

Informatik-Fachberichte

Herausgegeben von W. Brauer
im Auftrag der Gesellschaft für Informatik (GI)

31

Firmware Engineering

Seminar veranstaltet von der gemeinsamen
Fachgruppe „Mikroprogrammierung“ des GI Fach-
ausschusses 3/4 und des NTG-Fachausschusses 6
vom 12. – 14. März 1980 in Berlin



Herausgegeben von W. K. Giloi



Springer-Verlag
Berlin Heidelberg New York 1980

Herausgeber

W. K. Giloi

TU Berlin

Institut für Technische Informatik – CAMP

Einsteinufer 37

1000 Berlin 10

Der Programmausschuß:

H. Berndt

W. H. Burkhardt

W. K. Giloi

R. Hartenstein

R. Klett

H. Painke

R. Piloty

L. Richter

H. Schleich

R. Wendelin

AMS Subject Classifications (1980): 68-02

CR Subject Classifications (1974): 4.1, 4.21, 4.6, 6.2

ISBN-13: 978-3-540-10304-2

e-ISBN-13: 978-3-642-67774-8

DOI: 10.1007/978-3-642-67774-8

Library of Congress Cataloging in Publication Data Main entry under title: Firmware engineering. (Informatik-Fachberichte; 31) Papers in English or German of the seminar held in 1980. Bibliography: p. Includes index. 1. Microprogramming—Congresses. I. Giloi, Wolfgang. II. Gemeinsame Fachgruppe "Mikroprogrammierung" (Ger.) III. Series. QA76.6.F49. · 001.64'2. · 80-23952

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks.

Further, storage or utilization of the described programmes on data processing installations is forbidden without the written permission of the author.

Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to the publisher, the amount of the fee to be determined by agreement with the publisher.

© by Springer-Verlag Berlin · Heidelberg 1980

Druck- und Bindearbeiten: fotokop wilhelm weihert KG, Darmstadt

2145/3140 - 5 4 3 2 1 0

VORWORT DES HERAUSGEBERS

Seit vielen Jahren ist die Mikroprogrammierung das hauptsächliche Mittel zur Realisierung des Instruktionssatzes eines Rechners. In den letzten Jahren hat sich die Mikroprogrammierung aber auch zu einer Methode entwickelt, um die "semantische Lücke" zwischen den höheren Programmiersprachen und der Rechnerhardware zu verringern ("Spracharchitekturen"), um durch die Verlagerung besonders häufiger Operationen in die Firmware Leistungssteigerungen zu erzielen und um Kernoperationen des Betriebssystems dadurch besser zu schützen. Die selektive Verlagerung von Funktionen, die bisher der Software vorbehalten waren, in die Firmware und damit von einer höheren in eine tiefere Ebene der Hierarchie von Interpretationssystemen, durch die sich ein Rechner als Software-Hardware-Komplex modellieren läßt, nennt man daher die "vertikale Verlagerung". Daneben findet man vielfach auch die Ansicht vertreten, daß in Zukunft ganze Programme möglicherweise gleich in den Mikrocode übersetzt werden, statt sie wie bisher zunächst in eine konventionelle Maschinensprache zu übersetzen, die dann durch einen Emulator auf den Mikrocode abzubilden ist.

Die Erstellung der Mikroprogramme zur Emulation einer konventionellen Maschinensprache - einschließlich der eventuellen vertikalen Verlagerung von Systemfunktionen - ist ein Problem, das hauptsächlich den Entwerfer eines Rechnersystems angeht. Die Tatsache, daß es inzwischen auch Rechner mit 'writable control store' gibt, in den der Benutzer eigene Mikroprogramme laden kann, konfrontiert in steigendem Maße aber auch den Benutzer mit dem Problem der Mikroprogrammierung, zum Beispiel, um eine vorhandene Anlage für die spezielle Aufgabe, für die sie eingesetzt werden soll, besonders zu konditionieren ("fine tuning"). Es liegt auf der Hand, daß der Benutzer dies nur dann erfolgreich durchführen kann, wenn ihm entsprechende Programmier- und Testwerkzeuge zur Verfügung stehen.

Es gibt Maschinen, deren Mikroprogramm einige zehntausend Anweisungen umfaßt. Solche umfangreichen Mikroprogramme kommen bisher weniger durch Verlagerung von Software in Firmware zustande als durch eine starke Erweiterung der Hardware-Funktionalität, insbesondere auf dem Gebiet der Diagnose- und Wartungshilfe und durch integrierte Ein/Ausgabe-Steuerungen. Eine massive Verlagerung kann aber auch motiviert sein weniger durch Leistungserwägungen (wie bei der "vertikalen Verlagerung" allgemein üblich), sondern aufgrund der Tatsache, daß man für die Firmware Schutzrechte erlangen kann, die es für die Software nach der Entscheidung des U.S. Supreme Court nicht gibt. In jedem Falle zeigt dieses Beispiel die Komplexität an, die in gewissen Fällen die Firmware einer Rechenanlage erlangen kann. Der Entwurf, die Implementierung, die Verifizierung und unter Umständen auch die Wartung solch umfangreicher Firmware wirft ähnliche Probleme auf, wie wir sie vom Software Engineering her kennen.

Inspiziert durch die Fortschritte des Software Engineering hat sich im Bereich der Mikroprogrammierung in den letzten Jahren auch eine Disziplin des Firmware Engineering - zumindest in Ansätzen - entwickelt. Firmware Engineering beschäftigt sich mit der Anwendung wissenschaftlicher Prinzipien in der Praxis der Spezifikation, Verifikation, Konstruktion, Dokumentation und Wartung von Mikroprogrammen. Das steigende Interesse an einer speziellen Methodik zur Entwicklung und Herstellung von Firmware beruht maßgeblich auf der Tatsache, daß aus den oben angeführten Gründen mit einem starken Anstieg der Komplexität und Funktionalität von Firmware zu rechnen ist, so daß sich Endprodukt-Anforderungen, wie sie bisher nur für die Systemsoftware bestehen, auch auf die Firmware übertragen werden.

In Vorbereitung dieser Entwicklung wurde in einem von der Fachgruppe "Mikroprogrammierung" des Fachausschusses 3/4 der Gesellschaft für Informatik vom 12. - 14. März 1980 durchgeführten Seminars über FIRMWARE ENGINEERING versucht, eine Einführung und Übersicht über existierende und vorgeschlagene Firmware-Engineering-Techniken und -Werkzeuge zu geben sowie ein Forum des Erfahrungsaustauschs zwischen Firmware-Ingenieuren zu bieten. Dabei wurden nicht nur die bisher entwickelte Methodik für die Spezifikation, Implementierung, Verifizierung und Wartung von Firmware vorgestellt, sondern es wurde auch versucht, die Frage zu beantworten, ob sich nicht einfach die bekannten Software-Engineering-Methoden auch auf die Firmware anwenden lassen, oder aber wirklich spezifische Firmware-Engineering-Methoden benötigt werden. Breiten Raum wurde auch der Frage nach Sinn und Nutzen der vertikalen Verlagerung gewidmet, und es wurden repräsentative Fallstudien für solche Verlagerungen vorgetragen.

Die etwa 70 Teilnehmer des Seminars - die zum überwiegenden Teil in der Industrie tätig sind - bekamen durch einen Fragebogen die Gelegenheit geboten, ihre Bewertung und Kritik der Veranstaltung mitzuteilen und Anregungen für zukünftige Seminare dieser Art zu geben. Die Bewertung war überwiegend sehr positiv; soweit Kritik geübt wurde, bestand diese hauptsächlich darin, daß eine Reihe der Vorträge "zu akademisch" gewesen seien, daß im Verhältnis zu den Vortragenden, die als Wissenschaftler an Hochschulen tätig sind, zuwenig Vortragende aus der Industriepraxis berichteten.

Falls dies zutraf, war dies sicherlich nicht auf die Zusammensetzung der veranstaltenden Fachgruppe "Mikroprogrammierung" zurückzuführen. Sie besteht zu gleichen Teilen aus im Industriebereich und im Hochschulbereich Tätigen, wobei die Letzteren noch zum Teil aus der Industrie kommen oder mit ihr durch gemeinsame Forschungsprojekte verbunden sind. Sie hatte durchaus das Bestreben, eine praxisbezogene, nicht-akademische Veranstaltung zu bieten. Wenn dennoch mitunter der Eindruck entstehen konnte, einige der Themen seien zu akademisch behandelt worden, so reflektiert dies nicht zuletzt auch die Tatsache, daß es sich beim Firmware Engineering um eine neue Disziplin handelt, in der vieles noch im Forschungsstadium ist. In diesem Stadium kommen neue Methoden zwangsläufig zunächst vorwiegend aus dem Forschungsbereich. Selbstverständ-

lich findet solche Forschung nicht nur an Hochschulen sondern mindestens ebenso intensiv in der Industrie statt - es besteht hier in der Tat ein wechselseitiger fruchtbarer Austausch - und die am Ende erzielten, wirklich praktikablen Methoden entstehen naturgemäß meist erst aus der Anwendung in der Industrie heraus. Man darf aber auch nicht vergessen, daß wegen der strategischen Bedeutung, die die Firmware (im Hinblick auf Leistungssteigerung und Erhöhung der Funktionalität einer Maschine bzw. auf die Erlangung von Schutzrechten) für das Marketing haben kann, die Hersteller nicht immer unbedingt ihre Methoden und Werkzeuge anderen mitzuteilen wünschen. Schließlich ist zu bedenken, daß die steigende Komplexität der Aufgabenstellungen im Firmwarebereich wie im Softwarebereich die Entwicklung formaler Spezifikations-, Entwurfs- und Verifikationsmethoden notwendig macht, an die sich der in der Praxis Tätige unter Umständen erst noch wird gewöhnen müssen. Manches, was heute vielleicht noch "akademisch" anmutet, wird unter Umständen die Praxis von morgen sein. In diesem Sinne hoffen wir, daß die in diesem Buch zusammengestellten Beiträge des Firmware Engineering-Seminars vom März 1980 in Berlin dem Leser von Nutzen sein werden.

Berlin, im Mai 1980

W. K. Giloi

INHALTSVERZEICHNIS

	<u>Seite</u>
Firmware - Heute H. Berndt	1
Firmware Engineering: An Extensive Update B.D. Shriver et al.	25
Software Engineering - Firmware Engineering H.K. Berg et al.	72
Spezifikation von Firmware W.K. Giloi et al.	93
Firmware Development Systems, a Survey P. Kornerup	124
Firmware-Entwurfs- und Test-Systeme - Fallstudie H. Prechtl	138
Höhere Programmiersprachen für die Mikroprogrammierung L. Richter	156
Correctness of Firmware - An Overview H.K. Berg	173
Firmware Dokumentation und Wartung R. Hartwich	225
Vertikale Verlagerung - Verfahren, Voraussetzungen, Anwendung P. Albrich	242
Ein mikroprogrammiertes Unterbrechungswerk für einen Prozeßrechner R. Klett	273
Die vertikale Verlagerung von Systemfunktionen im System Nixdorf 8864 G. Schleich	280