen, wieder abgeschwächt. Dagegen lässt sich durch ein einheitliches, durchgängiges Bedienmodell der Engineeringprozess für automatisierungstechnische Anlagen rationeller abwickeln, wodurch letztendlich Kosten gespart werden.

Zur Verbesserung des Engineerings von intelligenten Feldgeräten wurde im Rahmen des bmb+f Verbundprojekts DeKOS (*Dezentrale Kooperierende Offene MikroSysteme*) ein einheitliches Konzept entwickelt, um die Handhabung der Geräte zu vereinfachen. Der verfolgte Ansatz war hierbei ein Paradigmenwechsel weg von der parameterorientierten Sicht auf die Gerätefunktionalität hin zu einer funktionsorientierten Sicht. Die Analyse der Gerätefunktionalität von verschiedensten Geräteklassen und Herstellern ergab eine Aufteilung in Produktivfunktionalität, d.h. die eigentliche Funktion, die das Gerät im Betrieb zu erfüllen hat, und Bedienfunktionalität, d.h. diejenige Funktionalität die dazu dient das Gerät in den Zustand zu bringen, damit die Produktivfunktionalität optimal genutzt wird. Dabei übersteigt die Menge der Bedienfunktionen die Menge der Produktivfunktionen, meist nur eine pro Gerät, bei weitem. Hier erfolgte im Rahmen des Projekts eine Standardisierung sowohl über Geräteklassen, als auch über Hersteller Grenzen hinweg. Spezielles Know-how eines Hersteller liegt in der Regel ohnehin in der Produktivfunktionalität verborgen, z.B. wie schnell und präzise steht ein Messwert zur Verfügung,

während die Funktionalität zur Bedienung eines Gerätes von jedem Hersteller gleichermaßen zur Verfügung gestellt werden muss.

Nach einer Analyse der Anforderungen sowohl der Geräteanwender als auch der Gerätehersteller wurde das sogenannte DeKOS-Bedienmodell entwickelt. Dieses beinhaltet eine Bibliothek aus vereinheitlichten Bedienfunktionen mit einer definierten Semantik. Zusätzlich dazu sind Interfaces festgelegt, die eine der Bedienerrolle angepasste Sicht auf die Funktionalität definieren. Die Beschreibung des Bedienmodells erfolgt plattformunabhängig mittels XML<sup>1</sup>.

Das DeKOS-Bedienmodell selbst stellt keine Implementierungsvorschrift dar. DeKOS-Funktionalität kann in den unterschiedlichsten Tools, bzw. auf unterschiedlichen Plattformen implementiert werden und ist vollkommen unabhängig vom unterlagerten Kommunikationssystem. Ebenfalls stellt das Bedienmodell keine Konkurrenz zu bereits existierenden Beschreibungssprachen wie z.B. EDDL<sup>2</sup>, bzw. Geräteprofilen dar. Vielmehr können bestehende Standards dazu eingesetzt werden, um DeKOS-Funktionalität zu beschreiben, z.B. EDDL, oder die Implementierung von DeKOS-Funktionalität zu erleichtern, z.B. Profile. Zur Verifikation der Ergebnisse erfolgten im Rahmen des Projekts Implementierungen in Form von Bedien-Proxies für Feldgeräte auf unterschiedlichen Plattformen. Als besonders geeignet für das Proxy-Konzept hat sich dabei die FDT/DTM-Technologie (*Fieldbus Device Tool / Device Type Manager*) erwiesen, so dass mittlerweile für die Demonstrationsanlagen des Projekts eine Reihe von DeKOS-konformen DTMs für die unterschiedlichsten Geräte existieren. Anhand dieser Implementierungen konnte das DeKOS-Bedienmodell in Zusammenarbeit mit den beteiligten Herstellerfirmen validiert werden.

Das vorliegende Buch behandelt ausführlich die Problematik des Geräte-Engineerings und beschreibt die Lösungen, die im Rahmen des DeKOS-Projekts zur Verbesserung der Situation geschaffen wurden. Dabei werden sowohl die theoretischen und formalen Grundlagen der DeKOS-Ergebnisse als auch die praktischen Umsetzungen detailliert erklärt. Im Folgenden wird ein kurzer Überblick über den Inhalt der einzelnen Kapitel gegeben:

Um einen Einstieg in die Thematik zu geben, wird in Kapitel zwei die zuvor angesprochene Problematik der neuen Geschlossenheit von Feldgeräten auf der Bedienebene ausführlich diskutiert und anhand von Beispielen verdeutlicht.

Fortführend verschafft Kapitel drei einen umfassenden Überblick über das Verbundprojekt DeKOS, beschreibt Projektstruktur, Umfang und Projektpartner, und fasst die erreichten Ergebnisse, sowie deren Verwertung in der industriellen Praxis zusammen.

---

<sup>1</sup> Extensible Markup Language

<sup>2</sup> Electronic Device Description Language

In Kapitel vier wird das entwickelte DeKOS-Bedienmodell ausführlich beschrieben und dabei die einzelnen Bestandteile auch in ihrem Zusammenspiel erklärt, so dass dem Leser letztendlich die Grundlagen für eine eigene Umsetzung vermittelt werden.

Die entwickelten Methoden zur formalen Beschreibung des Bedienmodells und dessen Bestandteile werden in Kapitel 5 ausführlich behandelt.

Aufbauend auf den vermittelten Grundlagen beschreibt Kapitel sechs die praktische Realisierung der einzelnen Proxy-Prototypen zur Validierung der Projektergebnisse. Dabei wird insbesondere auch auf den Vorgang der Erstellung einer DeKOS-konformen Beschreibung für die einzelnen Geräte, sowie auf die unterlagerten Software-Architekturen eingegangen.

Abschließend beschäftigt sich das Kapitel sieben mit der Industriellen Einführung der Projektergebnisse.