



Karl Kübler, Gerhard Krebser und Alexander Verl

Zusammenfassung

Um hochwertige Automatisierungslösungen mit kontinuierlich steigendem Softwareanteil anzubieten, sieht sich der Maschinen- und Anlagenbau der Herausforderung von steigendem Testaufwand und steigender Testkomplexität gegenüber. Der in einer Virtuellen Inbetriebnahme eingesetzte Digitale Zwilling bildet in diesem Beitrag die Basis für die Entwicklung einer Effizienzmethode, um den Testaufwand durch Automatisierung zu reduzieren und die Testkomplexität beherrschbar zu machen. Dieser Beitrag erarbeitet die aktuellen Defizite der Virtuellen Inbetriebnahme im Softwaretest des Maschinen- und Anlagenbaus und präsentiert eine Lösung zur Testautomatisierung. Das entworfene Testautomatisierungswerkzeug ermöglicht mit einer methodischen Testplanung und -durchführung die frühzeitige Behebung etwaiger Softwarefehler. Damit steigt die Qualität der Software deutlich und gleichzeitig wird der Aufwand der Testabläufe reduziert. Der Umfang der Testlösung wird beispielhaft an einem Bearbeitungszentrum aufgezeigt.

K. Kübler · A. Verl (✉)

Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW) der Universität Stuttgart, Stuttgart, Deutschland

E-Mail: alexander.verl@isw.uni-stuttgart.de

G. Krebser

ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Deutschland

© Der/die Autor(en) 2024

A. Verl, S. Röck und C. Scheifele (Hrsg.),

Echtzeitsimulation in der Produktionsautomatisierung,

https://doi.org/10.1007/978-3-662-66217-5_12

12.1 Einleitung

Im Engineering wird der Digitale Zwilling einer Maschine oder Anlage im Rahmen der Virtuellen Inbetriebnahme (VIBN) verwendet, um das Gesamtsystem schrittweise in Betrieb zu nehmen. Ziel bei der VIBN ist es, eine vorgezogene Validierung, Fehlerfindung und -behebung, sowie Optimierung durchzuführen [1], damit die Inbetriebnahme der realen Maschine oder Anlage schneller und reibungsfrei durchgeführt werden kann.

Bei der Fehlerfindung und -behebung konzentriert sich die VIBN im derzeitigen Einsatz im Maschinen- und Anlagenbau meist auf das Steuerungssystem und die dort ablaufenden Steuerungsprogramme. Dabei erkannte Fehler können in der VIBN-Phase im Vergleich zu einer realen Inbetriebnahme weitaus aufwandsarmer behoben werden. Die VIBN ist ein im Engineering integrierter Prozess, welcher mit der Erstellung eines Simulationsmodells des realen Zwillings beginnt. Mit dem Aufkommen der VIBN-Simulationstechnik für den Maschinen- und Anlagenbau, konnte der sequentielle Ablauf, nach dem 9-Phasen-Modell [2] (später auch Wasserfall-Modell), im Engineering aufgebrochen und ein teilweise parallelisierter Entwicklungsprozess erreicht werden. Die Teil-Parallelisierung des Entwicklungsprozesses mittels VIBN gehört heute, bei immer mehr Unternehmen aus dem Maschinen- und Anlagenbau zum Stand der Technik [3, 4]. Dabei werden häufig sogenannte Hardware-in-the-Loop Simulationen (HiLS) [5] benutzt, um Tests an realen Steuerungen in Verbindung mit einem virtuellen Modell der Maschine beziehungsweise Anlage durchführen zu können.

Die Steuerungsprogramme von Maschinen und Anlagen mittels Simulation bereits vor der Inbetriebnahme zu testen und Fehler zu korrigieren, wird zunehmend als ein maßgeblicher Wettbewerbsvorteil zwischen Unternehmen, welche in einem Konkurrenzverhältnis stehen, gesehen. Nicht zuletzt aufgrund von Maschinenführern, die, dank Technologien wie Smartphones, immer offener gegenüber intuitiven Bedienkonzepten und anpassbaren Oberflächen in ihren Anlagen sind, wird die Systemarchitektur der Steuerungssoftware moderner Anlagen zunehmend komplexer und heterogener. Sie besteht häufig aus verschiedenen Kernbestandteilen: HMI (Human Machine Interface, Benutzerschnittstelle), SPS (speicherprogrammierbare Steuerung) und NC-Kern (numerischer Steuerungskern). Da es zwischen den verschiedenen Kernbestandteilen, sowie der Anlage und der Leitebene im Unternehmen unterschiedliche Schnittstellen gibt, steigt die Komplexität der Steuerungssysteme weiter an.

Bei steigender Funktionalität und Komplexität von Maschinen und Anlagen, und damit auch von SPS-Programmcode, sind heutzutage mehr als zwei Drittel des Programmcodes nur zur Gewährleistung der Funktionsfähigkeit und Sicherheit notwendig [6]. Werden diese Mechanismen nicht ausreichend vor dem produktiven Einsatz der Anlage getestet, kann es zu unvorhersehbaren und oft teuren Folgeschäden oder Verzögerungen in der laufenden Produktion kommen. Die stetig zunehmenden Testumfänge steigern kontinuierlich die Kosten in der Qualitätssicherung und binden Personal.

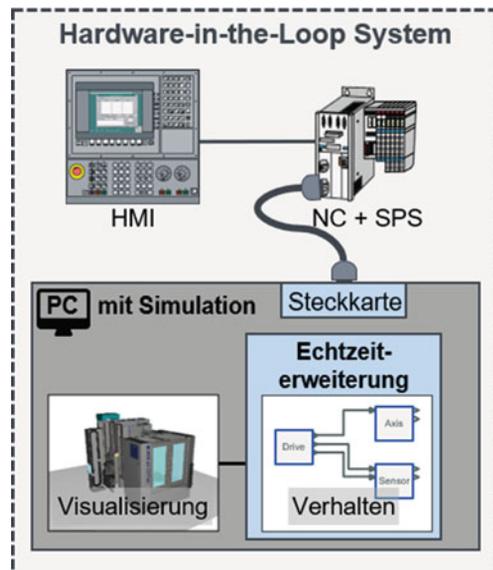
Die folgenden Kapitel zeigen exemplarisch für Werkzeugmaschinen, wie der Test im Rahmen der VIBN verbessert werden kann. Die dabei identifizierten Problemstellungen, sowie die präsentierte Lösung können auch auf andere automatisierte Produktionssysteme übertragen werden.

12.2 Stand der Technik

Um zu überprüfen, ob ein Steuerungsprogramm tatsächlich im gewünschten Sinne funktioniert, sind Tests notwendig. Der wichtigste Teil der Testaktivitäten findet auch heute noch vielfach erst während der Integration und Inbetriebnahme an der realen Maschine statt. Es geht dabei darum, die Reaktion der Steuerungsprogramme zu bewerten. Viele Unternehmen wünschen sich jedoch, früher mit der Testphase beginnen zu können. Um das zu ermöglichen, setzen immer mehr Firmen auf die VIBN [7, 8]. Im Rahmen der VIBN wird der Test entwicklungsbegleitend und damit früher durch die Nutzung einer virtuellen Umgebung ermöglicht [9]. Abb. 12.1 zeigt eine für die VIBN verwendete HiLS. Das System besteht aus einer realen Steuerung, einem realen Kommunikationssystem und einem Simulationsrechner [9]. In dem speziell betrachteten Fall handelt es sich um eine Computerized Numerical Control (CNC) als Steuerung, oben rechts in der Abbildung, und um eine Simulation einer Werkzeugmaschine, unten in der Abbildung. Ein CNC-System wird weiter unterteilt in: einen NC-Kern zur Geometriedatenverarbeitung für die einzelnen Achsen und eine weitere Steuerung, die SPS, zur Verarbeitung von logischen Schaltfunktionen. Als dritter Teil wird die Schnittstelle zum Benutzer gesehen, das Human Machine

Abb. 12.1

Hardware-in-the-Loop System bestehend aus Steuerung, Antriebsbus und virtueller Werkzeugmaschine, angelehnt an [3]



Interface (HMI) [10]. Das CNC-System und die simulierte Maschine bilden zusammen mit dem Antriebsbus eine komplette Werkzeugmaschine ab, wie sie in Realität in der Produktion steht.

Der PC benötigt zunächst zur Anbindung an einen industriellen Feldbus eine Steckkarte, welche mit einem speziellen Bustreiber ausgestattet ist, um eine Kommunikation in Steuerungsechtzeit zu ermöglichen. Das Betriebssystem des PCs muss ebenfalls echtzeitfähig sein, sodass die Simulation deterministisch Ergebnisse an die Steuerung liefern kann. Die großen Vorteile einer HiLS sind [9]:

- Eine saubere Trennung zwischen Simulation und Automatisierung, da die Steuerung mit unverändertem Steuerungscode und -konfiguration zum Einsatz kommt
- Umfangreiche Testmöglichkeiten (Integrationstests) mitsamt Peripherie der Steuerung zur Durchführung auch von komplexen Testfällen mit hoher Übertragbarkeit auf den realen Zwilling

Zum Testen von SPS-Programmcode von Werkzeugmaschinen sind HiLS besonders geeignet, da andere Architekturen, bspw. Software-in-the-Loop, Einrechner-Simulation und auch Simulation auf der Steuerung, siehe [9], keine Abbildung des Bussystems ermöglichen. Damit geht im Vergleich zur realen Werkzeugmaschine Realitätsnähe verloren was zu eingeschränkten Testmöglichkeiten führt.

Unter Verwendung einer HiLS ist es möglich, das Gesamtsystem im Entwicklungsprozess schon deutlich früher als bei Tests an der realen Werkzeugmaschine sowohl einem Gut- als auch einem Schlechtfalltest zu unterziehen [11]. Unabdingbare Abnahmetests an der realen Anlage beim Hersteller (Factory Acceptance Test) beziehungsweise beim Kunden (Site Acceptance Test) lassen sich nach einer VIBN wesentlich schneller abschließen [8]. Während die Gutfalltests darauf abzielen, die vom Kunden gewünschte Performance und Funktionalität zu überprüfen, werden bei den Schlechtfalltests Zuverlässigkeit und Robustheit der Software getestet, indem die Reaktion der Steuerungsprogramme in expliziten Gefahren- und Störsituationen überprüft wird. Die gewonnene Zeit nach Durchführung einer VIBN kann für Optimierungen oder zum früheren Ausliefern der Maschine genutzt werden.

12.3 Befragung von Maschinenherstellern

Eine eigens durchgeführte Befragung von vier führenden Werkzeugmaschinenherstellern aus Deutschland zeigt die vorherrschenden Mängel beim manuellen Test unter Einsatz einer HiLS auf. Alle befragten Unternehmen haben die VIBN in ihren Entwicklungsprozessen im Einsatz.

Insgesamt wurden sechs Abteilungen aus vier Unternehmen gebeten, einen Fragebogen zum Ist-Stand der VIBN und Tests auszufüllen. Dabei wurden 13 Fragen zu den Bereichen Testablauf, Testmanagement (Definition [13]) und Testauswertung gestellt. Im Nachgang zu den Fragebögen wurden Interviews mit den Abteilungen geführt, um die Antworten zu vertiefen. Durch die Befragungen wurden folgende Defizite sichtbar:

1. Ein Testmanagement fehlt
2. Die Testdauer ist sehr hoch, bis zu mehreren Tagen
3. Die VIBN ist eine aufwändige, manuelle Tätigkeit
4. Die Frequenz und Anzahl an Testabläufen steigen

Zu 1.: Die Befragten gaben an, dass Tests weder umfassend geplant noch umfassend dokumentiert würden. Zum einen sei das auf eine knapp bemessene Anzahl an Testern zurückzuführen, zum anderen darauf, dass viele Tests in Eigenregie durchgeführt würden und Rückmeldungen nicht einheitlich erfolgten. Weiterhin würden erstellte Testfälle nicht für eine Wiederverwendung konzipiert, sondern spezifisch auf den aktuellen Testablauf ausgelegt. Eine Einplanung von Zeitfenstern für den Test von Funktionalitäten erfolge nicht, es würde die übrige Zeit in der Entwicklungsphase für Tests genutzt, gaben über 60 % der Befragten an.

Zu 2.: Tests von Steuerungsprogrammen, welche früher erst bei der realen Inbetriebnahme (IBN) gemacht werden konnten, werden durch die VIBN vorgezogen. Die Zeitdauer der IBN selbst habe sich im Gegensatz dazu aber nicht verändert, so die Hersteller. Im Gegenteil, nun stünde als zusätzlicher Arbeitsaufwand die Erstellung des virtuellen Modells für die Simulation an. Da die Zeit für die Erstellung immer noch gering im Vergleich zur gesparten Zeit am Ende des Projekts sei, würden virtuelle Werkzeugmaschinen überhaupt modelliert und bei der VIBN eingesetzt. Die Testdauer, welche für die VIBN benötigt wird, sei aber weiterhin sehr lang, im Kontext der gesamten Projektdauer. Besonders hervorgehoben wurde von den befragten Personen, dass die Testschritte, welche in einem Testfall auszuführen seien, nicht den Hauptteil an der Zeitdauer ausmachten. Zum Testablauf gehöre auch immer, das System in einen definierten Ausgangszustand zu bringen, wie die befragten Personen erklärten. Diese Tätigkeit sei zum Teil zeitaufwändiger als der eigentliche Test einer Funktionalität. Einfache Testabläufe, gemessen ohne die Testvorbereitung, seien nach zwei Minuten, komplexere Testabläufe nach mehreren Tagen beendet.

Zu 3.: Fünf von sechs Abteilungen gaben an, ihre Tests manuell durchzuführen. Auch die VIBN binde Tester als Ressource während des gesamten Testablaufs. Dazu gehöre die Testerstellung, der Testablauf und die Testprotokollierung sowie -auswertung. Bei der Testerstellung würde sehr häufig nicht auf bestehende Testfälle zurückgegriffen, sondern neue erstellt, so die befragten Personen. Es existierten keinerlei Systeme in den Abteilungen, welche eine Wiederverwendbarkeit unterstützten, gaben vier von sechs Abteilungen an. Bei der Testprotokollierung und -auswertung würden unzureichende Aufschriebe erstellt. Die Hersteller testen nach Gefühl und notierten teilweise nur „i. O.“ (in Ordnung) oder „n. i. O.“ (nicht in Ordnung) am Ende des Testablaufs.

Zu 4.: Zu einem Anstieg an Testaufkommen für die Werkzeugmaschinenhersteller würden auch die Softwareupdates der Steuerungshersteller führen. Dies konnten alle Maschinenhersteller berichten, welche ihre Steuerung nicht selbst entwickeln. Nach Updates an der Steuerung müssten alle SPS-Programme für die neue zugrundeliegende Steuerungssoftware, in sogenannten **Regressionstests**, freigegeben werden. Insbesondere von den Maschinenbauern programmierte (Mess-)Zyklen seien davon betroffen. Nach Aussage eines Werkzeugmaschinenherstellers kommt es im Zuge einer neuen Politik bei den Releases von Softwareständen zur Erhöhung der Frequenz an Softwaretests. Dabei geht dieser Hersteller den Weg eines Softwareunternehmens und stellt regelmäßige und verbesserte Versionen seiner Steuerungsprogramme zur Verfügung. Jedes neue Release müsse im Zuge der Qualitätssicherung alle definierten Tests zur Abnahme bestehen. Dies erhöhe den Aufwand enorm und führe auch zu wenig herausfordernden Testabläufen, da immer wieder die gleichen Testabläufe ausgeführt würden.

12.4 Probleme der VIBN und abgeleiteter Handlungsbedarf

Es lässt sich aus den Befragungen feststellen, dass die Firmen einen Nachholbedarf im **Softwaretesting** haben. Zu diesem Schluss kam auch eine frühere Befragung im deutschen Maschinen- und Anlagenbau [14]. Aus der Entwicklung von Anwendersoftware in höheren Programmiersprachen bekannte Konzepte wie Wiederverwendbarkeit, Testprotokollierung, Testabdeckung, Testautomatisierung, beschrieben in [13], finden bei der Entwicklung von Steuerungsprogrammen kaum bis keine Anwendung [14]. Hinzu kommt, dass sich in der aktuellen Struktur der VIBN viel Potenzial zum Einsparen von Zeit- und Personalkosten findet. Weiterhin bleibt die VIBN:

- **Zeitaufwändig**, ein Tester muss die HiLS manuell bedienen
- **Fehleranfällig**, die Tests werden manuell ausgeführt
- **Unzureichend**, der Tester kann nur testen, was er auch sieht
- **Kostenintensiv**, durch hohen Personal- und Zeitaufwand

Ausführlicher beschrieben heißt das: Aus den Interviews ging hervor, dass der Prozess der VIBN mit anschließender IBN im Vergleich zur rein realen IBN keine absolute Zeitersparnis bewirkt. Die benötigte Zeitdauer einer Testausführung bleibt gleich hoch und bindet das Personal. Die zeitliche Entzerrung auf dem kritischen Entwicklungspfad entsteht durch die Parallelisierung der Tätigkeiten. Darüber hinaus lässt es sich nicht vermeiden, dass aufgrund der Modularität und der Konfigurierbarkeit der Werkzeugmaschinen hohe Wiederholungsraten einzelner Tests entstehen.

In einer HiLS, welche in harter Echtzeit arbeitet, können beim manuellen Test nicht alle Signale erfasst und ausgewertet werden. Solange die Tests manuell ausgeführt werden, kann es zudem zu Fehlern aufgrund menschlichen Versagens kommen. Beispielsweise kann eine manuelle Testausführung keine zeitlich deterministische Reproduzierbarkeit gewährleistet werden. Durch falsche Interpretation eines Testablaufs und Unachtsamkeit bei der Beobachtung von Zuständen können ebenfalls falsche Testergebnisse entstehen. Diese Sachverhalte führen im besten Fall nur zu einer Wiederholung des Tests (Mehraufwand), in anderen Fällen zu einem abgenommenen falschen Testergebnis.

Nicht zuletzt ergibt sich eine weitere Komplikation: Unternehmen müssen baugleiche Maschinen mit Steuerungen verschiedener Hersteller anbieten. Dementsprechend sind vorab jegliche Kombinationen aus Maschine und Steuerung zu testen.

Das nachfolgende Kapitel beschäftigt sich mit einem Lösungsansatz, welcher die oben genannten Probleme löst. Das Hauptaugenmerk liegt darauf, ein automatisiertes Testsystem zu schaffen, welches manuelles Testen weitgehend ersetzt.

12.5 Anforderungen und Lösungsansatz

Um die in Abschn. 12.4 dargestellten Probleme zu lösen, wurde am ISW der Universität Stuttgart in Zusammenarbeit mit der ISG Industrielle Steuerungstechnik GmbH ein Konzept erarbeitet, welches erlaubt, Testabläufe, wie sie bei der VIBN auftreten, zu automatisieren. Das Konzept trägt den Namen **Virtuelle Steuerungstestbench** und ist in Abb. 12.2 abgebildet. Im Jahr 2018 ist daraus das kommerziell verfügbare Testautomatisierungswerkzeug **ISG-dirigent** entstanden. Als Testumgebung wird die aus der VIBN bekannte HiLS genutzt, um die Werkzeugmaschine virtuell abzubilden. Mit der HiLS ist ein Testautomatisierungswerkzeug (TAW) verbunden, welches die Aufgaben des Testers übernimmt. Insbesondere die Testausführung und Testauswertung werden aufgrund der Befragung bei den Werkzeugmaschinenherstellern für die Automatisierung vorgesehen. Die drei Schnittstellen zwischen TAW und HiLS werden im nächsten Kapitel beschrieben.

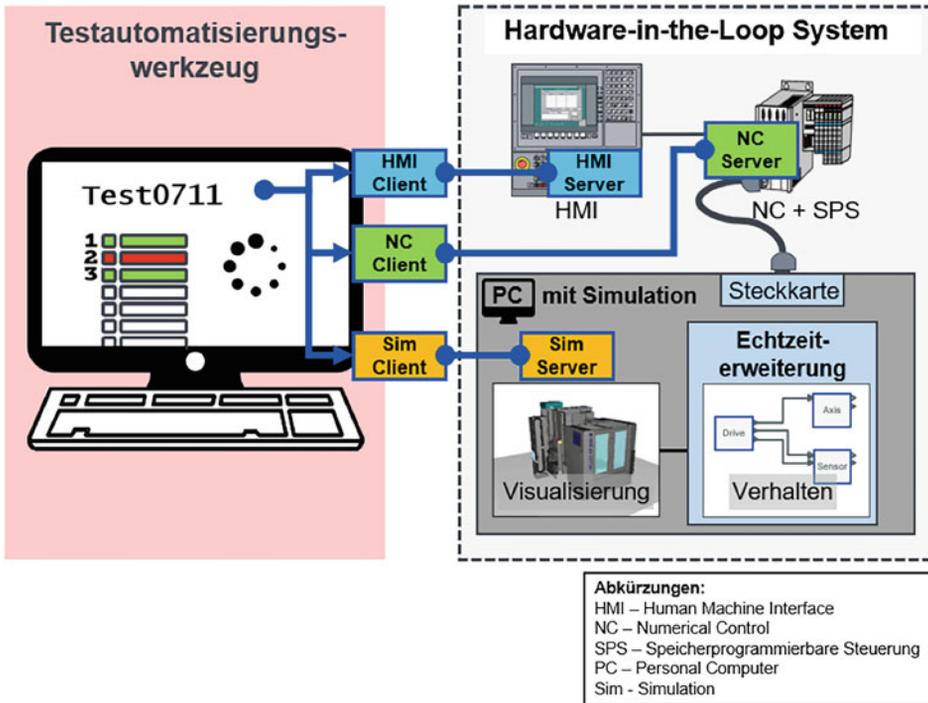


Abb. 12.2 Schema der Virtuellen Steuerungstestbench aus Testautomatisierungswerkzeug und HiLS

Für den erfolgreichen Einsatz der Virtuellen Steuerungstestbench für die VIBN werden die folgenden Anforderungen definiert:

- Damit die erstellten Testabläufe in verschiedenen Phasen der Entwicklung einer Anlage wiederverwendet werden können, muss das Testautomatisierungswerkzeug (TAW) die Definition von wiederverwendbaren Tests unterstützen.
- Die Interaktion mit dem TAW darf kein Expertenwissen hinsichtlich der Kommunikationsprotokolle o. Ä. der HiLS voraussetzen.
- Damit möglichst viele Beteiligte aus der Entwicklung unmittelbar an der Erstellung der Tests mitwirken können, dürfen für deren Erstellung keine Programmierkenntnisse notwendig sein. Die Darstellung der Testabläufe muss leicht verständlich sein. Das Werkzeug muss einfach bedienbar sein.
- Um komplexe Abläufe realitätsnah testen zu können, muss das TAW parallele Abläufe, Verzweigungen und Schleifen ermöglichen.
- Komplexe Anlagen enthalten oft mehrere Maschinen und Steuerungen. Das TAW muss daher in der Lage sein, mehrere Steuerungen in einer HiLS testen zu können.

- Um Fehlerursachen schnell und sicher lokalisieren zu können, muss der Testablauf detailliert protokolliert werden können.
- Testreports müssen eine reversionssichere Testdokumentation ermöglichen.
- Maschinen sind i. d. R. modular aufgebaut und ermöglichen dadurch eine hohe Varianz im Hinblick auf die jeweilige kundenspezifische Ausführung. Adressen von Komponenten, Parametern, Ein-/Ausgängen etc. ändern sich damit häufig abhängig von der jeweiligen Ausführung. Damit Tests diesbezüglich einfach, sicher und schnell an die jeweilige Ausführung der Maschine angepasst werden können, müssen diese symbolisch adressiert werden können.

Im Folgenden soll detaillierter auf das TAW und dessen Realisierung als Testframework eingegangen werden. Das TAW, als Kernkomponente der automatisierten VIBN, wird wie folgt spezifiziert: Eine **grafische Benutzerschnittstelle (GUI)** ermöglicht es Tätigkeiten aus dem Testmanagement auszuführen, z. B. Tests zu erstellen, Testabläufe zu starten und zu beobachten, sowie historische Testabläufe zu betrachten. In der **Testausführung** werden die grafisch definierten Tests in ausführbare Befehle übersetzt. Die Befehle beinhalten hauptsächlich Schritte für den automatisierten Zugriff auf die Schnittstellen der HiLS. Gleichzeitig protokolliert die Testausführung den Testablauf. In einer **Datenhaltung** werden wiederverwendbare Tests, sowie Testprotokolle und weitere relevante Framework-Daten abgelegt.

In Abb. 12.3 wird eine mögliche Realisierung des Testframeworks spezifiziert. Das Testframework ist primär für die Testausführung zuständig und muss dazu sekundär die Kommunikation und den Datenaustausch mit der GUI und der Datenhaltung realisieren.

Wie in Abb. 12.3 dargestellt, erzeugt der Tester durch Konfigurieren von neuen Tests die drei Dateiarnten **Test-Fall**, **Test-Szenario** und **Test-Konfiguration**. Logisch besteht eine Test-Konfiguration aus Test-Schritten, welche konkrete an einer HiLS ausführbare Aktionen beinhalten. Test-Schritte werden als atomare Bestandteile durch das Testframework mitgeliefert und können je nach Anwendungsfall neu hinzugefügt werden. Beispielsweise kann ein Test-Schritt enthalten: *Auslesen eines Steuerungsparameters aus der NC-Parametrierung*, oder *Überschreiben eines Signals im Simulationsmodell zur gewollten Fehlererzeugung*. Mehrere Test-Schritte werden zu einem Test-Fall verknüpft, welcher mehrere Test-Schritte miteinander verknüpft und auch die Schritte beinhaltet, um eine HiLS in einen definierten Ausgangszustand vor dem Ausführen der eigentlichen Test-Schritte zu bringen. Mehrere Test-Fälle werden zu Test-Szenarien gebündelt. Diese Bündelung findet z. B. funktionsweise statt, so können einzelne funktionale Einheiten einer Werkzeugmaschine bei der VIBN schrittweise separat getestet werden. Wobei das Szenario wiederum für andere Modelle des Werkzeugmaschinentyps benutzt werden kann, welche im Sinne einer modularen Maschine diese funktionale Einheit verbaut haben. Die Test-Konfiguration enthält neben mehreren Test-Szenarien zusätzliche Informationen zur HiLS. Die Beschreibungen dazu entnimmt das Framework den Dateien

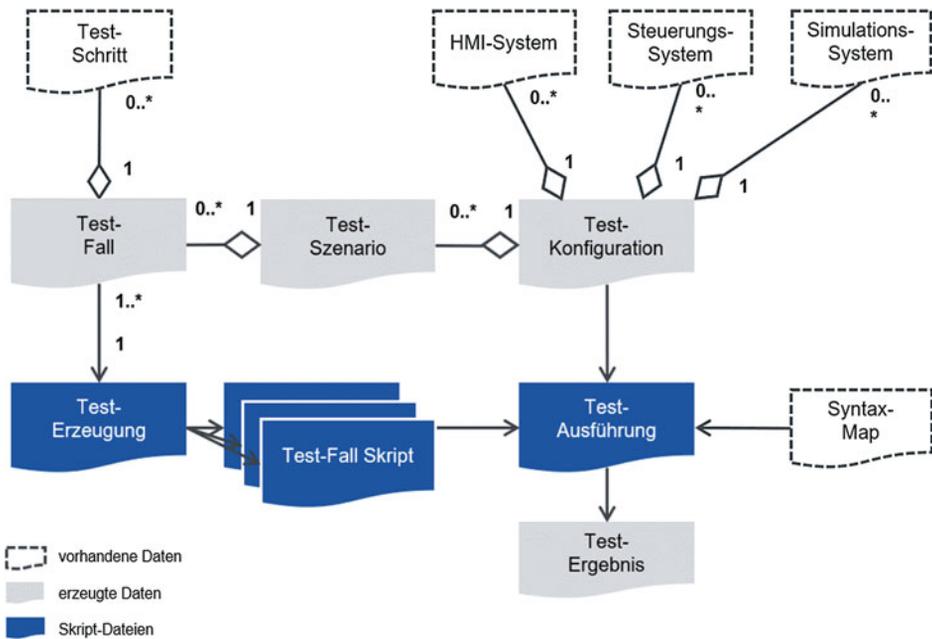


Abb. 12.3 Komponentendiagramm des Testframeworks im TAW

HMI-System, Steuerungssystem, Simulationssystem. Vor der Testausführung generiert die **Test-Erzeugung** aus den einzelnen Beschreibungen der Test-Fälle ausführbare Befehle, z. B. in Form von Skripten. Zur Laufzeit des Tests werden beim Ausführen der einzelnen Test-Fälle, durch die **Test-Ausführung**, generische Befehle der Testfälle in konkrete Syntax für das aktuelle CNC-System in der HiLS übersetzt. Als Testergebnisse werden die Resultate aus jedem Test-Schritt in eine **Test-Ergebnis** Datei abgespeichert.

12.6 Schnittstellen des Testframeworks

In diesem Kapitel wird vertieft auf die Schnittstellen zwischen TAW und der HiLS eingegangen. Zwischen TAW und HiLS sind drei Schnittstellen zu betrachten, um die Ausführung eines Tests automatisierbar zu machen. Zwei Schnittstellen bestehen zum CNC-System, wobei eine der Schnittstellen zwischen der Steuerung (NC und SPS) und TAW besteht und die zweite zwischen HMI und TAW. Die dritte Schnittstelle besteht zur Simulation der Werkzeugmaschine.

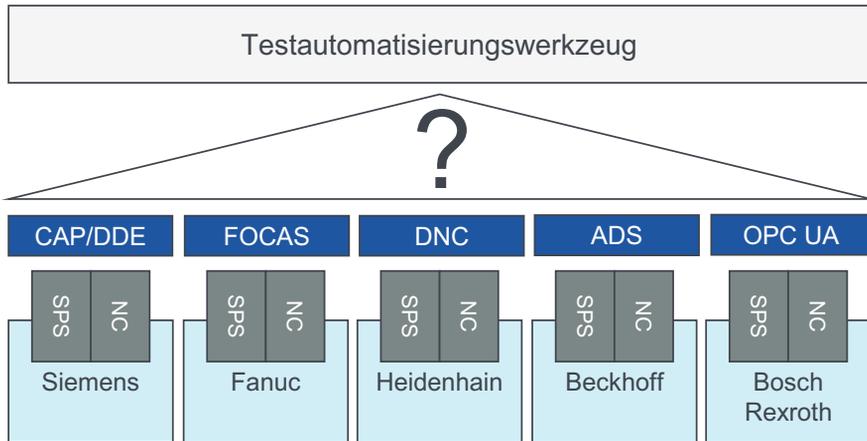


Abb. 12.4 Diversität proprietärer Schnittstellen von CNC-Steuerungen verschiedener Hersteller

Die **HMI-Schnittstelle** wird beim automatisierten Testen für zwei Dinge benötigt: Erstens, zum Auslösen von Bedienhandlungen, welche vormals von einem Tester ausgeführt wurden. Zweitens, um zu überprüfen, ob das HMI im Fehlerfall richtige Fehlermeldungen an den Benutzer weitergibt.

Über die **Steuerungsschnittstelle** werden Parameter und Variablen der SPS beziehungsweise der NC ausgelesen. Das gleiche Werkzeugmaschinenmodell wird häufig je nach Kundenwunsch mit einer unterschiedlichen Steuerung ausgestattet. Die Steuerungsschnittstelle muss deshalb herstellerunabhängig ausgeführt sein. Damit soll es ermöglicht werden, dass dieselben Tests unabhängig vom jeweiligen Steuerungssystem verwendet werden können. Abb. 12.4 zeigt die Diversität der Schnittstellen (z. B. ADS, OPC UA) anhand der gängigsten CNC-Systeme.

Zur Definition von herstellerunabhängigen Tests muss zusätzlich zur Verbindung eine herstellerunabhängige Adressierung von Parametern und Variablen in der Steuerung möglich sein. Mit einer sogenannten Syntax-Map werden die Adressen der Parameter und Variablen in einer Steuerung über mehrere Hersteller hinweg, für jeweils einen bestimmten Parameter oder eine bestimmte Variable mit einem generischen Befehl verknüpft. Abb. 12.5 zeigt die schematische Darstellung der umgesetzten Lösung.

Beim Konfigurieren von Testfällen benutzt der Tester den generischen Befehl und gibt in der Test-Konfiguration das Zielsystem an. Somit bleibt der einzelne Testfall generisch und kann ohne Anpassung in einer weiteren Test-Konfiguration auf einem weiteren Zielsystem ausgeführt werden. Abb. 12.5 zeigt exemplarisch den Aufruf für den Befehl „Aktueller Status NC-Kanal 1“. Dabei wird der generische Befehl „MachineInterface.CncChannelList.1.ActStatus“ benutzt. Als Zielsteuerung ist eine Fanuc CNC im Einsatz. Durch Nachsehen in der Syntax-Map wird aus dem generischen Befehl die Syntax von Fanuc ermittelt „cnc_statinfo“. Bei der Ausführung des Testfalls wird dann

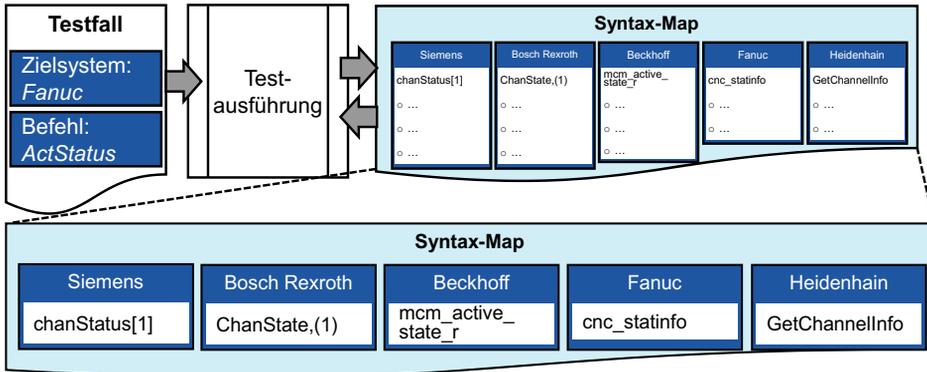


Abb. 12.5 Schema und Funktionsweise der Syntax-Map im Testframework

die Syntax von Fanuc anstatt des generischen Ausdrucks verwendet. Die Syntax-Map kann jederzeit um einen Hersteller oder eine Adressierung für Parameter und Variablen erweitert werden.

Die dritte Schnittstelle ist die **Simulationsschnittstelle** zur virtuellen Werkzeugmaschine. Mithilfe der Simulationsschnittstelle können Werte in der virtuellen Maschine gelesen und geschrieben werden. Durch das Lesen kann der Zustand von Komponenten der virtuellen Werkzeugmaschine bestimmt werden und mit dem Zustand in der Steuerung verglichen werden. Abfolgen wie „Setzen an der Steuerung und anschließendes Überprüfen an der Maschine“ dienen vornehmlich der Überprüfung des Gut-Falls, wie er z. B. im Pflichtenheft definiert wurde. Ein weitaus wichtigerer Bestandteil der VIBN ist es aber den Schlecht-Fall, sprich den Fall von auftretenden Fehlern und Störungen an der Maschine zu testen. Dazu besitzt die Simulationsschnittstelle die Möglichkeit, Werte des virtuellen Modells der Werkzeugmaschine zu beeinflussen – durch sogenanntes „Forcen“, bzw. „Erzwingen“, der Werte im Modell, beschrieben in [9]. Dieses dauerhafte Überschreiben funktioniert sowohl für Ein- und Ausgangssignale als auch für interne Zustandssignale des Modells. Somit kann beliebig oft während eines Testablaufs das Simulationsmodell der Werkzeugmaschine automatisiert beeinflusst und wieder in seinen Ursprungszustand gebracht werden.

Durch die Automatisierung des Zugriffs auf die drei Schnittstellen und die Wiederverwendung von angelegten Testabläufen wird es möglich, die VIBN schneller und fehlerfreier durchführen zu können. Zusätzlich hat der Tester am Ende des Testablaufs einen detaillierten Einblick in die getätigten Aktionen des Testframeworks durch das Protokoll.

12.7 Praxisanwendung

Aus den Überlegungen in den vorangegangenen Kapiteln ist für den industriellen Einsatz ein TAW für den automatisierten Test bei der VIBN entstanden. Als Basis wurde das bestehende Softwaretestwerkzeug *expecco* für den Einsatz bei der VIBN an Werkzeugmaschinen um Schnittstellen und Funktionen erweitert. Das entstandene TAW *ISG-dirigent* bietet damit die folgenden Funktionen:

- Grafische oder textuelle Erstellung von Tests
- Nutzen einer Bibliothek für den Test an von gängigen Steuerungssystemen
- Simultane Erstellung von Testprotokoll und (zertifizierungsfähigem) Testreport zur Testausführung

Gekoppelt mit der Simulationssoftware ISG-virtuos, konnte für einen Werkzeugmaschinenhersteller eine Umgebung für den automatisierten Ablauf und die automatisierte Auswertung seiner Tests aufgebaut werden. Die auf Wiederverwendung angelegten Bibliotheken von ISG-dirigent wurden dabei als besonders hilfreich empfunden. Abb. 12.6 zeigt einen laufenden Test an einem Bearbeitungszentrum im Rahmen der VIBN. Trotz vollautomatischen Ablaufs ist eine übersichtliche Betrachtung des Testablaufs an einem Laborarbeitsplatz weiterhin möglich. Dabei werden die virtuelle Werkzeugmaschine inklusive HMI (links) und der aktuell aktive Testbaustein (rechts) sichtbar.

In Abb. 12.7 wird ein abgeschlossener Testablauf in ISG-dirigent aufbereitet dargestellt. Während des Testablaufs ist ein Fehler aufgetreten, dieser wird gut sichtbar, bis in die oberste Hierarchieebene hinein, angezeigt.

Eine häufig eingesetzte Steuerung an Werkzeugmaschinen für die spanende Metallbearbeitung ist die SINUMERIK 840D sl von Siemens, an der stellvertretend im Folgenden der Umfang von ISG-dirigent gezeigt werden soll: Bereits das Basiswerkzeug *expecco* bringt eine Standard-Bibliothek zum Erstellen von automatischen Abläufen mit. So ermöglicht etwa die Qt-Bibliothek die Kommunikation der Testbausteine mit den Elementen der Bedienoberfläche der SINUMERIK 840D sl, während die VNC-Bibliothek (Virtual Network Computing) die Übertragung von Tastatureingaben an das Bedienfeld der SINUMERIK 840D sl erlaubt. Die SCP-Bibliothek (Secure Copy Protocol) wiederum ermöglicht die Übertragung von Dateien (etwa von NC-Programmen) von und zur SINUMERIK 840D sl.

Die spezifische Bibliothek für die Siemens SINUMERIK 840D sl ergänzt das Werkzeug zur vollen Funktionalität für die VIBN einer mit Sinumerik gesteuerten Maschine.

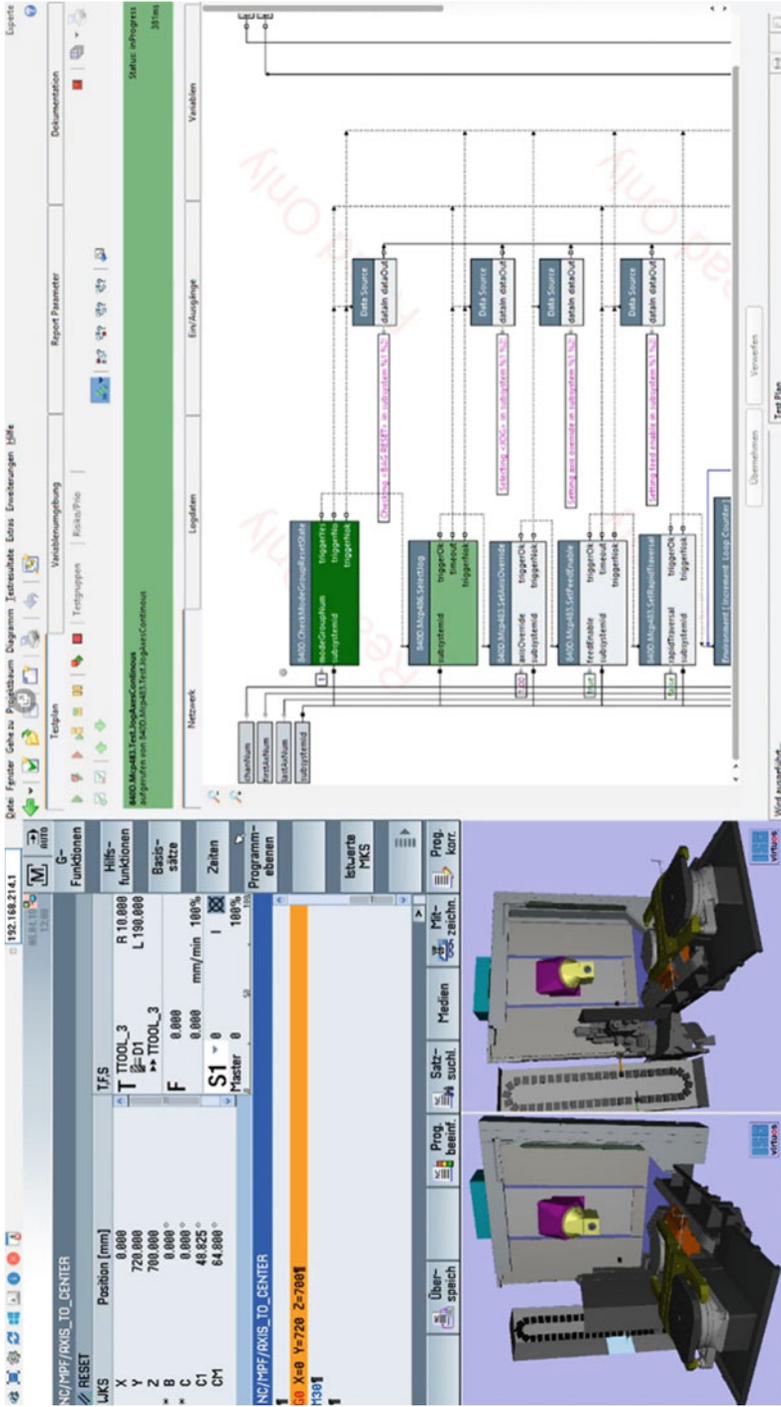


Abb. 12.6 Ansicht eines automatisiert ablaufenden Tests bei der Virtuellen Inbetriebnahme, rechts: ISG-dirigent, links oben: Siemens Sinumerik HMI, unten links: ISG-virtuos Simulationsumgebung mit virtuellem Bearbeitungszentrum

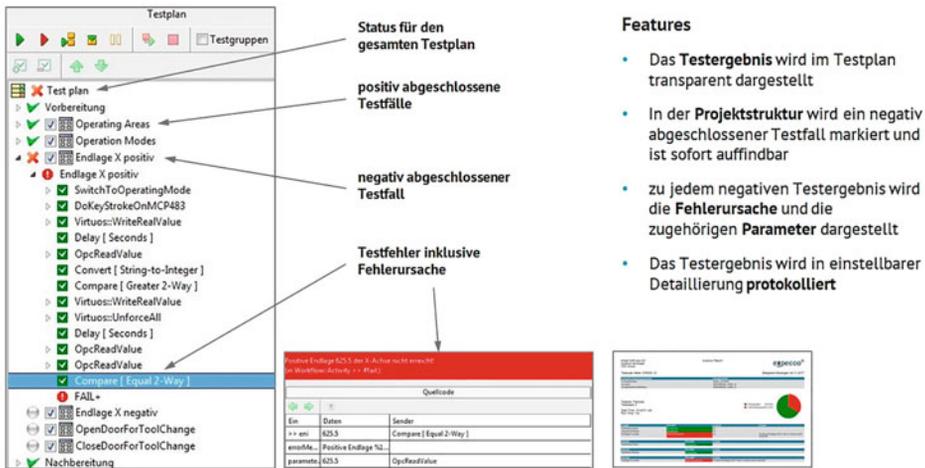


Abb. 12.7 Übersicht eines abgeschlossenen Tests in ISG-dirigent

Aufbauend auf den Bibliotheken OPC UA Client Interface und ISG-virtuos Client Interface zur Kommunikation mit der Steuerung beziehungsweise mit dem Simulationsmodell, ist diese Bibliothek in folgende Unterbereiche unterteilt, siehe Abb. 12.8:

- Die Bausteine der Gruppe *Bedienfeld* (OP 012 – Operators Panel) ermöglichen die nutzergerechte Interaktion der Testbausteine mit der Steuerung bezogen auf das Siemens-Bedienfeld.
- Die Bausteine der Gruppe *Maschinensteuertafel* (MCP483 – Machine Control Panel) ermöglichen die nutzergerechte Interaktion der Testbausteine mit der Steuerung bezogen auf die Siemens-Maschinensteuertafel.
- Die Bausteine der Gruppe *Maschinen- und Settingdaten* (Machine and Settings Data) ermöglichen den nutzergerechten Zugriff der Testbausteine auf diese Daten der Siemens Steuerung.
- Die Bausteine der Gruppen *Arithmetikparameter* (Arithmetic Parameters) und *Anwenderdaten* (User Data) ermöglichen den nutzergerechten Zugriff der Testbausteine auf diese Parameter der Siemens Steuerung.
- Und schließlich ermöglichen die Bausteine der Gruppe *SPS-Daten* (PLC Data) den nutzergerechten Zugriff der Testbausteine auf diese Daten der Siemens Steuerung.

Der Einstieg in die Testautomatisierung während der VIBN gelingt dann am schnellsten, wenn der Maschinenhersteller eine Systematisierung seiner Testabläufe vorgenommen hat. Dazu gehört die Definition der Testabläufe auf Basis einzelner Test-Schritte, die Erfassung der erwarteten Ergebnisse eines Testablaufs und die Erstellung eines Protokolltemplates. Danach kann mit der Identifizierung der Gleichteile zwischen Testabläufen

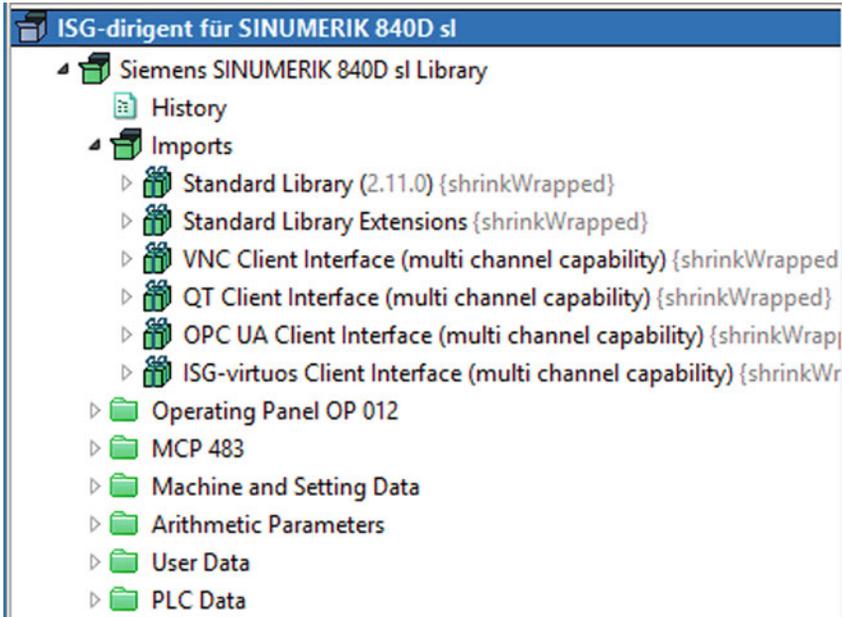


Abb. 12.8 Aus der Liste der geladenen Bibliotheken, die für den Testablauf zur Verfügung stehen, wählt der Benutzer Bausteine aus, die er für seinen Testablauf nutzen möchte

begonnen werden. So kann sukzessive eine Bibliothek an wiederverwendbaren Test-Schritten aufgebaut werden. Das Protokolltemplate dient zur schnellen Übersicht, wie ein Test verlaufen ist und wo eventuelle Fehler gefunden wurden.

Da mit dem TAW zumeist sehr komplexe Programme getestet werden, ist es hilfreich, wenn das Testframework parallele Abläufe, Verzweigungen und Schleifen im Testablauf zulässt, siehe Abb. 12.9.

In der Praxisanwendung hat es sich bewährt eine dynamische Entwicklung der Testabläufe zu nutzen. Das Basistestwerkzeug bietet in ISG-dirigent die Möglichkeit laufende Tests anzuhalten, zu modifiziert und fortzusetzen. Weiterhin können mit den Techniken **Haltepunkte definieren** und **schrittweise Ausführung** Tests selbst getestet und debuggt werden. Abb. 12.10 zeigt dazu einen an einem Haltepunkt gestoppten Test.

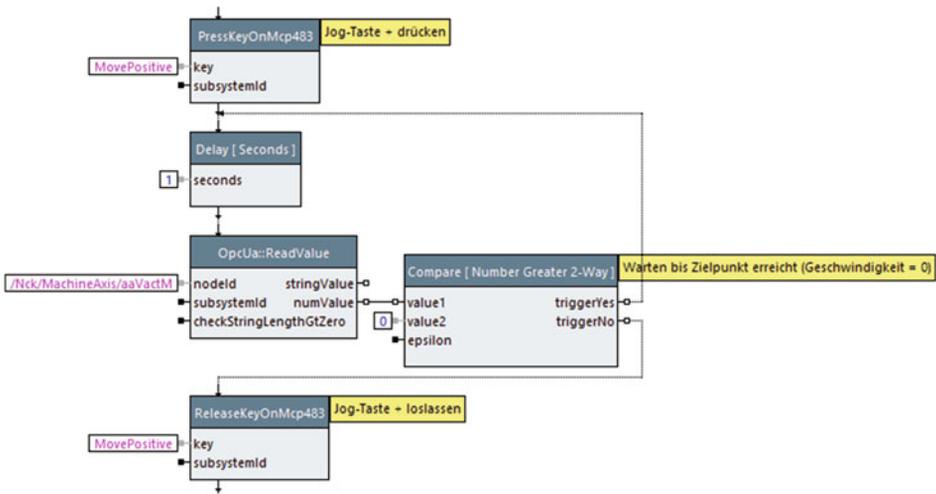


Abb. 12.9 Beispiel einer Schleife im Testablauf

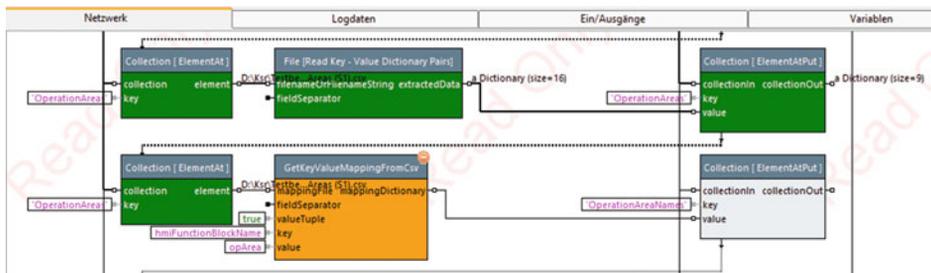


Abb. 12.10 Testablauf mit aktivem Haltepunkt, siehe Testbaustein in orange

12.8 Vorteile der Testautomatisierung für die Qualitätssicherung in der Entwicklung

Beim Einsatz eines Werkzeugs zur Testautomatisierung bei der VIBN entstehen die folgenden Vorteile:

- Sich häufig wiederholende Testabläufe werden aufwandsarm durchgeführt und können auch außerhalb normaler Arbeitszeiten stattfinden
- Durch ein häufigeres und umfangreicheres Testen steigen die Zuverlässigkeit und Robustheit der Steuerungsprogramme und damit auch der gesamten Maschine
- Die Abläufe und damit die Ergebnisse der Tests sind zuverlässig reproduzierbar
- Ein bestehendes Anforderungsmanagement im Unternehmen lässt sich mit der Testautomatisierung verknüpfen

Durch die Integration von Testautomatisierung in den Entwicklungs- und Inbetriebnahmeprozess von Werkzeugmaschinen lässt sich die primäre Arbeit bei der Qualitätssicherung auf die Erstellung der Tests eingrenzen. Zum anderen reduziert sich mit einem TAW der Zeitaufwand für das Testen, sodass Unternehmen viel eher bereit sind, Testautomatisierung als Standard einzuführen. Neben einer Zeitersparnis ermöglichen die automatisierten Tests frei gewordene Personalressourcen anderweitig einzusetzen. Weiterhin sind Dauertests kein Problem, sodass sporadisch auftretende Fehler häufiger entdeckt werden können. Der Kunde profitiert, da er mit weniger „Feuerwehreinsätzen“ direkt vor Ort rechnen muss, da der Hersteller der Maschine Fehler an den Steuerungsprogrammen vorab zuverlässiger beseitigen konnte.

12.9 Zusammenfassung und Ausblick

In diesem Kapitel zur Testautomatisierung wurde der aktuelle Stand der VIBN von Werkzeugmaschinen aufgezeigt. Aus diesem Stand der Technik wurde mithilfe von Fragebögen und Interviews, bei führenden Werkzeugmaschinenherstellern aus Deutschland, eine Auflistung an Problemen der VIBN erstellt. Aus den Problemen wurden im Folgenden ein Handlungsbedarf und eine passende Lösung erarbeitet. Der Fokus lag dabei auf der Verbesserung der Tests von Steuerungsprogrammen während der VIBN. Die Hersteller benötigen Unterstützung, um einen zeit-, personal-, und damit kostenreduzierten Test umzusetzen. Neben diesen drei Faktoren ist mit der erarbeiteten Lösung eine weit höhere Testabdeckung bei gleichem Aufwand möglich. Weiterhin werden Praktiken aus dem Testmanagement im Lösungskonzept unterstützt. Die Lösung sieht eine Automatisierung der Testfälle an einer HiLS vor. Dabei werden Gut- und Schlecht-Fall Tests mittels eines TAW durchgeführt. Das Testframework im TAW unterstützt den Tester beim Erstellen und Wiederverwenden von Testfällen, sowie durch Protokollierung und Auswertung während und nach dem Testablauf.

Die vorgestellte Lösung ist in Form der Bibliothekssammlung ISG-dirigent in Verbindung mit expeco kommerziell verfügbar und wird von Werkzeugmaschinenherstellern erfolgreich eingesetzt. Ausstehend ist die Herausforderung den automatisierten Test als neues „Werkzeug“ in den Entwicklungsprozess der Hersteller zu integrieren.

Literatur

1. Verein Deutscher Ingenieure e.V. (VDI); Verband der Elektrotechnik Elektronik Informationstechnik e.V. (VDE), VDI/VDE 3693 Blatt 2 Virtuelle Inbetriebnahme - Einführung der virtuellen Inbetriebnahme in Unternehmen, Berlin: Beuth Verlag, 2018.
2. Benington H (1987) Production of Large Computer Programs. ICSE '87 Proceedings of the 9th international conference on Software Engineering, S 299–310

3. Kaefer M (2016) Steuerungsarchitektur und Kommunikationstechnik. Vorlesung Steuerungstechnik II, Universität Stuttgart S. 67–76
4. Daniel C (2015) Virtuelle Komponenten verkürzen Inbetriebnahme. *MaschinenMarkt* 41:40–42
5. Pritschow G, Röck S (2004) Hardware in the loop. Simulation of machine tools. *CIRP Annals – Manufactur Technol* 53:295–298
6. Preiml H (2013) Erfahrungen mit der virtuellen Inbetriebnahme in der Anlagen-Projektierung. *MaschinenMarkt*, 19 07 2013. <http://www.virtual-reality-magazin.de/heitec-erfahrungen-mit-der-virtuellen-inbetriebnahme-der-anlagen-projektierung>. Zugegriffen: 10. Mai 2016
7. VDMA Software und Digitalisierung, Simulation und Visualisierung – Erste Ergebnisse der Trendumfrage 23 03 2017 https://sud.vdma.org/documents/15012668/20460216/Trend-Analyse-Simulation_Internet_1505303416307.pdf/9021347b-0670-4d6b-aaed-e1409d9e2d13. Zugegriffen: 3. Juli 2020
8. Willrett H (2007) Zwilling im Computer – Virtuelle Werkzeugmaschine: die Zukunft in der Arbeitsvorbereitung. *Industrieanzeiger* S. 28–31
9. VDI/VDE (2015) Virtuelle Inbetriebnahme – Modellarten und Glossar, VDI/VDE-Richtlinien 3693–1. Beuth, Berlin
10. Pritschow G (2006) Einführung in die Steuerungstechnik. Carl Hanser, Wien
11. Röck S (2007) Echtzeitsimulation von Produktionsanlagen mit realen Steuerungselementen. Jost-Jetter, Stuttgart
12. Verband Deutscher Maschinen- und Anlagenbau e. V. (VDMA) (2020) Leitfaden Virtuelle Inbetriebnahme – Handlungsempfehlungen zum wirtschaftlichen Einstieg. VDMA Verlag, Frankfurt a. M.
13. ISO/IEC/IEEE (2013) Software and systems engineering—Software testing—ISO/IEC/IEEE 29119 Part 1: Concepts and definitions, Geneva; New York: ISO;IEC;IEEE
14. Kormann B, Vogel-Heuser B, Friedrich M (2012) Befragung deutscher Maschinenbauunternehmen zum Thema Softwaretest – Handlungsbedarf für den Maschinen-/Anlagenbau und Lösungsvorschlag. *Automation 2012 - 13*. Branchentreff der Mess- und Automatisierungstechnik, Baden-Baden

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

