

Testabdeckung und systematische Testfallerstellung

Dr. Martin Beißer
sepp.med gmbh

Motivation

Murphys Law

„Wenn es mehrere Möglichkeiten gibt, eine Aufgabe zu erledigen, und eine davon in einer Katastrophe endet oder sonst wie unerwünschte Konsequenzen nach sich zieht, dann wird es jemand genau so machen.“

Motivation



Murphys Law für Softwaresysteme

„Wenn es eine Abfolge von Anwenderaktionen in einem Softwaresystem gibt, die zu einem Fehlverhalten führt, so wird ein Anwender es genauso machen“.

d.h. für den Softwaretest:
„Wenn nicht jede mögliche Anwenderaktion getestet wurde, sind Fehlerzustände nicht auszuschließen.“

3



Umkehrschluss



Ein Softwaresystem ist dann vollständig getestet, wenn alle möglichen Kombinationen von Anwenderaktionen getestet sind

Also vollständige Testabdeckung ==
Test aller möglichen Anwenderaktionen !

Aber wie?

4



Testabdeckung auf Codebasis



- Zeilenüberdeckungstest
 - Ausführung aller Quellcode-Zeilen
- Anweisungsüberdeckungstest (C0-Test)
 - Aufdecken nicht erreichbarer Anweisungen
- Zweigüberdeckungstest (C1-Test)
 - Aufdecken nicht erreichbarer Zweige
- Pfadüberdeckungstest (C2-Test)
 - Testen aller Pfade, möglicherweise unendlich viele
- Bedingungsüberdeckungstest (C3-Test)
 - Verifizierung der Logik

5



Testabdeckung im Blackboxtest



- Anforderungsabdeckung
- Use-Case Abdeckung
- Workflow Abdeckung
- ...

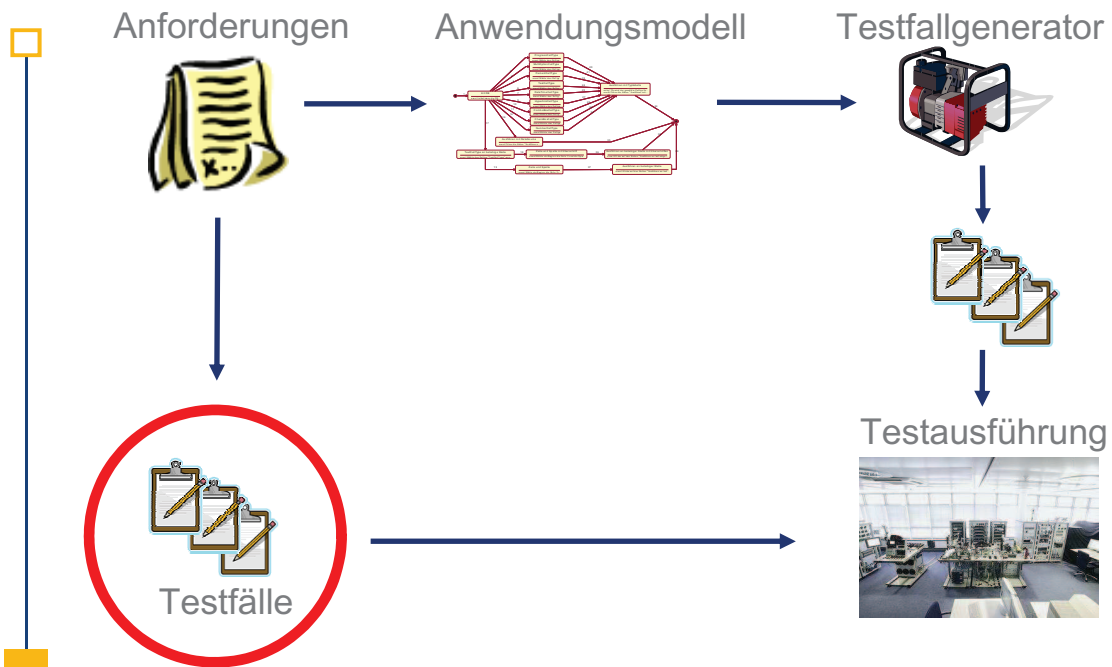
Testabdeckung gegen alle Anwenderaktionen



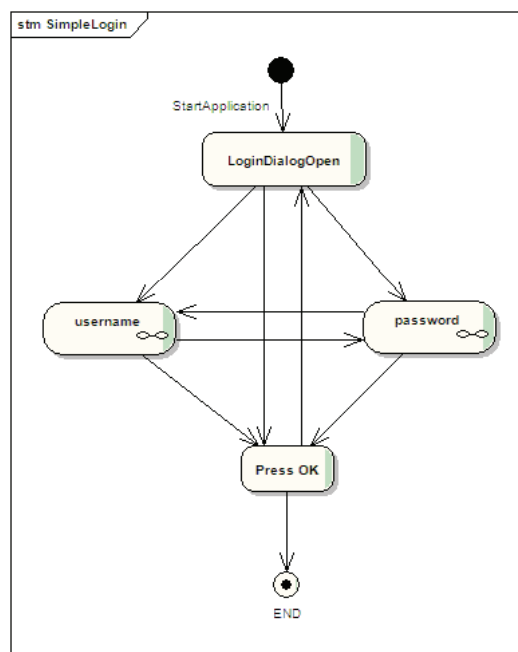
6

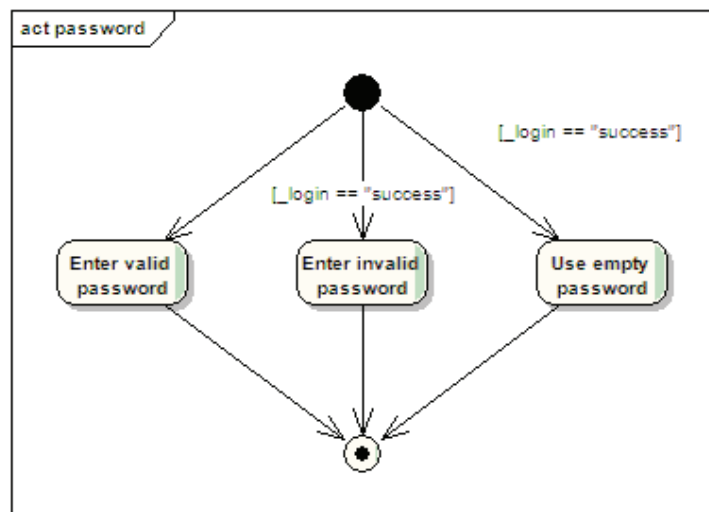
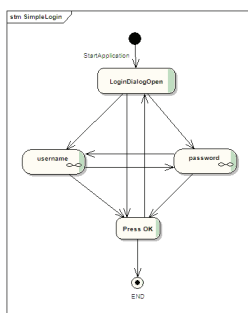
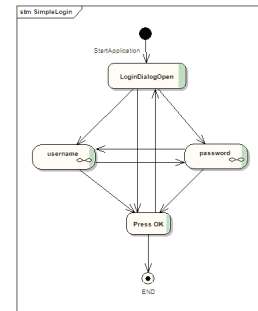
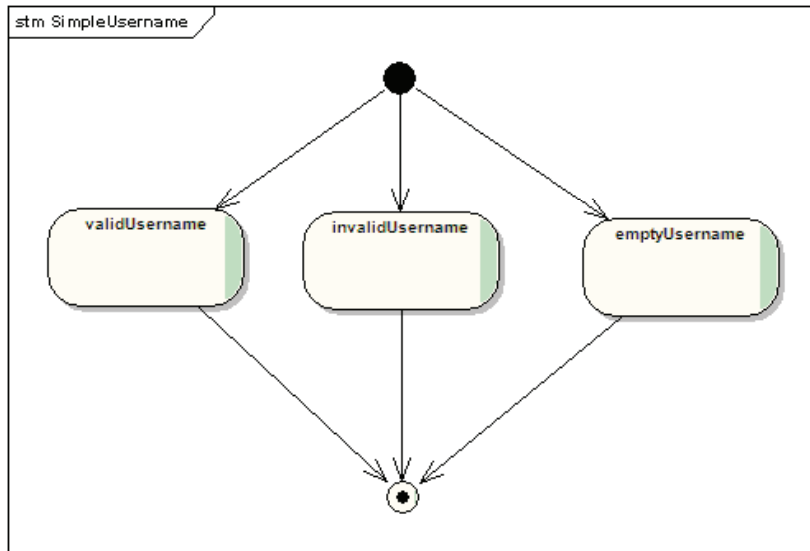


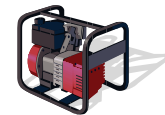
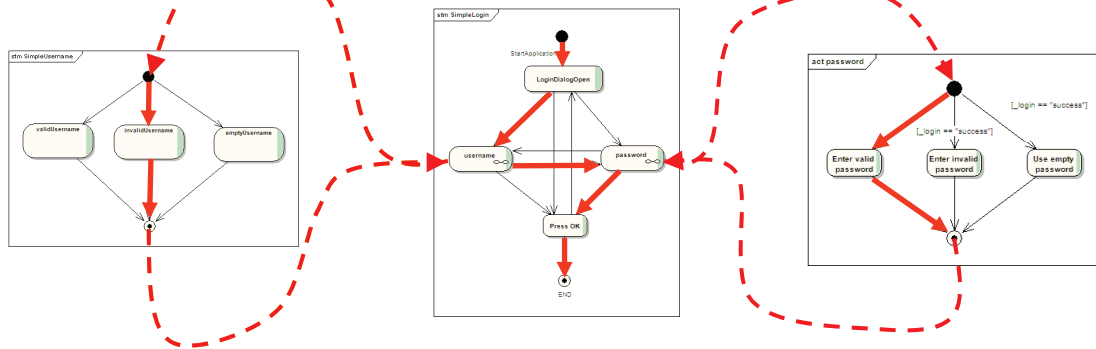
Testdesign



Anwendungsmodell: Login-Dialog







Systematisches Testdesign + Testabdeckung

- Anwendungsmodells des SUT
- Ableitung der Testfälle aus dem Modell
- Definition einer Testabdeckung bzgl. des Modells
 - volle Pfadabdeckung
 - Übergangsabdeckung

Vielen Dank für Ihre Aufmerksamkeit.



Fon: +49 (0) 91 95 - 9 31 - 0
Fax: +49 (0) 91 95 - 9 31 - 300
E-Mail: martin.beisser@seppmed.de
Web: www.seppmed.de

Beispiel: Datengetriebener Test

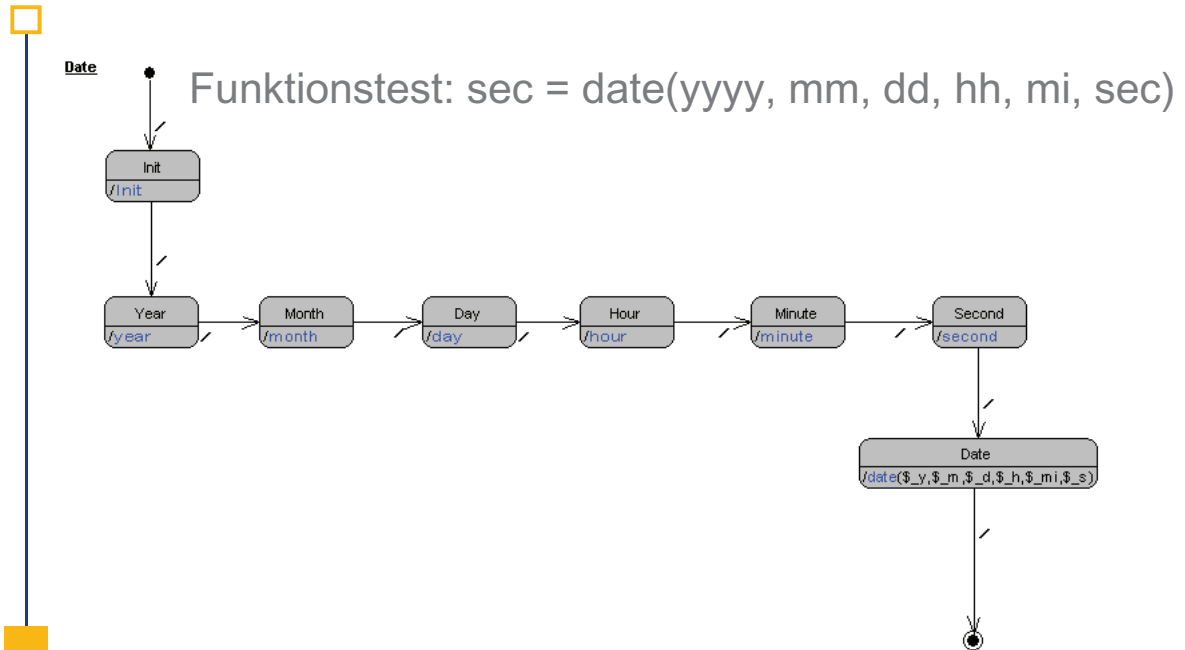


Funktionstest: `sec = date(yyyy, mm, dd, hh, mi, sec)`

Gültigkeitsbereich: 01.01.1970 – 31.12.3000

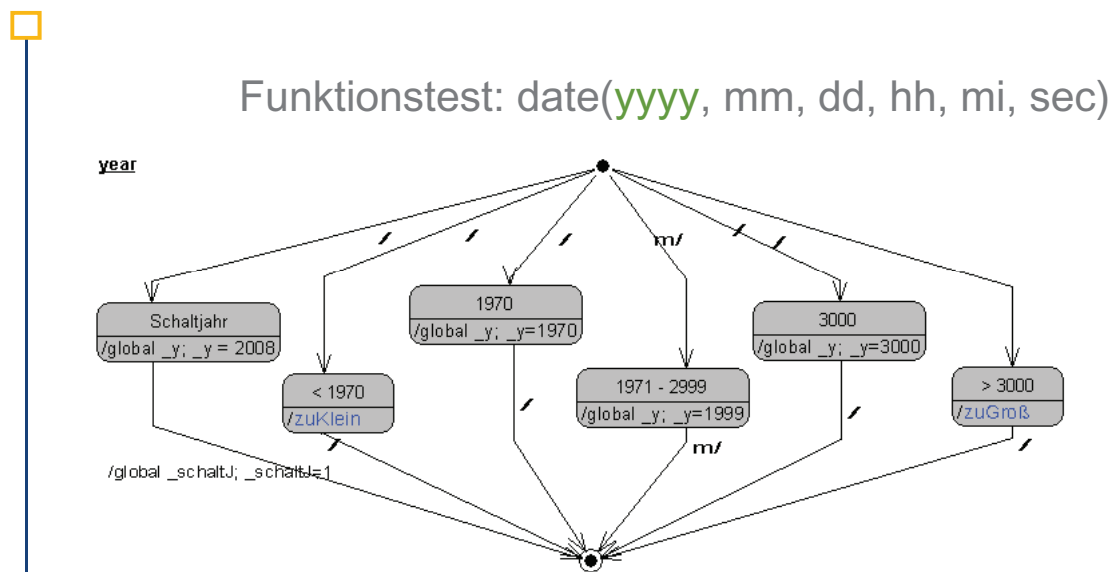


Beispiel: Datengetriebener Test



15

Beispiel: Datengetriebener Test

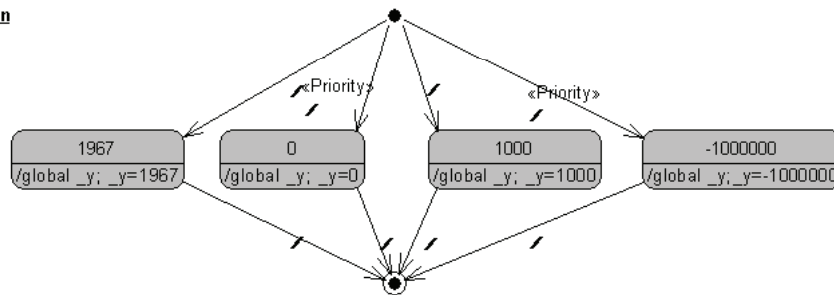


16

Beispiel: Datengetriebener Test

Funktionstest: date(yyyy, mm, dd, hh, mi, sec)

zuKlein

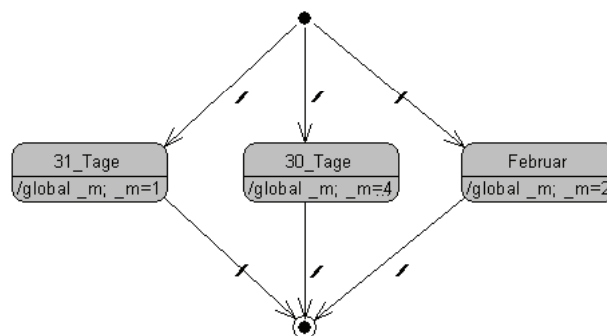


17

Beispiel: Datengetriebener Test

Funktionstest: date(yyyy, mm, dd, hh, mi, sec)

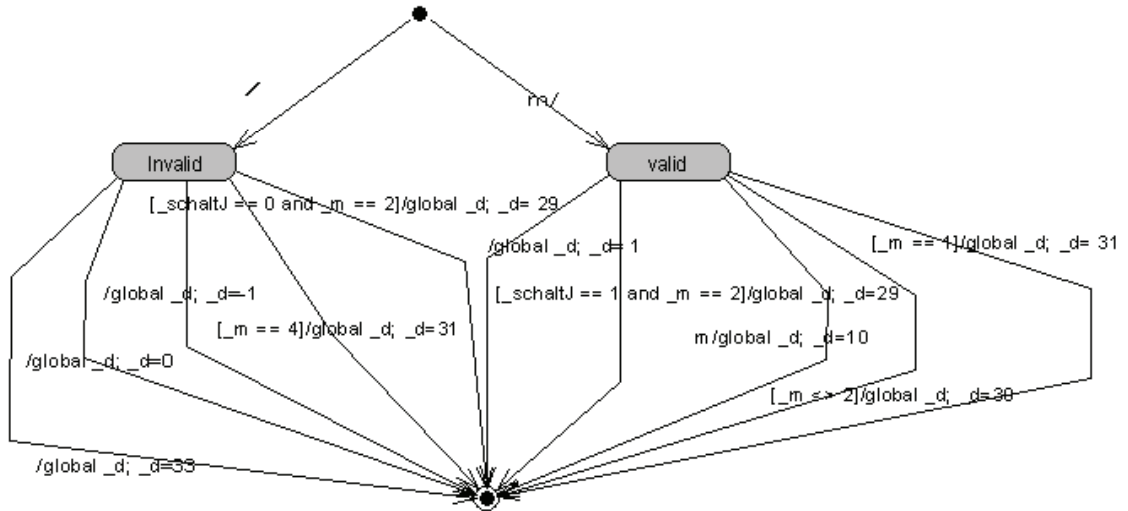
month



18

Beispiel: Datengetriebener Test

Funktionstest: date(yyyy, mm, dd, hh, mi, sec)

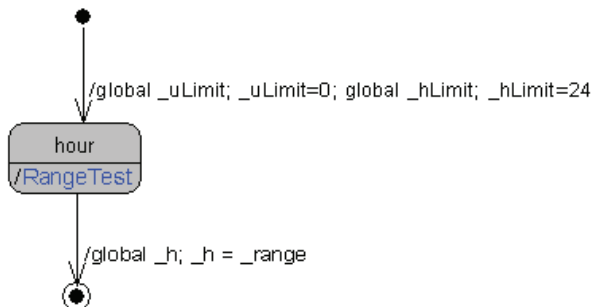


19

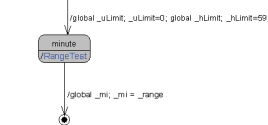
Beispiel: Datengetriebener Test

Funktionstest: date(yyyy, mm, dd, hh, mi, sec)

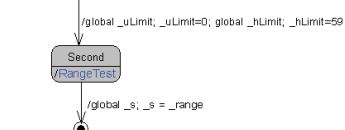
hour



minute



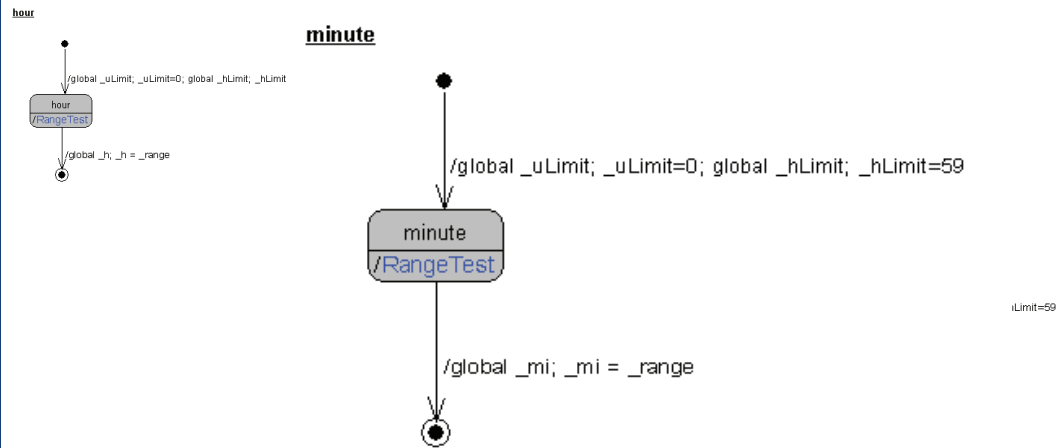
second



20

Beispiel: Datengetriebener Test

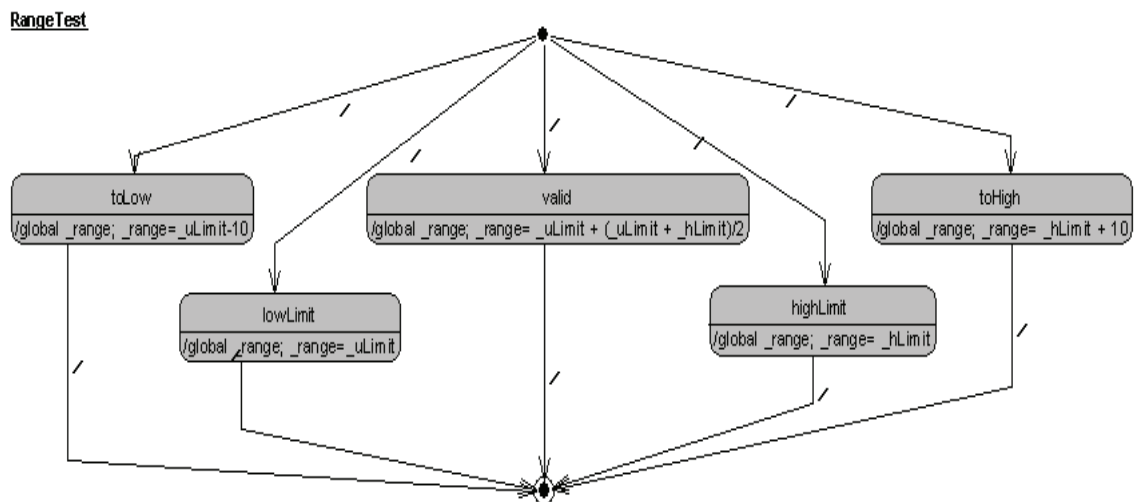
Funktionstest: date(yyyy, mm, dd, hh, mi, sec)



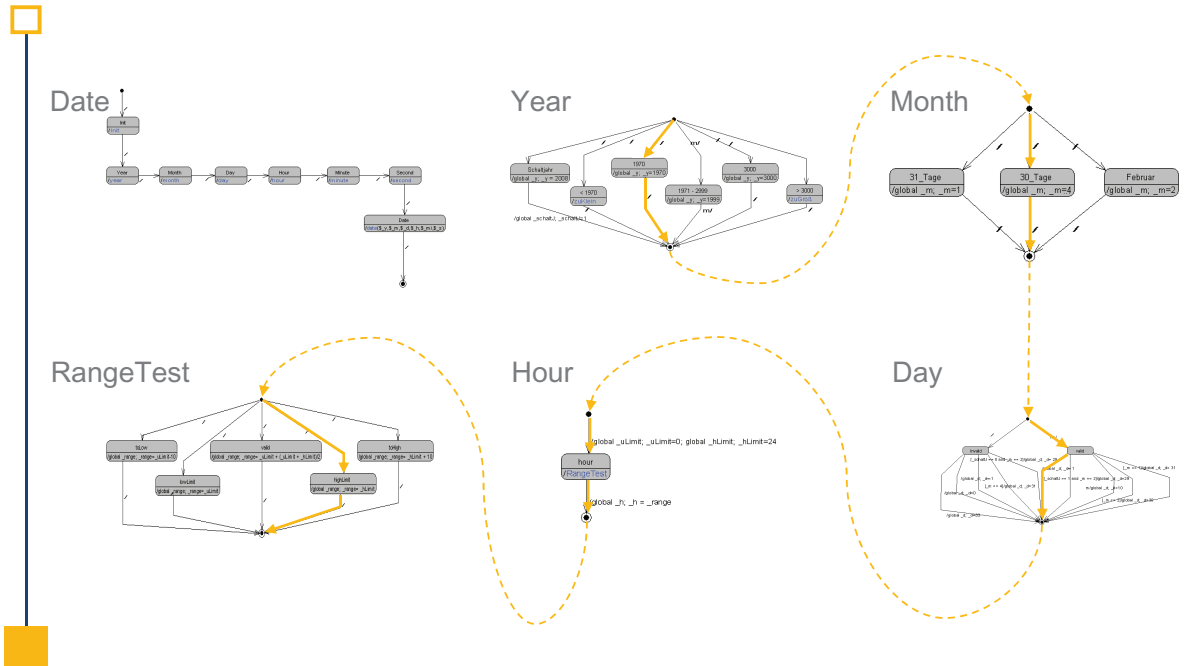
21

Beispiel: Datengetriebener Test

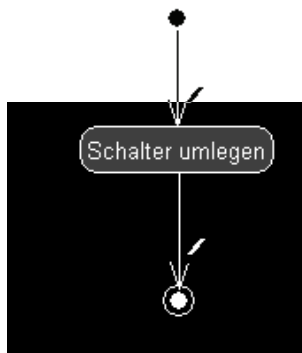
Subdiagram: Grenzwerttest



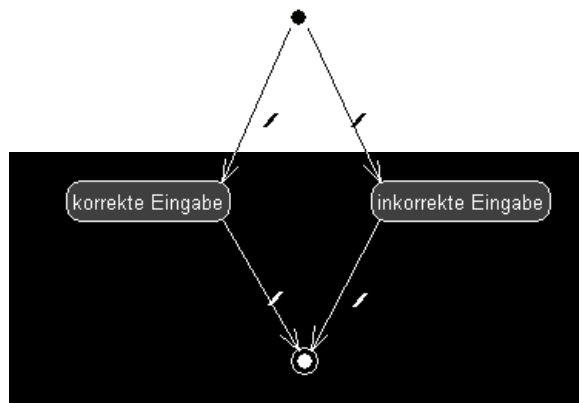
22



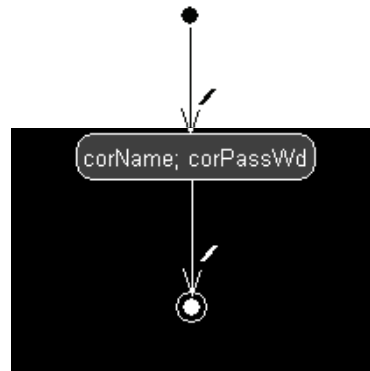
Test Lichtschalter



Test Login-Vorgang

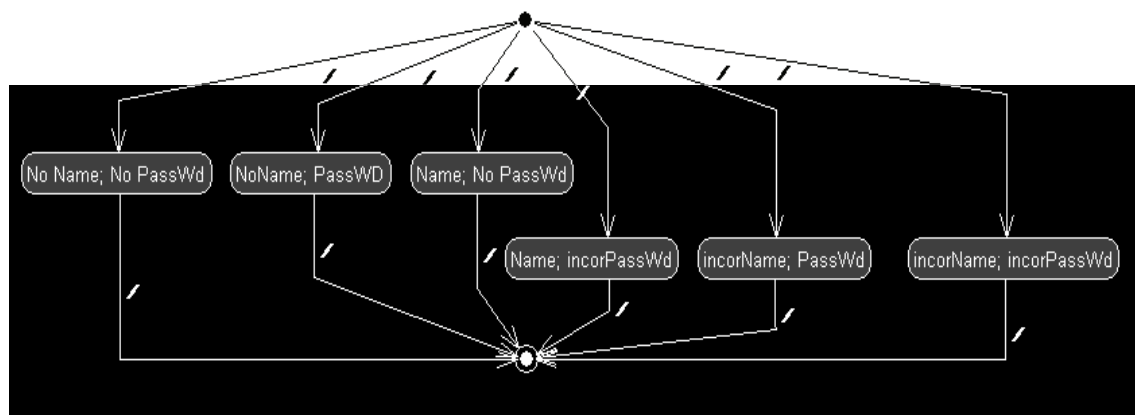


Test korrekter Login



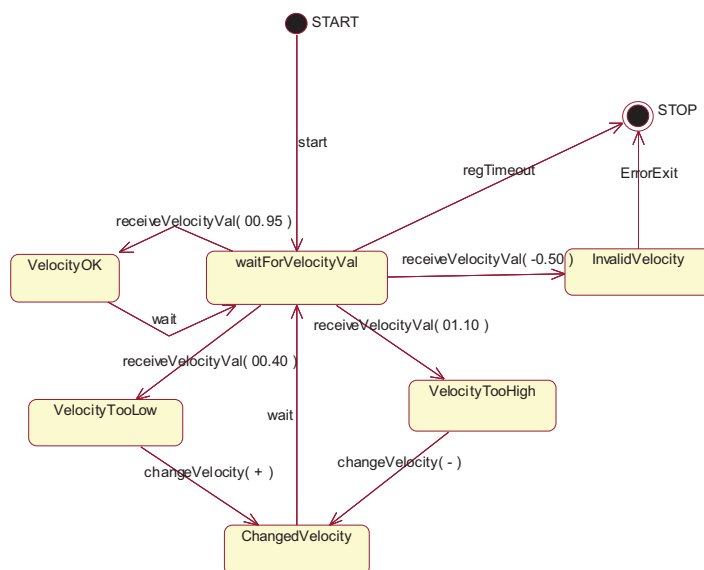
25

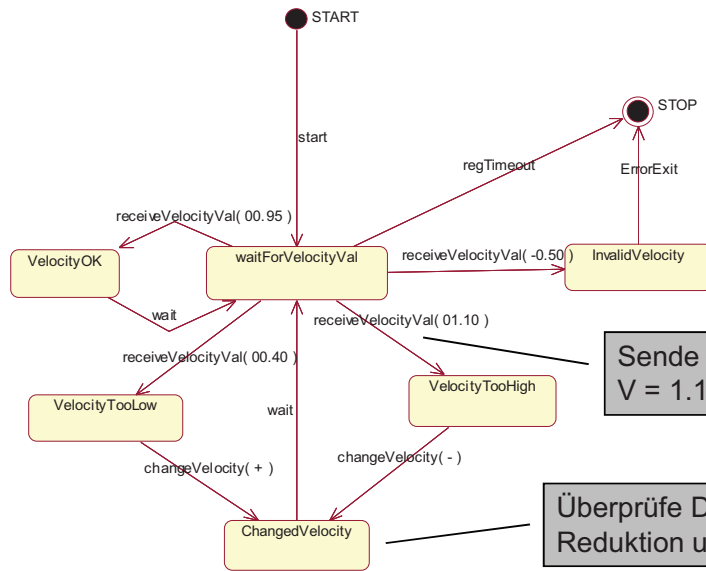
Test inkorrektter Login



26

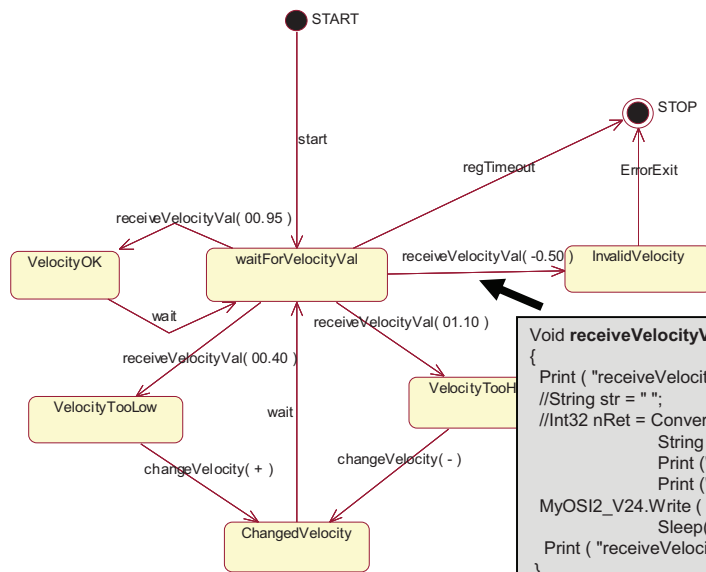
- Motivation
- .mzT-Konzept
- Einsatzgebiete
- Beispiel
- Zusammenfassung
- Diskussion



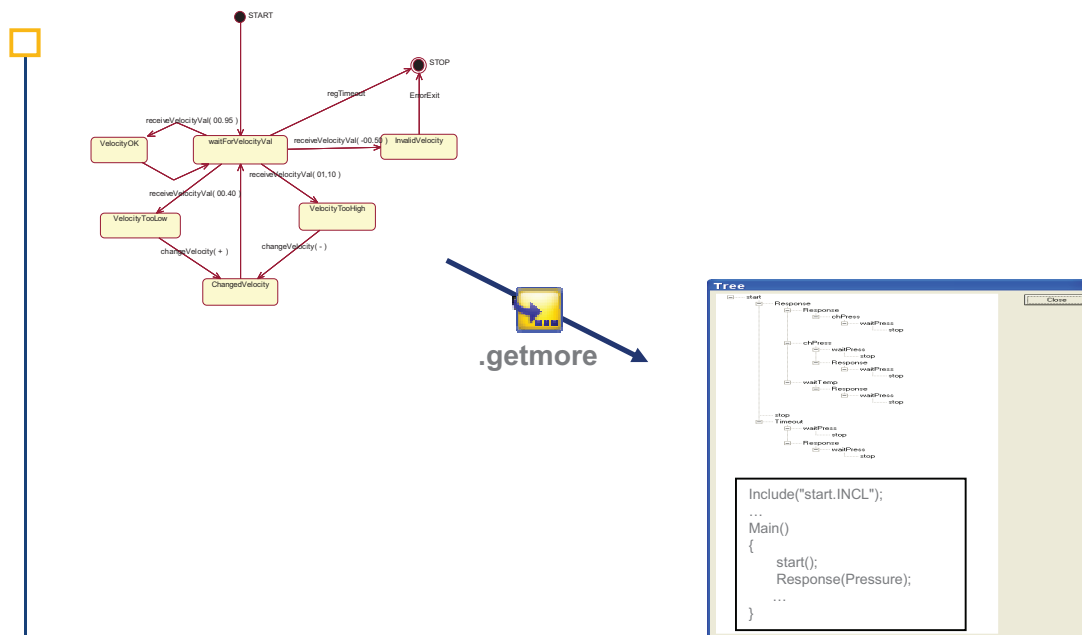


Sende Vel.-Telegramm:
V = 1.10 m/sek

Überprüfe Drehzahl:
Reduktion um 10 U/min



```
Void receiveVelocityVal(String str)
{
    Print ("receiveVelocityVal>> START");
    //String str = " ";
    //Int32 nRet = Convert ( val, str ); // Convert Float to String
    String v24Str = "TO20.0C:V" + str + "MS";
    Print ("receiveVelocityVal>> inputStr: ", str);
    Print ("receiveVelocityVal>> outStr: ", v24Str);
    MyOSI2_V24.Write ( v24Str + "|MJ");
    Sleep(10);
    Print ( "receiveVelocityVal>> STOP");
}
```



- systematische Generierung der Testfälle
 - volle Pfadüberdeckung
 - volle Übergangsüberdeckung
 - Morkovketten
- Steuermechanismen zu Begrenzung der Testfälle
 - Prioritäten
 - Guard-Conditions
 - Begrenzung der Testfalllänge
 - Begrenzung von Schleifendurchläufe



Die Methode des modellzentrierten Tests

- fasst alle testrelevanten Informationen im Modell zusammen
- visualisiert das zu testende System
- systematisiert das Auffinden von Testfällen
- automatisiert die Erstellung von konkreten Testfällen
- liefert eine Möglichkeit quantitative Aussagen über die Testabdeckung zu treffen



- Motivation
- .mzT-Konzept
- **Einsatzgebiete**
- Beispiel
- Zusammenfassung
- Diskussion



Einsatzgebiete



- Blackbox – Test
 - Schnittstellentest
 - Integrationstest
 - Funktionstest
 - Systemtest
 - Kundenabnahmetest

35



Einsatzgebiete



- Testtypen
 - datengetriebener Test
 - Beispiel: Datumsfunktion
 - funktionaler Test
 - Beispiel: Start/Stopp Funktion
 - Workflow Test
 - Beispiel: Service Funktion

36



Übersicht

- ■ Motivation
- .mzT-Beschreibung
- Einsatzgebiete
- **Beispiel**
- Zusammenfassung
- Diskussion



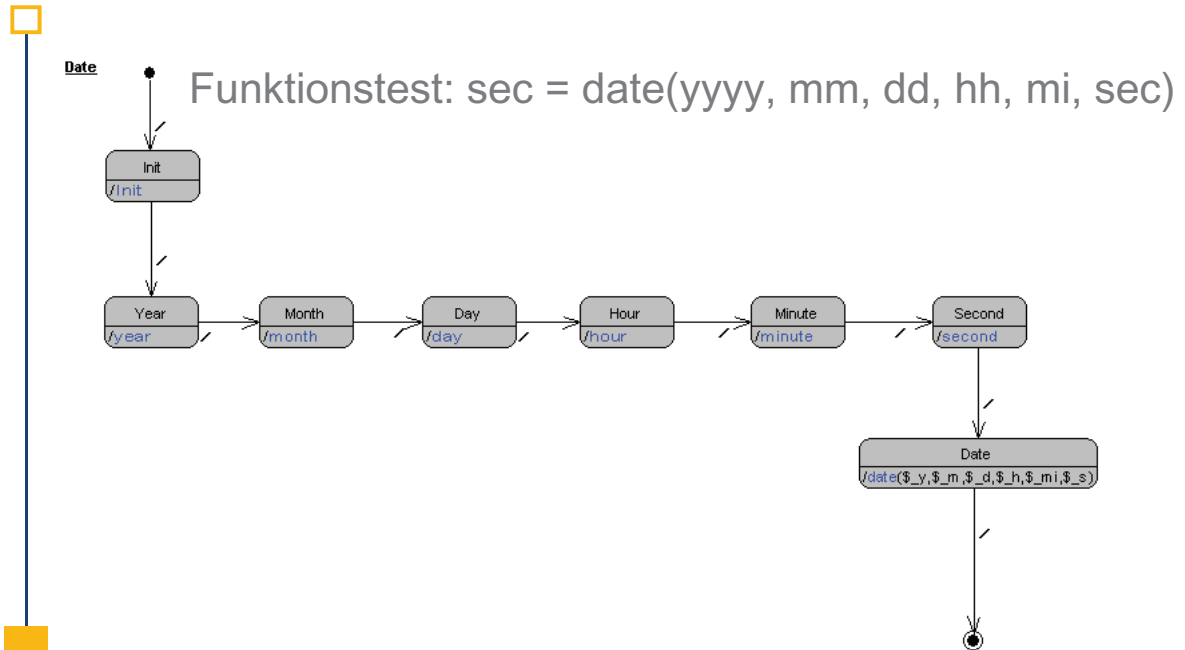
Beispiel: Datengetriebener Test

- Funktionstest: `sec = date(yyyy, mm, dd, hh, mi, sec)`

Gültigkeitsbereich: 01.01.1970 – 31.12.3000

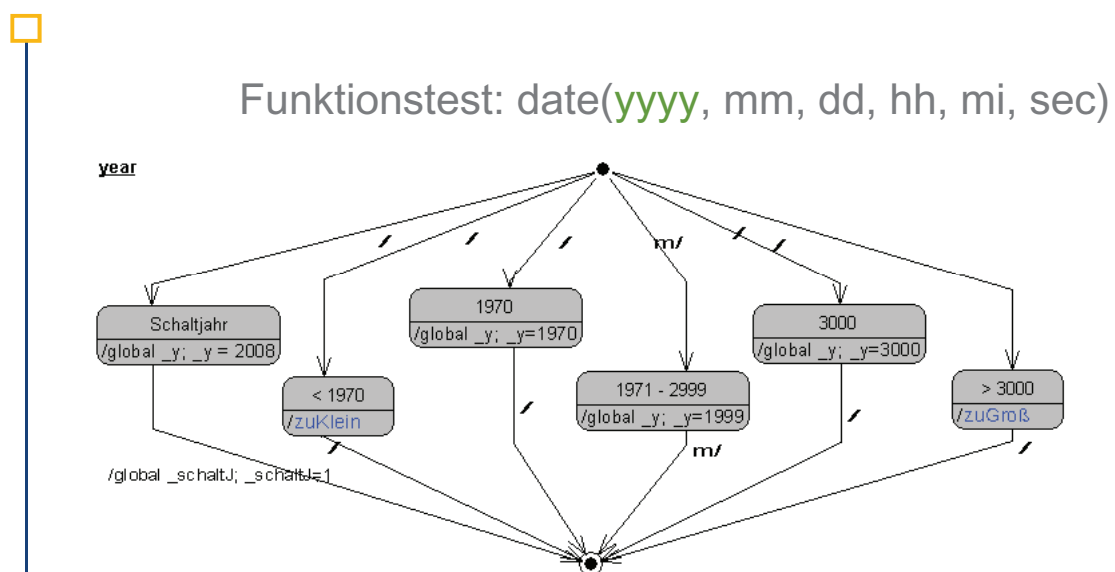


Beispiel: Datengetriebener Test



39

Beispiel: Datengetriebener Test

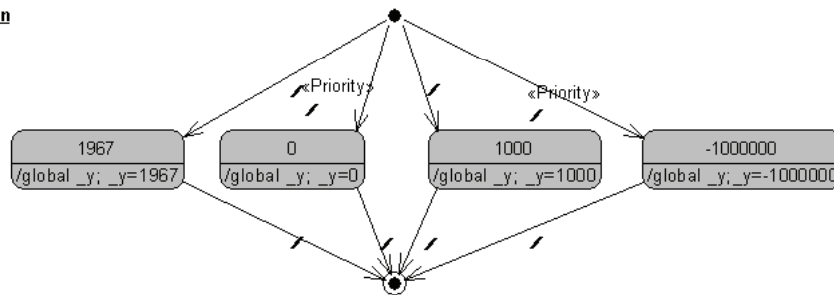


40

Beispiel: Datengetriebener Test

Funktionstest: date(yyyy, mm, dd, hh, mi, sec)

zuKlein

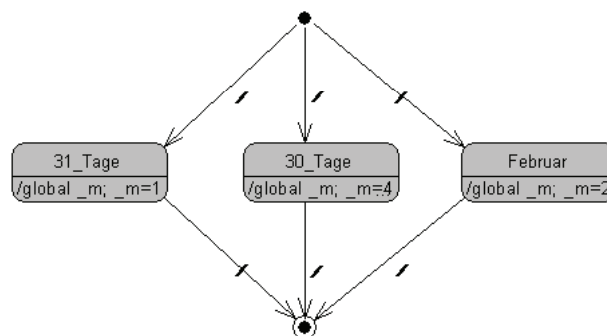


41

Beispiel: Datengetriebener Test

Funktionstest: date(yyyy, mm, dd, hh, mi, sec)

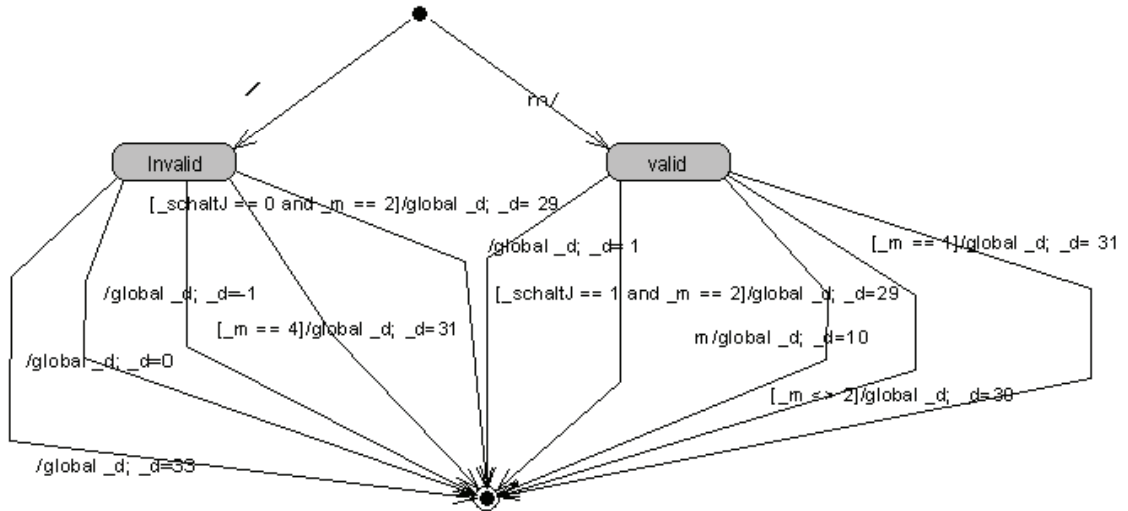
month



42

Beispiel: Datengetriebener Test

Funktionstest: date(yyyy, mm, dd, hh, mi, sec)

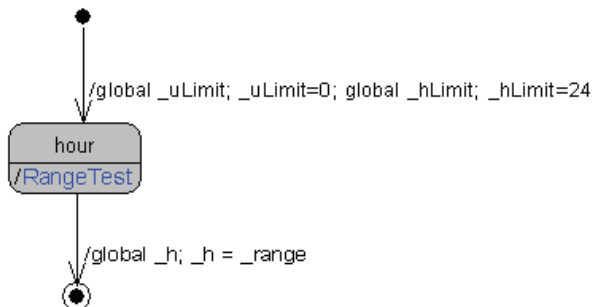


43

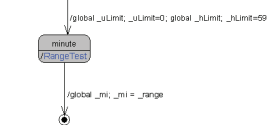
Beispiel: Datengetriebener Test

Funktionstest: date(yyyy, mm, dd, hh, mi, sec)

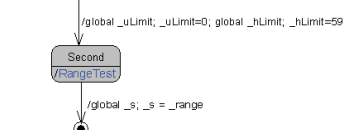
hour



minute



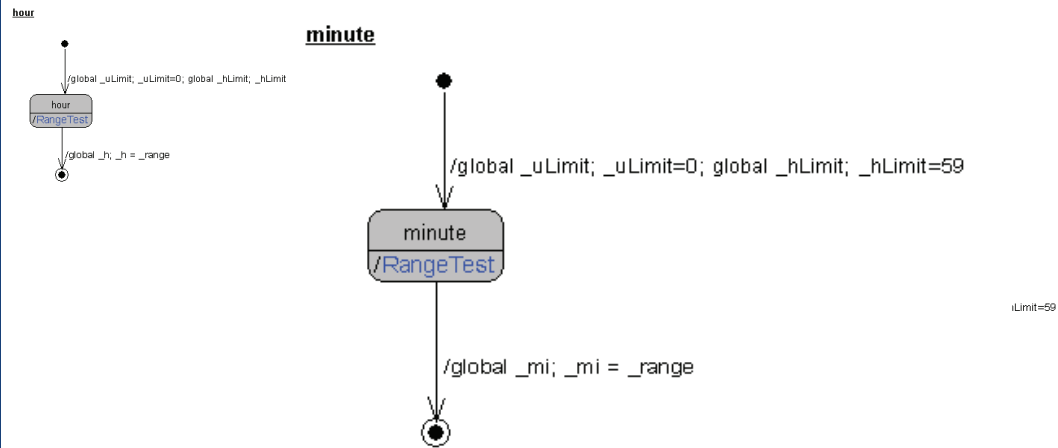
second



44

Beispiel: Datengetriebener Test

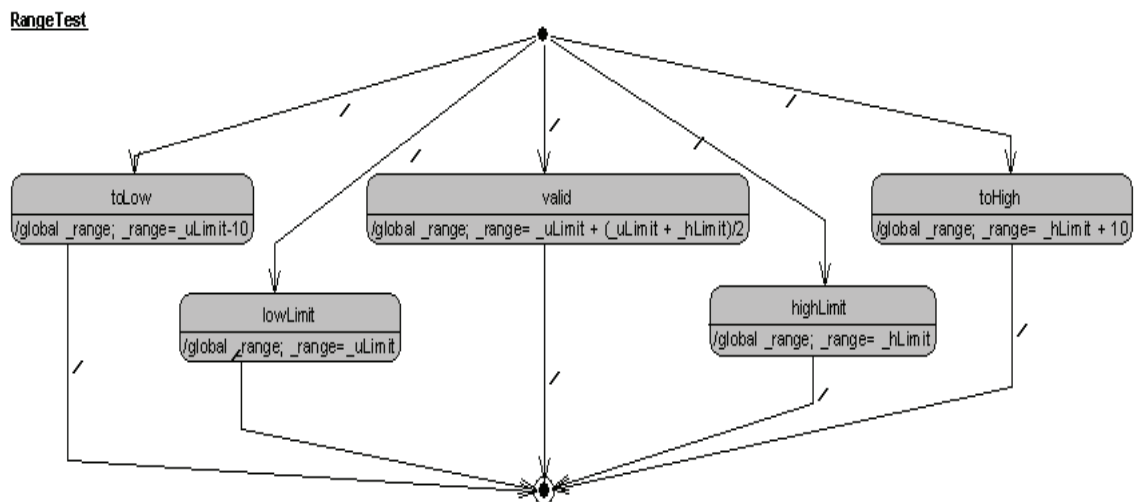
Funktionstest: date(yyyy, mm, dd, hh, mi, sec)



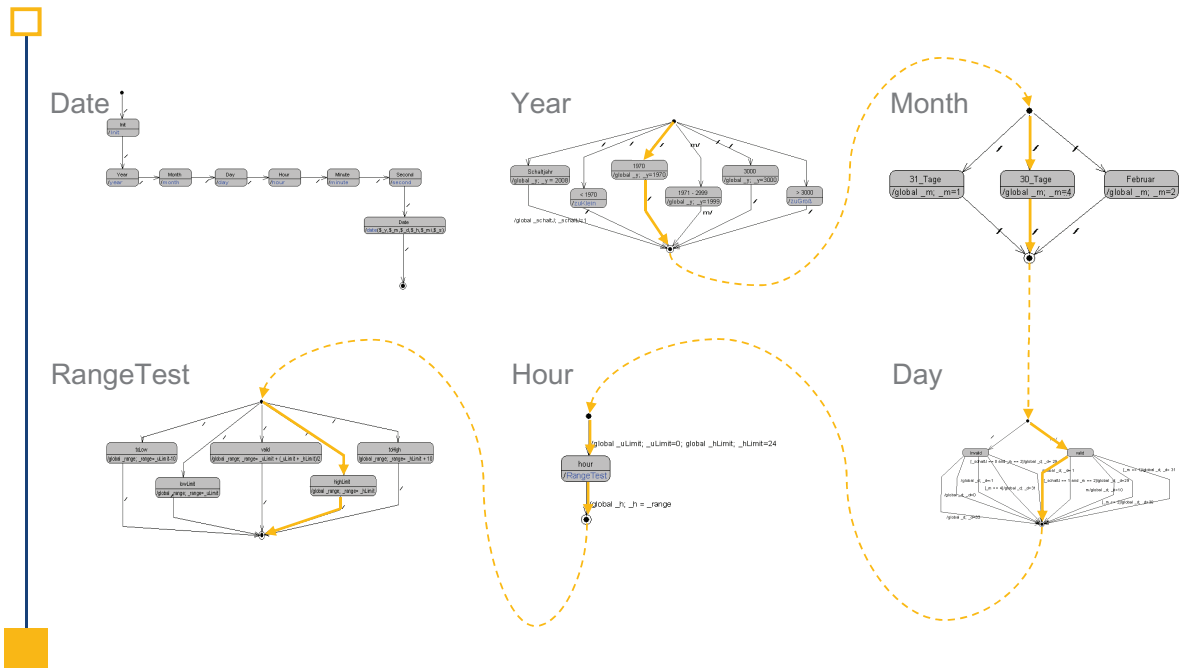
45

Beispiel: Datengetriebener Test

Subdiagram: Grenzwerttest



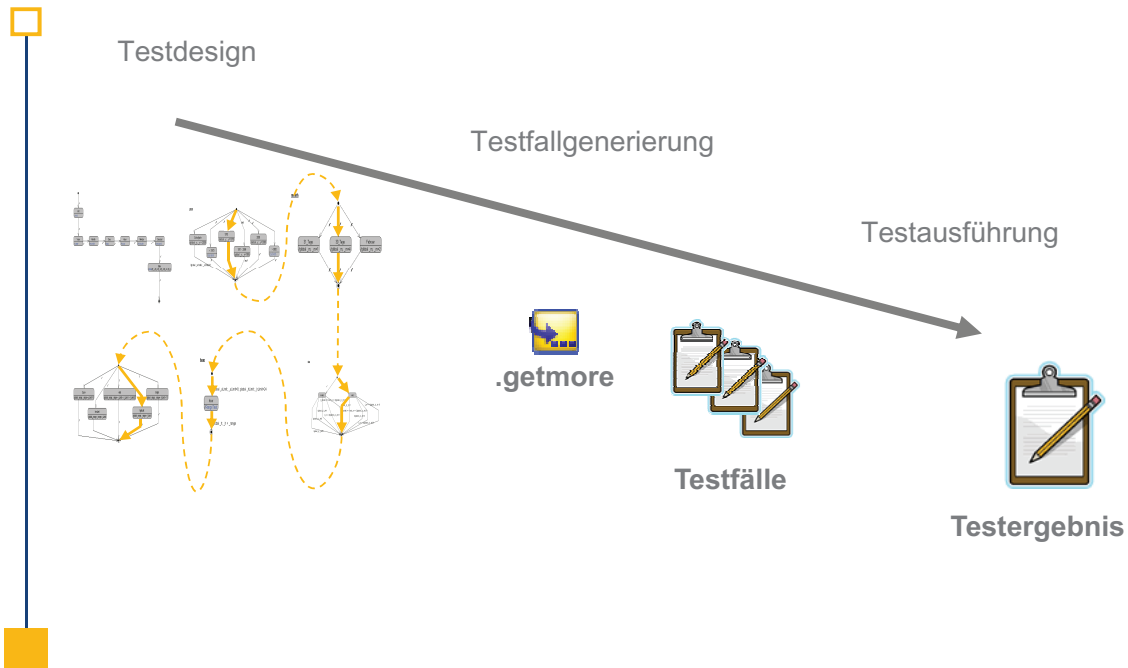
46



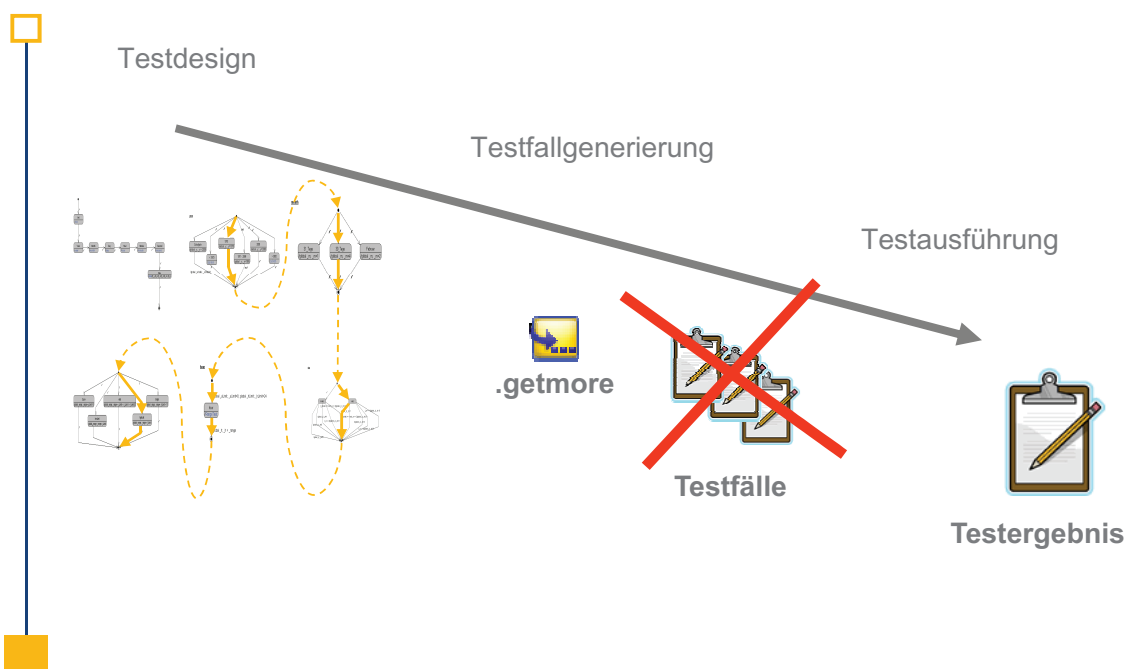
Übersicht

- Motivation
- .mzT-Beschreibung
- Einsatzgebiete
- Beispiel
- **Zusammenfassung**
- Diskussion

Die Vision von .mzT



Die Vision von .mzT



- Mit einer Kombination von Modellierung, automatischer Testfallgenerierung und automatischer Testdurchführung kann es gelingen, dem Anwender keine Gelegenheit zu geben, eine ungetestete Aktion durchzuführen.

