

Informatik-Fachberichte 225

Herausgeber: W. Brauer
im Auftrag der Gesellschaft für Informatik (GI)

Wilhelm Stoll

Test von OSI-Protokollen



Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo Hong Kong

Autor

Wilhelm Stoll

Weingartener Straße 66, D-7520 Bruchsal 4

CR Subject Classification (1987): C.2.1, C.2.2, D.2.5

CIP-Titelaufnahme der Deutschen Bibliothek.

Stoll, Wilhelm:

Test von OSI-Protokollen / Wilhelm Stoll. – Berlin; Heidelberg; New York; London; Paris; Tokyo:

Springer, 1989

(Informatik-Fachberichte; 225)

Zugl.: Karlsruhe, Univ., Diss., 1989 u.d.T.: Stoll, Wilhelm:

Werkzeuge und Methoden für den Test von OSI-Protokollen

ISBN-13: 978-3-540-51824-2 e-ISBN-13: 978-3-642-83993-1

DOI: 10.1007/978-3-642-83993-1

NE: GT

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der Fassung vom 24. Juni 1985 zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Springer-Verlag Berlin Heidelberg 1989

2145/3140 – 543210 – Gedruckt auf säurefreiem Papier

Vorwort

Mit der stark wachsenden Anzahl heterogener Rechnernetze gewinnt der Test der eingesetzten Kommunikationssoftware bezüglich Konformität und Kompatibilität immer mehr an Bedeutung. Die im vorliegenden Buch vorgestellten Werkzeuge und Methoden bieten sowohl dem Anwender als auch dem Ersteller derartiger Kommunikationssoftware Möglichkeiten zum Test der eingesetzten bzw. in der Entwicklung stehenden Kommunikationssoftware.

Das Karlsruher Testsystem (KATE) ermöglicht eine automatische Testdurchführung. Bedingt durch die zugrunde liegende Konzeption ist das KATE nahezu unabhängig von der Testumgebung und der Implementierung der Kommunikationssoftware. Dies wurde durch zahlreiche Konfigurierungen in der Testumgebung bestätigt, die durch das experimentelle, heterogene OSI-Netzwerk am Institut für Telematik der Universität Karlsruhe zur Verfügung stand. Die Konfigurierungen wurden zum Test der OSI-Protokolle vorgenommen, die im Projekt 'Kommunikation in Netzwerken über ISO/OSI-Protokolle' (Kooperation IBM - Universität Karlsruhe) eingesetzt wurden. Durch die Tests konnten Unstimmigkeiten bezüglich der Kompatibilität und der Konformität der getesteten Protokollimplementierungen aufgezeigt werden. Die verantwortlichen Implementierer konnten anhand der bei den Tests angelegten Ergebnis-Dateien Rückschlüsse auf die Ursachen der Unstimmigkeiten ziehen und die Unstimmigkeiten beheben. Weiterhin konnte das KATE ohne größere Probleme von der Firma Philips Kommunikations Industrie AG Nürnberg zum Test ihrer OSI-Transportprotokollimplementierung in einer Testumgebung erfolgreich eingesetzt werden, die aus mehreren Mikro-VAXen und mehreren Philips-Rechnern besteht. Die eigentliche Mächtigkeit des KATE wird durch das Testkoordinierungsprotokoll und durch den Testagenten erreicht. Durch die besondere Realisierungsform des Testagenten wird sowohl die Beobachtung aller zwischen den Protokollinstanzen ausgetauschten Informationen als auch ein intensiver Test der Robustheit des Prüflings ermöglicht. Das Testkoordinierungsprotokoll stellt ein wesentliches Hilfsmittel zur Synchronisierung der während der Testdurchführung auftretenden Aktivitäten dar.

Mit dem TRANS-CHECK-Verfahren wurde ein Werkzeug zur automatischen Generierung von Testdaten in Form von Testsequenzen geschaffen. Die durch das TRANS-CHECK-Verfahren erzeugten Testsequenzen zeichnen sich durch kleine Längen und durch eine große Fehlererkennungsmächtigkeit aus. Wie Vergleiche mit ähnlichen Testdatengenerierungsverfahren gezeigt haben, wurde durch das TRANS-CHECK-Verfahren eine Verbesserung hinsichtlich der Testsequenzlänge bei nur geringfügiger Verschlechterung der Fehlererkennungsmächtigkeit erreicht. Mit dem erweiterten TRANS-CHECK-Verfahren konnte auch die Fehlererkennungsmächtigkeit gegenüber gleichartigen Verfahren (auf Kosten der Testsequenzlänge) wesentlich verbessert werden.

Eine Erweiterung des Protokollautomaten und die entsprechende Anpassung des TRANS-CHECK-Verfahrens ermöglichte die automatische Generierung von Testszenarien, die neben den eigentlichen Testdaten auch die für die Testdurchführung benötigten Steuer- und Kontrollkommandos enthalten. Bei der Generierung werden allerdings keine

Testfälle zur Überprüfung der Wertebereiche von Parametern (Dienstelementparameter, Paketkopfkomponenten usw.) erzeugt.

Die Test Control Language (TCL) erlaubt die Spezifizierung ausführbarer Testschritte. Durch die Einbeziehung von Blöcken und Schleifen in die Syntax der TCL können Testfälle für den intensiven Test der Robustheit und der Leistung einer Protokollimplementierung spezifiziert werden. Die Syntax dieser Sprache ist allerdings speziell auf die Konzeption des KATE ausgerichtet und kann somit nicht zur Spezifizierung von Testschritten für andere Testsysteme verwendet werden.

Die Festlegung einer Teststrategie und die Testauswertung stehen in engem Zusammenhang. So kann z.B. bei der während der Testauswertung stattfindenden Analyse eine spezielle Teststrategie zur Eingrenzung der Ursachen, die zur 'Nichterfüllung' der Testziele geführt haben, notwendig werden. Die Wahl der Teststrategie ist dann abhängig von der Art der Unstimmigkeiten zwischen dem beobachteten und dem erwarteten Verhalten.

Die vorliegende Arbeit entspricht inhaltlich meiner im Februar 1989 von der Fakultät für Informatik der Universität Karlsruhe genehmigten Dissertation mit dem Thema *Werkzeuge und Methoden für den Test von OSI-Protokollen*. Die Dissertation entstand, während ich als wissenschaftlicher Mitarbeiter am Lehrstuhl für Telematik der Fakultät für Informatik an der Universität Karlsruhe tätig war.

Mein besonderer Dank gilt Herrn Prof. Dr. O. Drobnik für die Anregung zum Thema und für die Betreuung der Arbeit. Er hat mit zahlreichen Ratschlägen und Diskussionen sehr zum Gelingen der Arbeit beigetragen.

Herrn Prof. Dr. G. Krüger sei für das Korreferat und für seine Hinweise und Ratschläge zur Gestaltung der Arbeit gedankt. Herrn Prof. Dr. G. Krüger möchte ich auch meinen Dank dafür aussprechen, daß er die Arbeit an seinem Lehrstuhl ermöglicht hat.

Den Herren D. Gantenbein und Dr. E. Mumprecht des IBM-Forschungslabors in Rüschlikon (Schweiz) sei für ihre Ratschläge und ihre Unterstützung bei der Realisierung des Karlsruher Testsystems gedankt.

Darüber hinaus danke ich allen Kollegen am Lehrstuhl für Telematik, insbesondere Herrn Dr. H. Krumm, für die hilfreiche Unterstützung; den Studenten, besonders den Herren M. Bannert, G. Blakowski, R. Bolz, R. Brandner, R. Göstenmeier und G. Klug, bin ich für ihr großes Engagement bei der Durchführung des praktischen Teils der Arbeit ebenfalls dankbar; nicht zuletzt sei auch meiner Frau Barbara für ihr Verständnis und für ihre Unterstützung gedankt.

Karlsruhe, im August 1989

Wilhelm Stoll

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 1 |
| 1.1 | Problematik | 1 |
| 1.2 | Bekannte Werkzeuge und Methoden | 2 |
| 1.3 | Motivation | 2 |
| 1.4 | Gliederung der Abhandlung | 3 |
| 2 | Test | 5 |
| 2.1 | Grundprinzipien des Testens | 5 |
| 2.2 | Test von OSI-Protokollen | 6 |
| 2.2.1 | OSI-Basisreferenzmodell | 8 |
| 2.2.2 | Besonderheiten | 9 |
| 2.3 | Testverfahren | 10 |
| 2.3.1 | Verfahren auf der Basis von Testzentren | 11 |
| 2.3.2 | OSI-Testverfahren | 12 |
| 2.4 | Testdaten für den Protokolltest | 15 |
| 3 | Automatische Generierung von Testdaten | 21 |
| 3.1 | Grammatikverfahren | 21 |
| 3.2 | Automatenverfahren | 22 |
| 3.2.1 | Definitionen | 22 |
| 3.2.2 | Transition Tour Verfahren | 24 |
| 3.2.3 | W-Methode | 25 |
| 3.2.4 | Checking Sequence Verfahren | 26 |
| 3.2.5 | UIO-Verfahren | 28 |
| 3.2.6 | Random Sequence Verfahren | 30 |
| 3.2.7 | Bewertung | 30 |
| 3.3 | Probleme | 33 |
| 4 | Das Karlsruher Testsystem | 36 |
| 4.1 | Forderungen | 36 |
| 4.2 | Konzeption | 37 |
| 4.2.1 | Testkoordinator | 39 |
| 4.2.2 | Testtreiber | 41 |
| 4.2.3 | Testagent | 44 |
| 4.2.4 | Testkoordinierungsprotokoll | 50 |
| 4.3 | Vergleich mit anderen Testverfahren | 52 |
| 4.4 | Bewertung | 59 |
| 5 | TRANS-CHECK-Verfahren | 65 |
| 5.1 | Algorithmus | 65 |
| 5.1.1 | Minimierung der Testsequenzlänge | 66 |
| 5.1.2 | Definitionen | 71 |

| | | |
|------------------|---|------------|
| 5.1.3 | Phase 1 | 74 |
| 5.1.4 | Phase 2 | 79 |
| 5.2 | Bewertung des TRANS-CHECK-Verfahrens | 83 |
| 5.2.1 | Maximale Länge der erzeugten Testsequenz | 83 |
| 5.2.2 | Fehlererkennungsmächtigkeit | 83 |
| 5.3 | Vergleich mit anderen Verfahren | 88 |
| 6 | Generierung von Testszenarien | 92 |
| 6.1 | Anforderungen | 92 |
| 6.2 | Test Control Language | 95 |
| 6.3 | Erweiterung des Protokollautomaten | 97 |
| 6.4 | Verfahren zur automatischen Erstellung | 112 |
| 6.5 | Abarbeitung eines Testszenarios | 120 |
| 7 | Teststrategie | 125 |
| 7.1 | Testszenarien | 126 |
| 7.1.1 | Änderung der Testszenariostruktur | 126 |
| 7.1.2 | Änderung der Test Control Language | 127 |
| 7.1.3 | Änderung des Generierungsverfahrens | 129 |
| 7.1.4 | Bewertung | 131 |
| 7.2 | Deterministisches/indeterministisches Protokollverhalten | 132 |
| 7.2.1 | Auswirkungen auf die Testdurchführung | 132 |
| 7.2.2 | Maßnahmen zur Eindämmung indeterministischer Verhaltensweisen | 134 |
| 7.2.3 | Testauswertung | 138 |
| 7.3 | Festlegung einer Teststrategie | 140 |
| Anhang A. | Definitionen und Theoreme | 146 |
| Anhang B. | Testsequenzen für den Automaten B | 152 |
| B.1 | TRANS-CHECK-Verfahren | 153 |
| B.2 | Transition Tour Verfahren | 154 |
| B.3 | W-Methode | 155 |
| B.4 | Checking Sequence Verfahren | 156 |
| B.5 | UIO-Verfahren | 157 |
| Anhang C. | Experimente | 158 |
| C.1 | Experimente für das Transition Tour Verfahren | 158 |
| C.2 | Vergleich der Testdatengenerierungsverfahren | 159 |
| C.2.1 | Transition Tour Verfahren | 159 |
| C.2.2 | W-Methode | 159 |
| C.2.3 | Checking Sequence Verfahren | 160 |
| C.2.4 | UIO-Verfahren | 161 |
| C.2.5 | Zusammenfassung | 162 |

| | |
|---|------------|
| Anhang D. Zeitliche Abfolge der Testausführung | 163 |
| Anhang E. Algorithmen | 170 |
| E.1 Erweiterungsalgorithmus | 170 |
| E.2 Algorithmus zur Darstellung der Funktionsweise des Testkoordinators | 175 |
| Anhang F. Automaten | 192 |
| F.1 Erweiterter Automat B2'' | 192 |
| F.2 Erweiterter Automat B3' | 197 |
| Literatur | 202 |