

Überdeckungskriterien auf Modellebene - Erfahrungsbericht -

Mirko Conrad

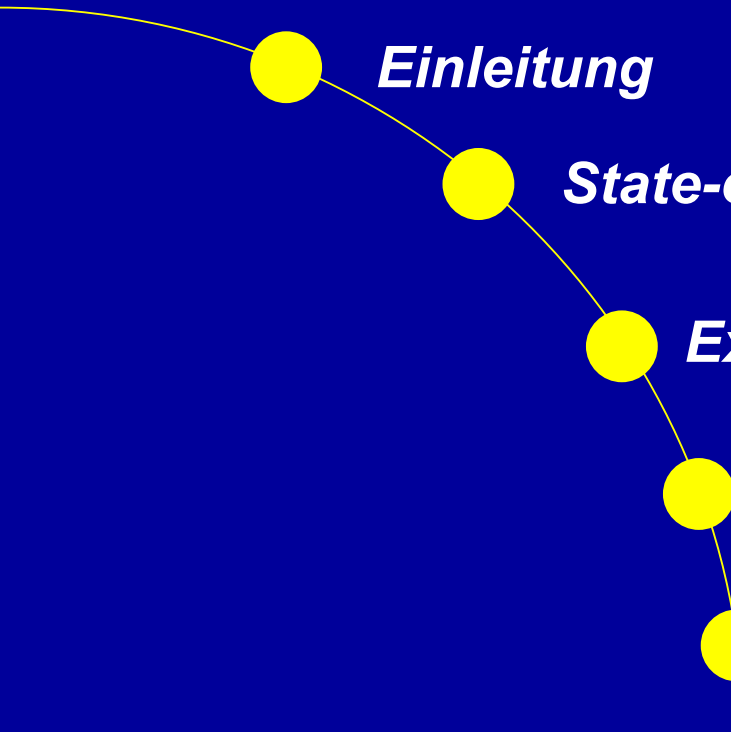
DaimlerChrysler AG

 Mirko.Conrad @ DaimlerChrysler.com

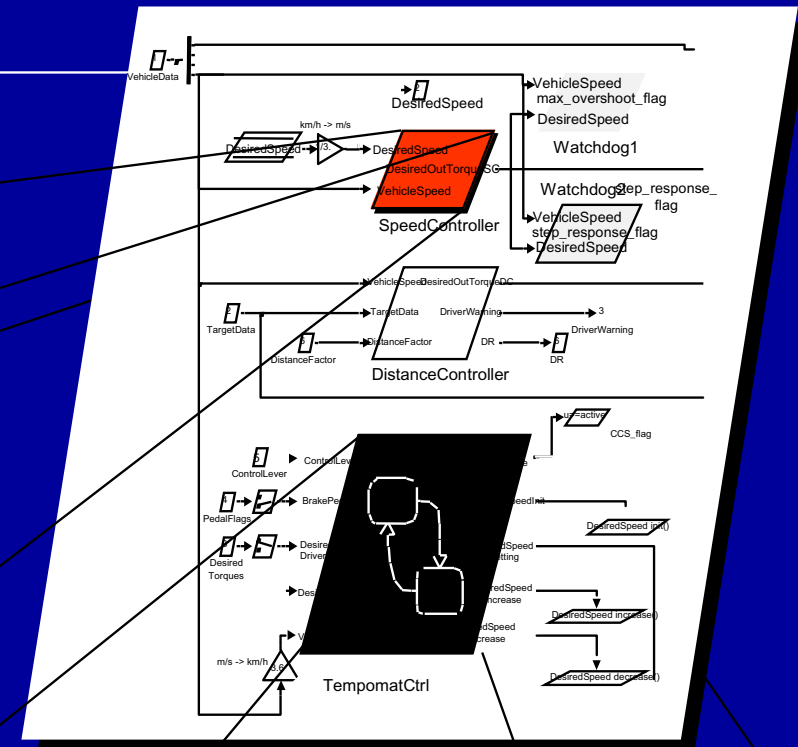
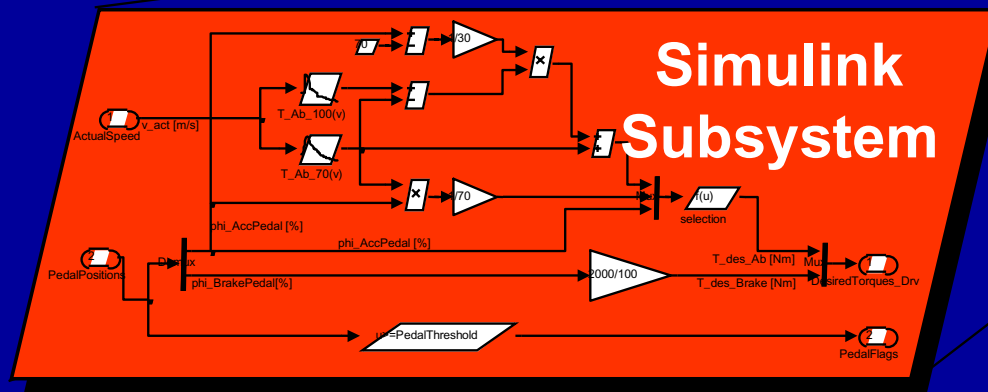
Sadegh Sadeghipour

ITPower Consultants

 S.Sadeghipour@ITPower.de

-
- 
- Einleitung***
 - State-of-the-Art: Model Coverage***
 - Experimentelle Untersuchungen***
 - Lessons learned***
 - Zusammenfassung und Ausblick***

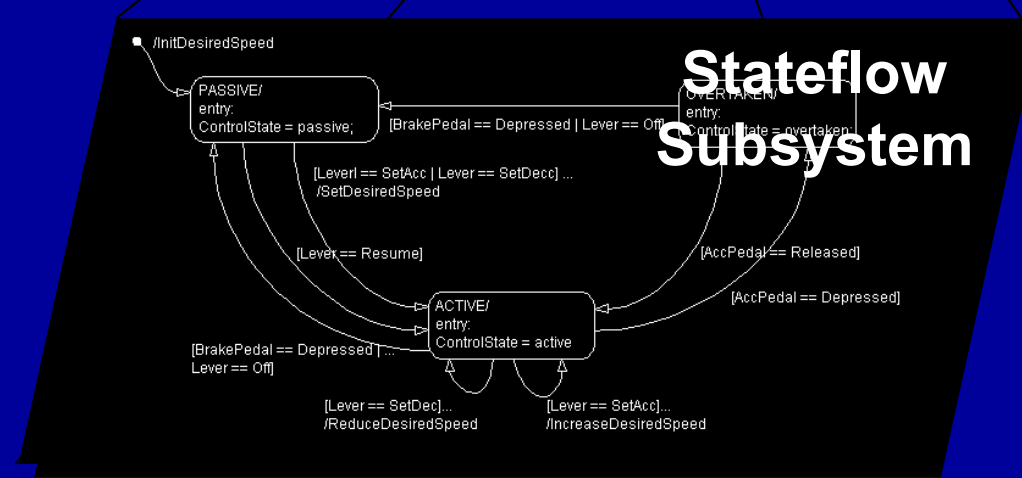
Modell-basierte Entwicklung



Modell-basiertes Design

mit Blockschaltbildsimulatoren wie Simulink/Stateflow erstellte ausführbare Spezifikationen, d.h.

- Funktionsblock-orientierte, berechnungskausale mathematische Modelle
- Komplexitätsbeherrschung durch hierarchisch strukturierte Modularität (Dekomposition in Subsysteme)



Modell-basierter Test

- ~ Die Modell-basierte Entwicklung ermöglicht und erfordert zugleich neue Herangehensweisen an den Softwaretest.
- ~ Nutzung des ausführbaren Modells als zusätzliche, reichhaltige Informationsquelle für den Test:
 - Modell als Testobjekt (Modelltest / MiL)
 - Modell als Testbasis für die Ermittlung der Testszenarien
 - Wiederverwendung der Testszenarien aus Modelltest (MiL SiL PiL HiL)
 - Modell als Testorakel
 - Modell als Basis für strukturorientierte Überdeckungsmessungen (Model Coverage)

Modell-basierter Test

mit der Modell-basierten Entwicklung eng verzahnter Testprozess, der eine Kombination unterschiedlicher, sich gut ergänzender Testmethoden umfasst und dabei das ausführbare Modell als reichhaltige Informationsquelle für den Test benutzt.

State-of-the-Art: Model Coverage Komponenten verschiedener Simulations/Modellierungswerkzeuge

Statemate Coverage Tool

Statecharts

ObjectGEODE TestComposer

SDL

IAR Visual State Dynamic Analysis Tool

Statecharts

Test RealTime / Rose RealTime integration

Statecharts

Reactis Simulator / Tester

Blockdiagramme / Statecharts

Beacon Tester (AUTT)

Blockdiagramme / Statecharts

Simulink / Stateflow Model Coverage Tool

Blockdiagramme / Statecharts

State-of-the-Art: Commercial Model Coverage Tools for Simulink/Stateflow (Oct. 2002)

Reactis Simulator / Tester

Reactive Systems Inc.

www.reactive-systems.com

- n test vector generation for structural coverage
- n 2 coverage metrics for SL models / 3 coverage metrics for SF models

Beacon Tester

Applied Dynamics International

www.adi.com/products/be_bss_te.htm

- n test vector generation for structural coverage
- n > 15 coverage / static analysis metrics for SL/SF models

Simulink / Stateflow Model Coverage Tool

The MathWorks Inc.

www.mathworks.com/products/slperftools

- n 4 coverage metrics for SL/SF models

State-of-the-Art: Reactis Simulator / Tester

record how many of a given class of syntactic constructs that appear in a model have been executed at least once

Simulink metrics:

- n **Conditional subsystem coverage.**
A (conditional) subsystem is deemed covered if has been executed at least once in some simulation step.
- n **Branch coverage.** Blocks include: switch, multiport switch, relational op., logical op., dead zone, and saturation.

Stateflow metrics:

- n **State coverage.** which states have been entered at least once.
- n **Condition action coverage.** which transition segments have had their condition actions executed at least once.
- n **Transition action coverage.** which transition segments have had their transition actions executed at least once.

State-of-the-Art: Beacon Tester (1)

exhaustively analyse and
examine the coverage
of the software architecture

Path-Related (Structural Coverages)

- n Structured Path Coverage
- n Modified Condition / Decision Coverage
- n Decision (Branch) Coverage
- n Statement Coverage
- n Boundary Value Analysis (arithmetic comparisons)

Value-Related Stressing (Numerical Coverages)

- n Input Range Stressing
- n Output Range Stressing
- n Table Data Access

Interface-Related

- n Stub Hierarchy Coverage
- n Inlined Hierarchy Coverage

State-of-the-Art: Beacon Tester (2)

Static Analysis Metrics

- n McCabe Cyclomatic Complexity Metric (Standard / Modified)
- n Predicate Equation Production for Control Paths
- n Control Path Enumeration
- n Data Item Listing with Usage Information
- n Vertical Output Coupling Analysis
- n Supremum Range Analysis
- n Supremum Scaling Analysis

State-of-the-Art: Simulink/Stateflow Model Coverage Tool

SL/SF Model Coverage Tool

part of the Simulink performance tools available since advent of Matlab R12 (2000)

4 different coverage metrics for SL/SF models

- n Decision Coverage (D1) ^{1) 2) 3)}
- n Condition Coverage (C1) ^{1) 3)}
- n MCDC Coverage (MCDC) ^{1) 3)}
- n Look-up Table Coverage (LUT) ¹⁾

- 1) for selected SL blocks
- 2) for SF states
- 3) for SF transitions:

The screenshot shows the Simulink Coverage tool interface. At the top right is the MATLAB logo with the number 6. The main window is titled 'Simulink Coverage' and contains the following sections:

Model Systems:

Model Hierarchy:	Test 1	Test 2	Total
1. Sensor correction and Fault Redundancy	100% (1/1)	50% (5/12)	100% (1/1)
2. ... MAP Estimate	100% (2/2)	50% (1/2)	100% (2/2)
3. ... Speed Estimate	100% (2/2)	50% (1/2)	100% (2/2)
4. ... Throttle Estimate	100% (2/2)	50% (1/2)	100% (2/2)

Details:

1. Subsystem "Sensor correction and Fault Redundancy"

Total Decision Coverage 100% (1/1), 6 decisions, 6 fully covered. (incl. descendants).
 Parent: [fuel_rate_controller](#)
 Child Systems: [MAP Estimate](#), [Speed Estimate](#), [Throttle Estimate](#)

Switch block "Switch"

Total Decision Coverage 100% (2/2), 1 decision, 1 fully covered.
 Parent: [Sensor correction and Fault Redundancy](#)

Decision Coverage Details:

Test of the logical trigger input	100%	50%	100%
... evaluated false (output is from 3rd input port)	5951	7561	14482
... evaluated true (output is from 1st input port)	510	0	510

Switch block "Switch3"

Total Decision Coverage 100% (2/2), 1 decision, 1 fully covered.
 Parent: [Sensor correction and Fault Redundancy](#)

Decision Coverage Details:

Test of the logical trigger input	100%	50%	100%
... evaluated false (output is from 3rd input port)	3753	7561	11254
... evaluated true (output is from 1st input port)	3740	0	3740

Switch block "Switch4"

Total Decision Coverage 100% (2/2), 1 decision, 1 fully covered.
 Parent: [Sensor correction and Fault Redundancy](#)

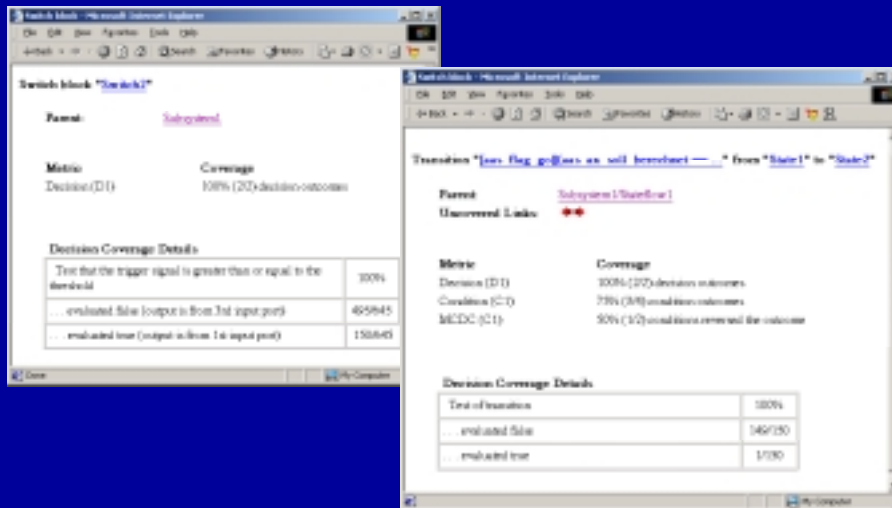
Annotations on the screenshot:

- A bracket on the right side of the 'Model Systems' table points to the text: "Summary coverage information based on the model hierarchy".
- Arrows point from the 'Details' section to the text: "Hyperlinks to the model".
- A bracket on the right side of the 'Decision Coverage Details' table for 'Switch3' points to the text: "Number of times the model component was accessed during the test".

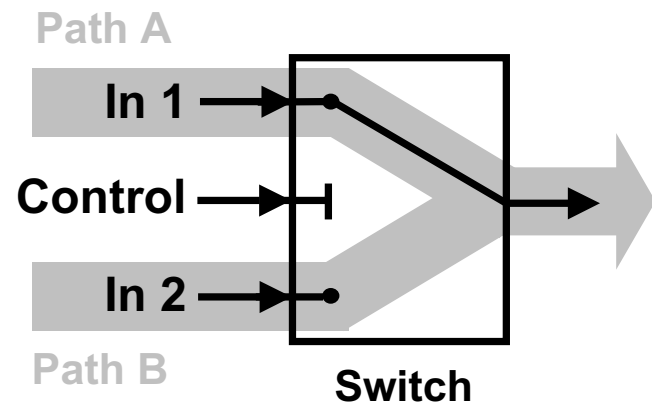
Simulink/Stateflow Model Coverage Tool

Decision Coverage (D1)

- n Examines items that represent decision points in a model, such as the Switch blocks and Stateflow states.
- n For each item, decision coverage determines the percentage of the total number of simulation paths through the item that the simulation actually traverse



Decision Coverage



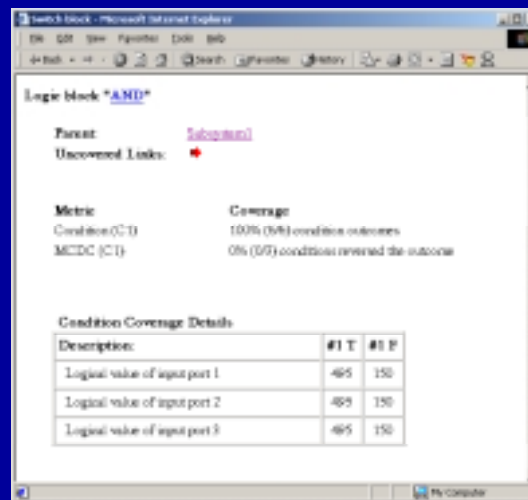
Switch

Pass through In1 when Control ≥ Threshold; otherwise, pass through In2

Simulink/Stateflow Model Coverage Tool

Condition Coverage (C1)

- n Examines blocks that output the logical combination of their inputs, e.g., the Logic block, and Stateflow transitions.
- n A test case achieves full coverage if it causes each input to each instance of a logic block in the model and each condition on a transition to be true at least once during the simulation and false at least once during the simulation.



Logic block "AND"

Parent: [Subsystem](#)

Uncovered Links: *

Metric Coverage

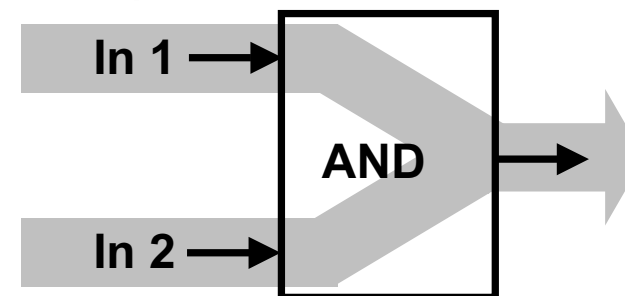
Condition (C1)	100% (5/5) condition outcomes
MEDC (C1)	0% (0/5) conditions reversed the outcome

Condition Coverage Details

Description:	# T	# F
Logical value of input port 1	4/5	1/5
Logical value of input port 2	4/5	1/5
Logical value of input port 3	4/5	1/5

Condition Coverage

In 1 : true / false



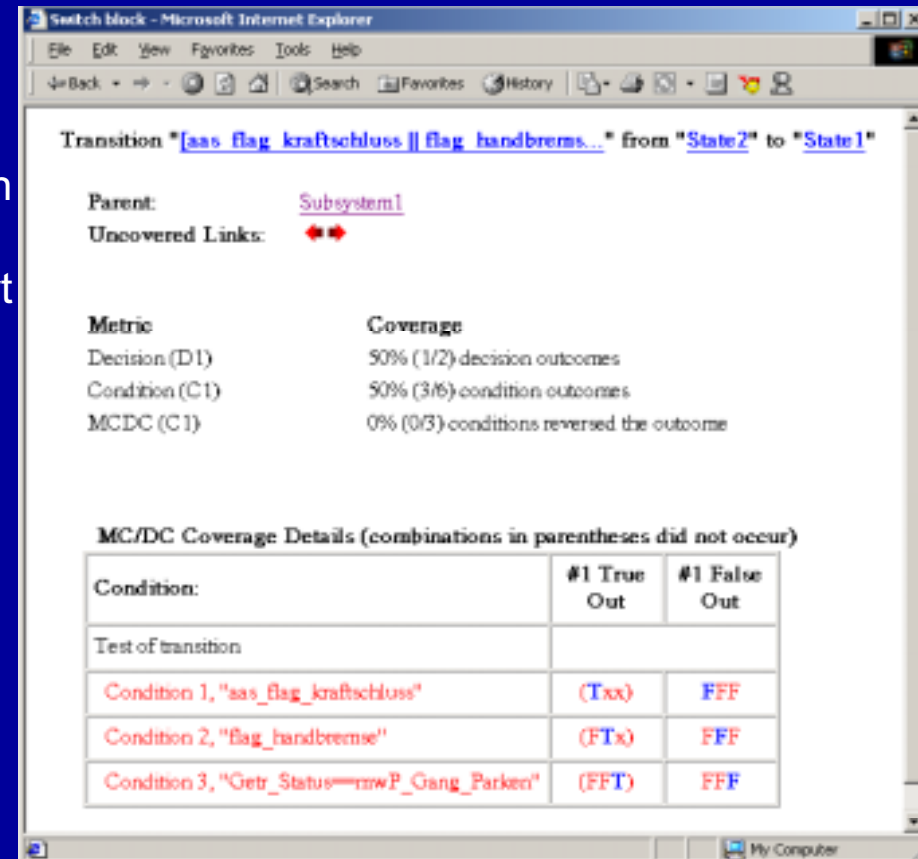
In 2 : true / false

Logical Operator

Simulink/Stateflow Model Coverage Tool

Modified Condition/Decision Coverage (MCDC)

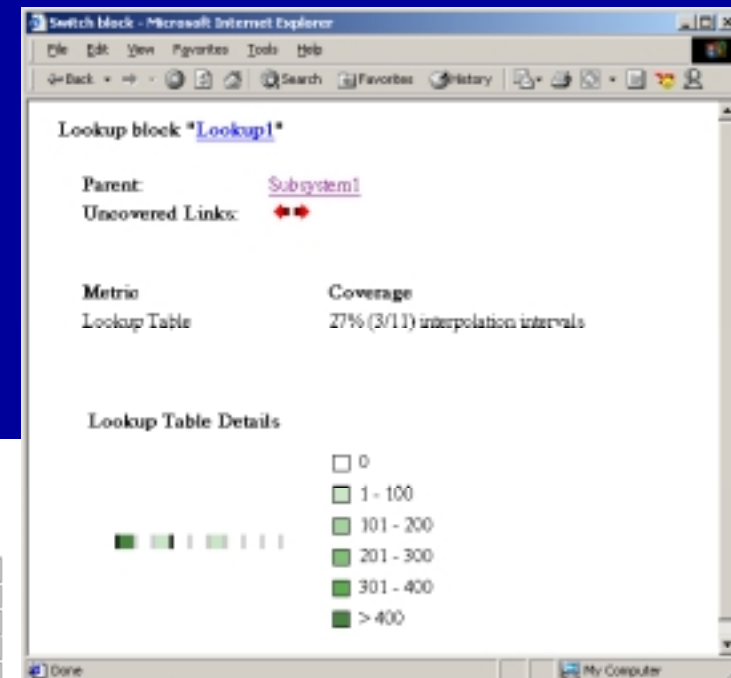
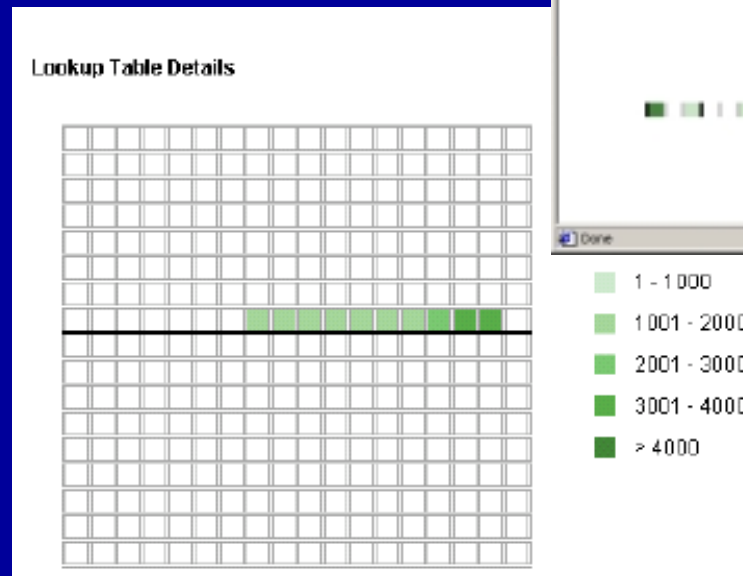
- n Bei diesem Überdeckungskriterium wird die Abhängigkeit des logischen Gesamtausdrucks von den einzelnen Eingaben gemessen. Durch Variation der einzelnen Eingabevariablen bei festgehaltenen anderen Eingaben soll der Wahrheitswert des Gesamtausdrucks einmal wechseln:
- n Gegeben Eingaben i_1, \dots, i_n . Das Kriterium ist zu 100 % erfüllt, wenn es zu jedem i_k Belegungen i_{1k}, \dots, i_{nk} gibt, so dass allein die Variation von i_k den Gesamtwert von falsch zu wahr wechseln lässt.



Simulink/Stateflow Model Coverage Tool

Look-Up Table Coverage (LUT)

- n Examines blocks, such as the 1D Look-Up block, that output the result of looking up one or more inputs in a table of inputs and outputs, interpolating between or extrapolating from table entries as necessary.
- n Lookup table coverage records the frequency that table lookups use each interpolation interval.
- n A test case achieves full coverage if it executes each interpolation and extrapolation interval at least once.



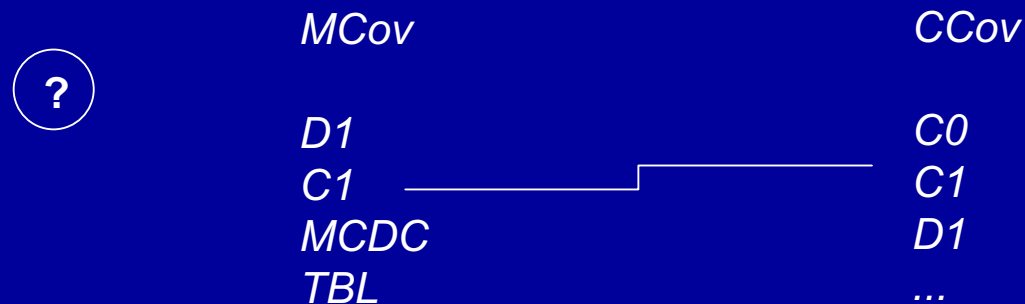
Experimentelle Untersuchungen - Fragestellungen (1)

Läßt sich aus vorhandenen Model Coverage Messungen die zu erwartende Code Coverage abschätzen?

?

$$\begin{array}{rcl}
 MCov & > & CCov \\
 & = & \\
 & < &
 \end{array}
 \qquad
 CCov = f(MCov)$$

Welche Kriterien stehen miteinander in Beziehung?



Subsumation verschiedener Modellüberdeckungskriterien

?

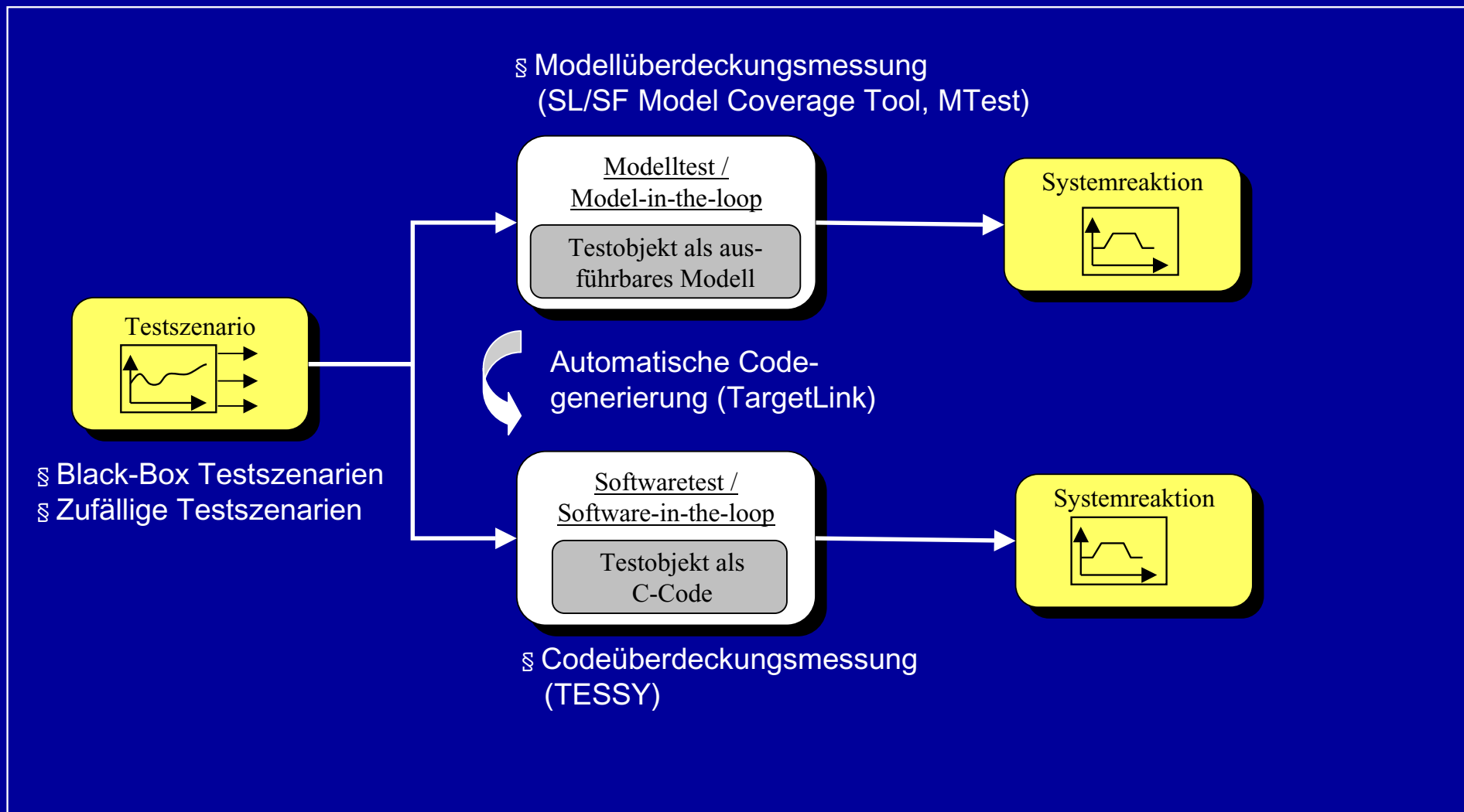
$$D1 > C1 \text{ etc.}$$

Experimentelle Untersuchungen - Fragestellungen (2)

Gründe für Unterschiede zw. Model- und CodeCoverage

- n verschiedene Implementierungen ein und desselben Modellelements
- n Optimierungen des Codegenerators
- n ...

Experimentelle Untersuchungen - Überblick



Experimentally determined model and code coverage (summary)

Test Object	Test	Model Coverage [%]		Code Coverage [%]	
		D1	C1	C0	C1
Subsystem 1	Σ Test Scenarios Black-box Test #1	49	58	57	54
	Σ Test Scenarios Black-box Test #2	37	55	54	43
	Σ Test Scenarios Black-box Test #3	25	42	47	33
	Σ Test Scenarios Black-box Test #4	49	53	51	48
	Σ Black-box Tests	54	66	61	58
	Σ Test Scenarios Random Test	35	62	55	41
	Σ Total	62	84	68	58
Subsystem 2	Σ Test Scenarios Black-box Test #1	63	65	68	66
	Σ Test Scenarios Black-box Test #2	87	82	87	88
	Σ Black-box Tests	88	85	88	89
	Σ Test Scenarios Random Test	75	85	87	79
	Σ Total	95	90	96	95
Subsystem 3	Σ Test Scenarios Black-box Test #1	100	69	100	100
	Σ Black-box Tests	100	69	100	100
	Σ Total	100	69	100	100

Correlation between model and code coverage

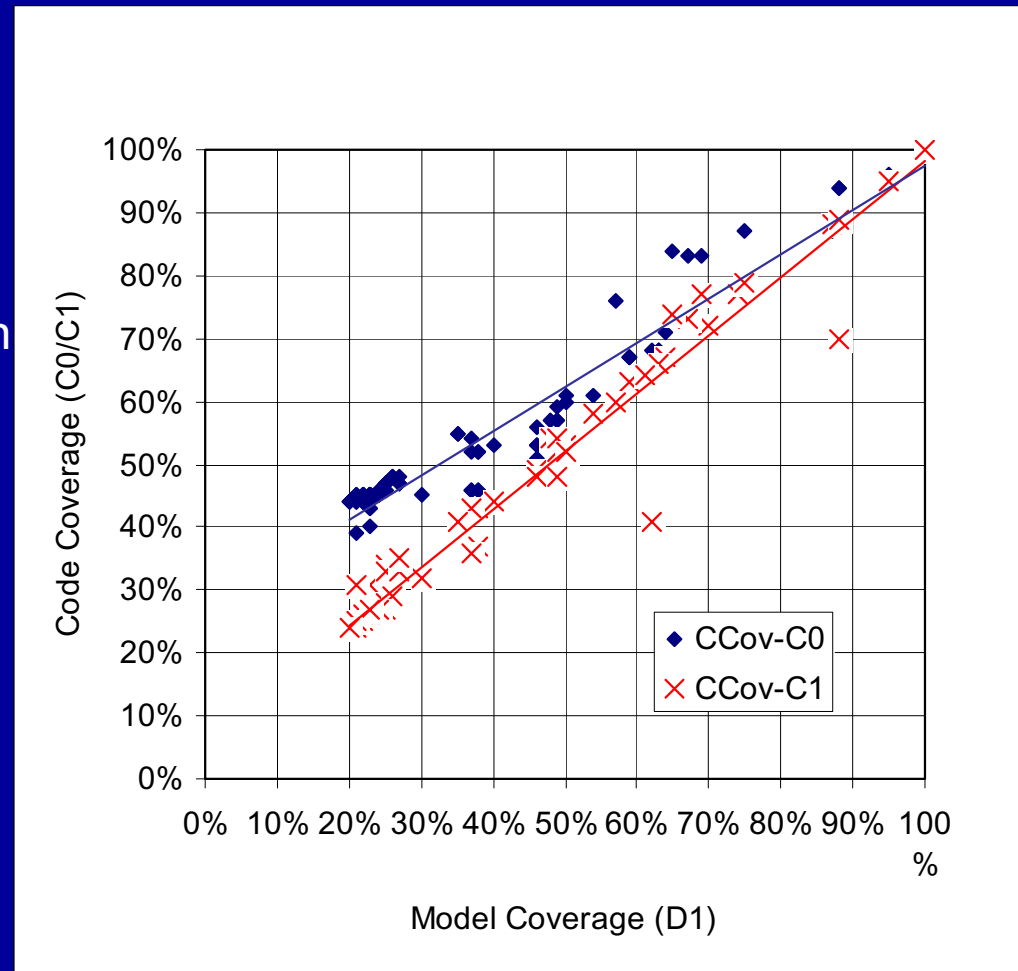
calculated correlation

n between D1 and C1 is 0.983

n between D1 and C0 0.973

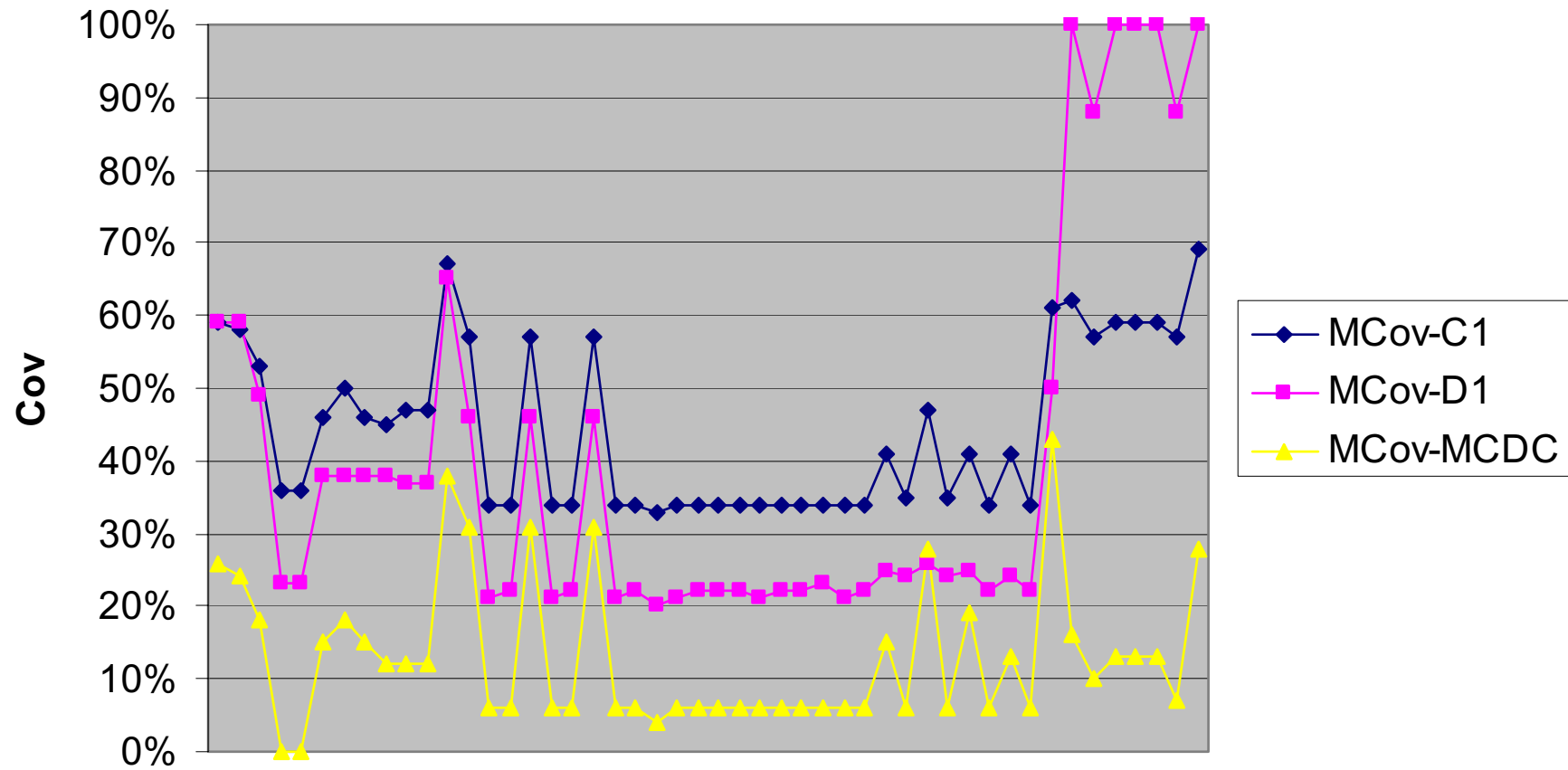
(taking the individual scenarios into account)

close statistical interrelation between D1 coverage at model level and C1 or C0 coverage at code level



Subsumation Relationships

Subsumation



Simulink/Stateflow Model Coverage Tool

Berücksichtigte Modellelemente

- n unterschiedliche Modellelemente für die einzelnen Coverage-Kriterien
- n unterschiedliche Modellelemente in den einzelnen Werkzeugversionen
- n keine tool-unabhängige Beschreibung der Kriterien
- ~ Erschwerte Vergleichbarkeit durch Versions- und Werkzeugabhängigkeit

Block	Decision	Condition	MC/DC	LUT
1D Look-Up				
2D Look-Up				
Abs				
Combin. Logic				
For				
If				
Logic				
MinMax				
Multiport Switch				
Relay				
Saturation				
Subsystem				
Switch				
SwitchCase				
While				

Matlab R12.1

Lessons learned (1)

Die Modell-Überdeckungsmetriken des SL/SF Model Coverage Tools sind überwiegend kontrollfluss-orientiert (Ausnahme LUT coverage)

Während die gängigen Code-Coverage-Kriterien allgemeiner Natur (d.h. werkzeugunabhängig definiert) sind und auf verschiedene imperative Programmiersprachen der 3. Generation anwendbar sind, sind die durch das Model Coverage Tool bestimmbaren Überdeckungsmaße eher toolspezifisch

Die vier durch das Model Coverage Tool bereitgestellten Überdeckungskriterien haben teilweise deutlich unterschiedliche 'Bemessungsgrundlagen'. Die breiteste Bemessungsgrundlage hat das Decision Coverage. Diese Tatsache erschwert die Vergleichbarkeit der einzelnen Model Coverage Maße untereinander.

Die unterschiedlichen Modell-Überdeckungskriterien erlaubten andererseits eine differenziertere Betrachtung der Modell-Überdeckungen: Decision Coverage (D1) ermöglicht einen ersten Eindruck über die Güte der Tests hinsichtlich der Modell-Überdeckung. Die weiteren Metriken können dann zur Ermittlung der Ursachen fehlender Überdeckungen herangezogen werden.

Lessons learned (2)

Die mittels Black-box-Tests erreichten Modellüberdeckungsgrade liegen in der Größenordnung vergleichbarer Code-Überdeckungen

Durch Zufallstests wurden nur geringe Überdeckungsgrade erreicht, jedoch trugen sie zur Erhöhung der Gesamtüberdeckung bei.

Die erreichte Modell-Überdeckung für ein- und denselben Testlauf hängt von der gewählten Simulationsschrittweite ab. Es ist deshalb darauf zu achten, dass Modell und Code mit der gleichen Schrittweite ausgeführt werden. Das selbe trifft auch auf den Solver zu. Wird er beim Übergang zum Code geändert, so kann dadurch die Überdeckung empfindlich abweichen.

In den Experimenten ergab sich ein starker statistischer Zusammenhang zwischen D1-Modellüberdeckung und C0/C1-Codeüberdeckung.

Unter den gegebenen Randbedingungen kann also die C1-Code-Überdeckung durch die D1-Modell-Überdeckung angenähert werden. Die D1-Modell-Überdeckung erlaubt hier eine Abschätzung der zu erwartenden C1-Code-Überdeckung.

Lessons learned (3)

Der exakte mathematische Zusammenhang variiert in Abhängigkeit von den bei der Codegenerierung verwendeten Übersetzungsalgorithmen. So können unterschiedliche Codegeneratoren bzw. unterschiedliche Codegeneratoreinstellungen zu unterschiedlichen Umsetzungen ein und des selben Modellteils und damit potenziell zu unterschiedlichen Überdeckungen führen. Ebenso erscheint es möglich, dass durch Optimierungen des Codegenerators bestimmte im Modell vorhandene Zweige 'wegoptimiert' werden. Eine Nichtüberdeckung dieser Zweige im Modell würde dann im Code nicht mehr auftreten.

Zusammenfassung und Ausblick (1)

Die Modell-basierte Entwicklung ermöglicht, dass neben den verbreiteten strukturorientierten Überdeckungsmaßen auf Code-Ebene auch Strukturtestkriterien auf Modell-Ebene ermittelt und in die Steuerung und Bewertung des Testprozesses einfließen können.

Diese Überdeckungsmaße können bereits frühzeitig im Entwicklungs- und Testprozess, d.h. bevor C-Code vorhanden ist, erhoben werden.

Die Modell-Überdeckung kann während der Durchführung der (funktionalen) Modelltests im Hintergrund erhoben werden. Der Tester kann durch eine grobe Analyse der erzeugten Coverage-Reports feststellen, welche Modellteile nicht geprüft wurden. Zur Abdeckung der zugehörigen Funktionalitäten müssen dann neue Testsequenzen erstellt werden.

Es existieren vergleichbare Modell- und Code-Überdeckungsmaße. Die Korrelation zwischen diesen Maßen hängt von der Art der (manuellen oder automatischen) Umsetzung des Modells in Code sowie von den verwendeten Simulationsparametern ab.

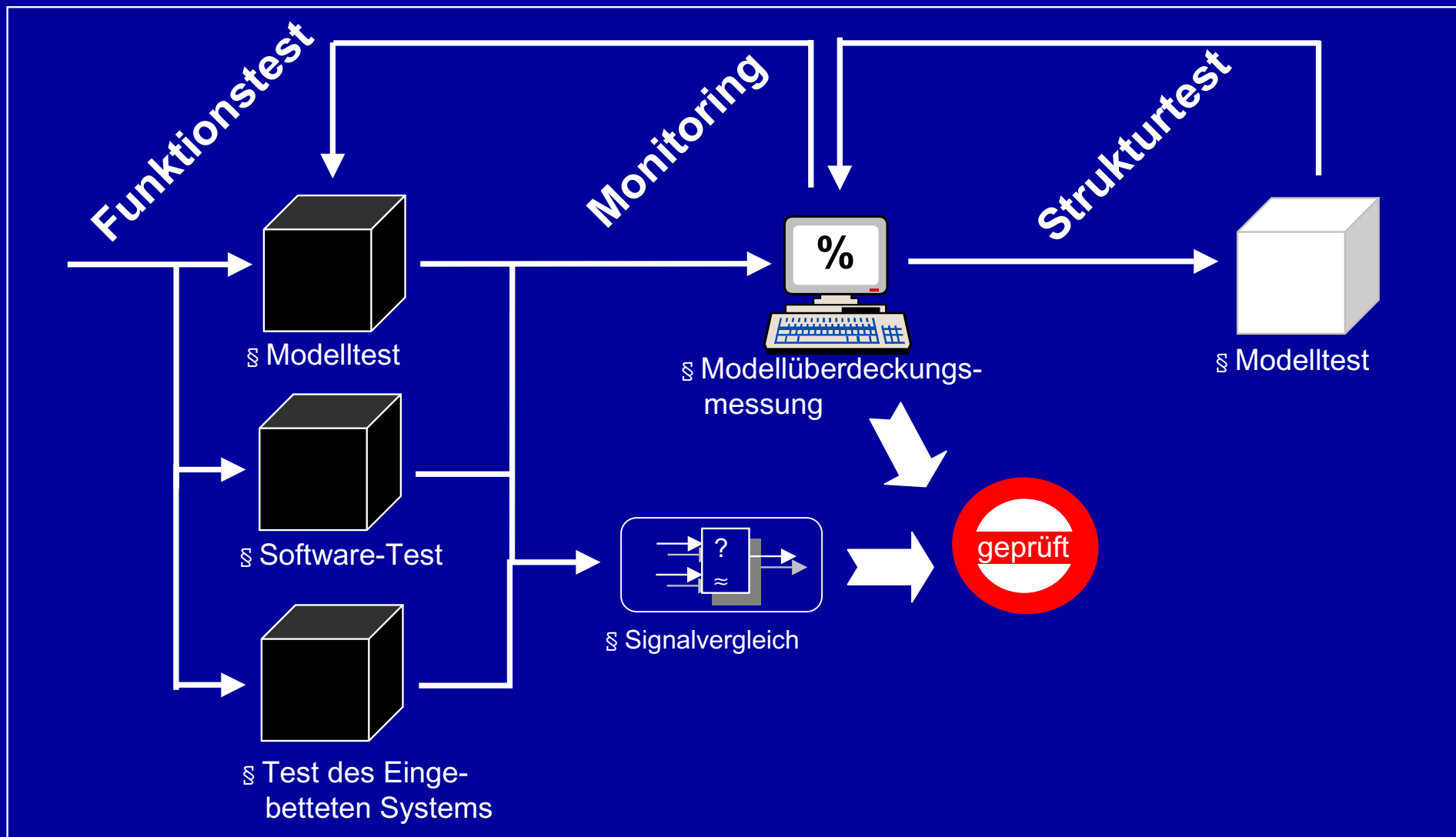
Zusammenfassung und Ausblick (2)

Zu erreichende Modell-Überdeckungen können (neben anderen Kriterien) zur Steuerung des Testumfangs und als (notwendiges) Testendekriterium für den funktionalen Modelltest verwendet werden. Unter bestimmten Umständen erlauben sie eine Abschätzung der zu erwartenden Code-Überdeckung und können zukünftig für die mittels Codegeneratoren erzeugten Programmteile möglicherweise bestimmte Strukturtestkriterien auf Code-Ebene ersetzen.

Ziel weiterer Untersuchungen sollte es sein, die aus dem klassischen Software-Test bekannte effektive Teststrategie an die Besonderheiten des Modell-basierten Tests und der Möglichkeit der Erhebung von Strukturtestkriterien auf Modell-Ebene anzupassen.

In diese effektive Teststrategie für den Modell-basierten Test sollten auch neuere Entwicklungen im Bereich der Verifikations- und Testwerkzeuge einfließen. Bspw. könnten mit Model-Checkern und evolutionären Algorithmen automatisiert Testdaten zur Erreichung bisher nicht überdeckter Modell- oder Codeteile ermittelt bzw. die Nichterreichbarkeit bestimmter Strukturelemente untermauert oder gar bewiesen werden.

Ausblick - Effektive Teststrategie Modell-basierter Test



Literatur Model Coverage

- /Ald02/ W. J. Aldrich: **Using Model Coverage Analysis to Improve the Controls Development Process**. Proc. of AIAA Modeling and Simulation Technologies Conference and Exhibit, Monterey, USA, 2002
- /CS02/ M. Conrad, S. Sadeghipour: **Einsatz von Überdeckungskriterien auf Modellebene – Erfahrungsbericht und experimentelle Ergebnisse**. Softwaretechnik-Trends 22:2, S. 1-6, Mai 2002