Absicherung der Rekonfigurationen von Produktionssystemen während des Betriebs

Warum Assistenzsysteme beim Testen verteilter IT-Systeme an Relevanz gewinnen

Validating reconfigurations of production systems within the operating phase

Why assistance systems gain relevance for testing of distributed systems

A. Zeller, M.Sc., Institut für Automatisierungstechnik und Softwaresysteme, Universität Stuttgart;
Prof. Dr.-Ing. **M. Weyrich**, Institut für Automatisierungstechnik und Softwaresysteme, Universität Stuttgart

Kurzfassung

Industrie 4.0 verursacht aufgrund rekonfigurierbarer Produktionssysteme einen disruptiven Wandel des Qualitätsbegriffs. Die scharfe Trennung zwischen Test (in der Engineering- und Inbetriebnahmephase) und Qualitätssicherung (in der Betriebsphase) löst sich auf. Verursacht durch Rekonfigurationen in der Betriebsphase müssen betroffene Produktionssysteme erneut abgesichert werden. Insbesondere während des Betriebs kommt einem effizienten Test eine herausragende Rolle zu, da durch minimalen Testaufwand der Betrieb möglichst wenig beeinträchtigt und gefährdet werden soll. In diesem Beitrag wird ein Verfahren vorgestellt, das, im Gegensatz zu konventionellen Verfahren, die Absicherung in der Betriebsphase betrachtet. Dabei werden explizit zukünftige Rahmenbedingungen der Produktion, wie Rekonfigurierbarkeit und verteilte Netzwerkarchitekturen, berücksichtigt.

Abstract

“Industrie 4.0” causes a disruptive change of the quality management. The sharp distinction between test (in the engineering and commissioning phase) and the quality assurance (in the operation phase) dissolves. Caused by reconfigurations within the operation phase affected systems have to be assured once again. Especially within the operation phase the case selection matters because the operation shouldn’t be affected by unnecessary test runs. For this reason the concept presented regards, in contrast to conventional concepts, the test case selection within the operating phase. Thereby future conditions of production like reconfigurability and distributed network architectures are considered.

1. **Motivation und Zielstellung**

Aufgrund sehr hoher Anforderungen an Zuverlässigkeit und Verfügbarkeit stellt die Absicherung von funktionalen Anforderungen einen unverzichtbaren Prozess nach Änderungen an Produktionsanlagen dar. Diese Änderungen können Rekonfigurationen, SW-Updates oder der Austausch von Komponenten sein. Da die Absicherung der Anlage meist mit hohem personellem und zeitlichem Aufwand verbunden ist, gilt bei vielen Anlagenbetreibern der Grundsatz „never change a running system“.

Ermöglicht durch den flächendeckenden Einzug von Informationstechnik und getrieben durch die Globalisierung und sich verändernden Kundenanforderungen geht der Trend in Richtung Flexibilisierung der Produktion. Dabei gibt es schon zahlreiche Konzepte und Lösungsansätze, wie diese Flexibilität informationstechnisch durch Service-orientierte Architekturen gewährleistet werden kann. Die daraus resultierenden Herausforderungen zur Absicherung von rekonfigurierbaren Anlagen werden jedoch kaum betrachtet. Im Rahmen dieser Veröffentlichung werden die Auswirkungen der Änderungen der Produktions-IT auf die Absicherung analysiert. Aufbauend auf diese neuen Anforderungen wird ein Konzept entwickelt, wie Testmanager beim Absicherungsprozess der Produktionssysteme unterstützt werden können.

2. Produktions-IT im Wandel

Rekonfigurierbarkeit, Ad-Hoc-Vernetzung, Wiederverwendbarkeit, Updates im Feld und durchgängige Kommunikation sind Attribute, welchen eine „Smart Factory“ genügen soll [1]. Dabei nimmt das IT-System, welches die Produktion koordiniert, eine zentrale Rolle ein und soll in diesem Aufsatz betrachtet werden. Damit Produktionsanlagen diese Anforderungen erfüllen, ist ein grundlegender Wandel der IT-Systemstrukturen notwendig. Verteilte Steuerungsarchitekturen weisen dabei aufgrund der hohen Flexibilität zahlreiche Vorteile auf [2]. Zur Umsetzung dieser flexiblen und verteilten Produktions-IT werden Paradigmen wie Service-orientierte Architekturen (SOA) oder Software-Agenten genutzt [3] [4]. Typischerweise haben diese Formen verteilter IT-Systeme zentrale Elemente, um den Kommunikationsoverhead zu minimieren. Wie in Bild 1 dargestellt, kann es sich dabei um eine Verwaltung der Teilsysteme handeln, welche analog zu einem Telefonbuch, Informationen über die im Netzwerk vorhandenen Teilsysteme bereitstellt.

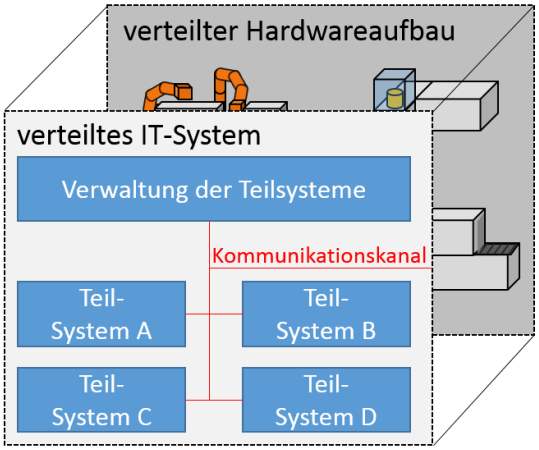

Bild 1: Typischer Aufbau eines verteilten IT-Systems mit zentraler Komponente. Verteilte IT-Systeme erlauben eine flexible Koordination und Steuerung von Produktionsanlagen.

Dabei stellen die Teilsysteme üblicherweise einen „Server“ oder „Client“ dar. „Server“ bieten Dienste an. Dienste besitzen eine standardisierte Schnittstelle, über die auf eine dahinter liegende Funktionalität zugegriffen werden kann. Dies kann beispielsweise die Funktionalität einer Produktionsmaschine sein, die über die Schnittstelle eines Dienstes im IT-Netzwerk vertreten wird (Basisdienst). Alternativ kann über Dienste auch auf eine Steuerungseinheit zugegriffen werden, welche einen Produktionsablauf koordiniert und somit über Prozesswissen verfügt (Prozessdienst). Clients greifen als Kunden auf die Dienste zu, welche die Server anbieten.

Damit verteilte IT-Systeme beherrschbar und flexibel sind, werden folgende Eigenschaften gefordert, welchen auch SOA und Software-Agenten genügen:

- *Kapselung* der Funktionalität: Ermöglicht die Trennung der Implementierung vom Aufrufmechanismus. Dadurch wird die Komplexität der Implementierung vor dem Ingenieur verborgen.
- *Semantisch beschriebene Schnittstellen:* Ein Service wird über seine Schnittstelle beschrieben. Die vollständige Beschreibung der Schnittstelle inklusive deren Semantik ermöglicht eine Interoperabilität d. h. eine Vernetzung unterschiedlicher Hersteller und IT-Ebenen. Dies wird auch als „wohldefinierte Schnittstelle“ [5] bezeichnet. Zur Beschreibung der Module könnten Technologien wie FDI (Field Device Integration) verwendet werden [6]. Als Minimum an selbstbeschreibenden Daten der Dienste zur Integration in ein bestehendes Netzwerk sind die eindeutige ID des Servers sowie die Dienstbeschreibung notwendig.

- *Sichtbarkeit:* Es muss bekannt sein, welche Komponenten im Netzwerk vertreten sind. Bei Ad-Hoc-Vernetzung muss somit der neu hinzugefügte Dienst bekannt gemacht werden. Dies wird meist über eine Diensteverwaltung realisiert. Bei der Anmeldung eines Servers im Netzwerk registriert dieser seine angebotenen Dienste bei der Diensteverwaltung, welche eine fest definierte ID besitzt. Clients durchsuchen bei Bedarf die Diensteverwaltung nach verfügbaren Diensten.
- *Skalierbarkeit:* Die Eigenschaft der Sichtbarkeit und semantisch beschriebene Schnittstellen erlauben eine einfache Ad-Hoc-Vernetzung und somit eine Skalierung des Netzwerks.
- *Lose Kopplung:* Server und Client sind nicht statisch miteinander verbunden, sondern binden sich, wenn das Bedürfnis des Clients mit den Fähigkeiten des Servers übereinstimmt. Dies ermöglicht die Wiederverwendung eines Dienstes durch verschiedene Clients.
- *Orchestrierbarkeit:* Abstraktere Dienste, sogenannte Prozessdienste, koordinieren Basisdienste zur Durchführung eines Produktionsprozesses. Dabei interagieren die Prozessdienste ihrerseits als Clients mit Basisdiensten. Dies erlaubt den Aufruf eines höherwertigen Komplettangebots, wozu kein explizites Prozesswissen notwendig ist.

Um in verteilten, flexiblen IT-Systemen agieren zu können, bedürfen die weitgehend autonomen Teilsysteme einer komplexeren Kommunikationsschnittstelle als bei zentralen IT-Systemen. Die Teilsysteme müssen befähigt sein, mit verschiedenen Kommunikationspartner zu interagieren und sich bei der Integration in ein bestehendes IT-Netzwerk proaktiv zu melden. Dies resultiert in einer höheren Komplexität der Teilsysteme. Dieses Gesamtsystem ist aufgrund der genannten Eigenschaften sehr flexibel, setzt dadurch aber neuartige Rahmenbedingungen für dessen Absicherung.

3. Anforderungen an die Absicherung eines flexiblen Produktionsverbunds

Bei konventionellen Produktionsanlagen werden, wie im V-Modell beschrieben, Tests hauptsächlich während der Engineering- und Inbetriebnahmephase durchgeführt [7]. Dabei werden während der Engineeringphasen zuerst das einzelne Modul (Modultest), anschließend die Kooperation mehrerer Module (Integrationstest) und schließlich das Gesamtsystem (Systemtest) gegen die Anforderungen getestet. Abgeschlossen wird der Testprozess bei der Inbetriebnahme der Anlage beim Kunden (Abnahmetest). Ist der Abnahmetest erfolgreich, beginnt die Betriebsphase der Anlage. Dabei finden hauptsächlich qualitätssichernde Maßnahmen statt. Sind, meist nach längerer Betriebsdauer, größere Wartungs- bzw.

Umbaumaßnahmen an der Anlage notwendig, müssen diese erneut durch Tests abgesichert werden [8].

Im Gegensatz zur Absicherung von konventionellen Produktionsanlagen unterscheiden sich die Rahmenbedingungen zu rekonfigurierbaren, verteilt gesteuerten Produktionsanlagen stark.

Durch Rekonfigurationen und Software-Updates im Feld ist von einer Zunahme von Änderungen an der Produktionsanlage während der Betriebsphase auszugehen. Die erfolgreiche Durchführung der Änderungen muss über Testmaßnahmen verifiziert werden, weshalb von einer Zunahme notwendiger Testfälle auszugehen ist. Zum einen muss öfters verifiziert werden, zum anderen ergeben sich aus der gesteigerten Kommunikationsfähigkeit und der Komplexität der Teilsysteme eine große Anzahl notwendiger Testfälle [9].

Durch die Anforderung an Interoperabilität bei gleichzeitiger Heterogenität der Systeme ergeben sich neuartige Herausforderungen. Ermöglicht durch eine semantisch standardisierte Schnittstelle existiert eine Vielzahl neuer Kooperationspartner, die in noch nie vorgekommener Situation miteinander interagieren sollen. Wurden bei konventionellen Produktionsanlagen die Kommunikationsfähigkeit zur einfacheren Absicherung bewusst gering gehalten, stellt dies eine Kehrtwende dar.

Im Allgemeinen kann aufgrund der losen Kopplung zwischen Client und Server nicht davon ausgegangen werden, dass einem Dienst bekannt ist, von welchen Clients er aufgerufen werden kann. Diese Information muss zwingend nur der Client besitzen, da dieser die Interaktion initiiert. Dies erzeugt eine Verteilung der Kenntnis über die Abhängigkeiten auf die Teilsysteme des Netzwerks. Durch Ad-Hoc-Vernetzung, Rekonfigurationen und Software-Updates können sich diese Abhängigkeiten ändern, sodass kein statisches funktionales Abhängigkeitsmodell existiert. Die Kenntnis der Abhängigkeiten ist aber zwingend notwendig, um die Auswirkungen von Änderungen auf andere Teilsysteme bestimmen zu können.

In Bild 2 ist der physische Aufbau eines verteilten IT-Systems dargestellt. Die Dienste befinden sich auf den Mikrocontrollern. Die Basisdienste (A und B) kapseln die Steuerung der jeweiligen Produktionsmaschine und befinden sich somit auf deren Steuerungseinheiten. Die Prozessdienste (C, D und E) befinden sich virtualisiert auf Mikrocontrollern im Netzwerk. Wie aus Abbildung 2 ersichtlich werden die funktionalen Abhängigkeiten zwischen den Diensten aus dem physischen Aufbau des Produktionsnetzwerks nicht ersichtlich. Durch den modularen Hardwareaufbau ist oft auch keine Aussage über Abhängigkeiten zwischen einzelnen Modulen möglich. Diese Abhängigkeitsinformation ist auf die einzelnen Dienste verteilt. Dies erschwert es dem Testmanager enorm, einen Überblick über die Abhängigkeiten des Gesamtsystems zu erhalten.

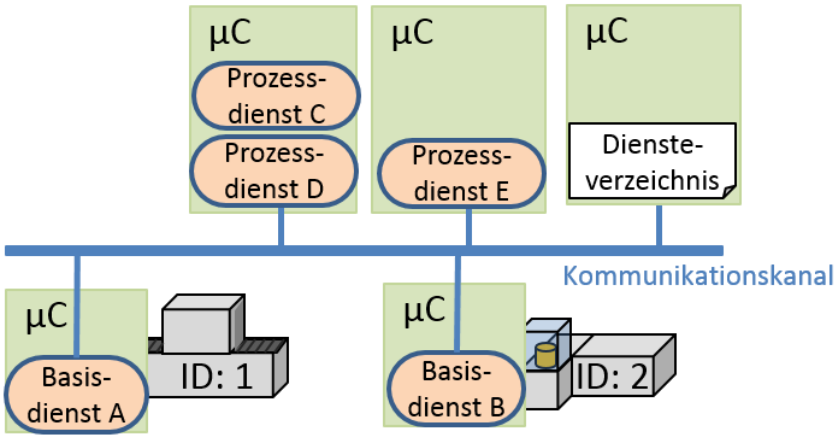

Bild 2: Physischer Aufbau der IT-Infrastruktur eines verteilten Produktionssystems. Dieser erlaubt oft keine direkte Aussage über logische Abhängigkeiten des IT-Systems.

Durch die komplexere Kommunikationsschnittstelle der Teilsysteme entstehen aber auch neue Chancen Informationen aus dem System zu beziehen.
Zur Unterstützung des Testmanagers wird im folgenden Kapitel ein Assistenzsystem vorgestellt, welches die fraktalen Informationen, die in dem verteilten IT-System des Produktionsverbunds vorhanden sind, fusioniert und daraus einen globalen Abhängigkeitsgraph generiert.

4. Unterstützung bei der Absicherung mithilfe eines Testmanagement-Assistenzsystems

Ziel des Konzepts eines Testmanagement-Assistenzsystems ist der Aufbau eines funktionalen Abhängigkeitsgraphs aus Informationen von verteilten Informationsquellen des IT-Systems eines Produktionssystems. Anhand dieser Informationen werden relevante Testfälle selektiert und dem Testmanager zur Absicherung von Änderungen empfohlen. Der schematische Aufbau dieses Testmanagement-Assistenzsystems ist in Bild 3 dargestellt. Das Konzept besteht im Wesentlichen aus vier logischen Komponenten. Dabei werden Informationen aus dem Produktionsverbund akquiriert, welche strukturiert in einer Informationsbasis verwaltet werden. Diese werden durch die Informationsverarbeitungskomponente interpretiert, in der Darstellungskomponente grafisch aufbereitet und dem Testmanager über eine grafische Benutzungsschnittstelle ausgegeben. Auf die Funktionsweise der einzelnen Komponenten wird im Folgenden eingegangen.

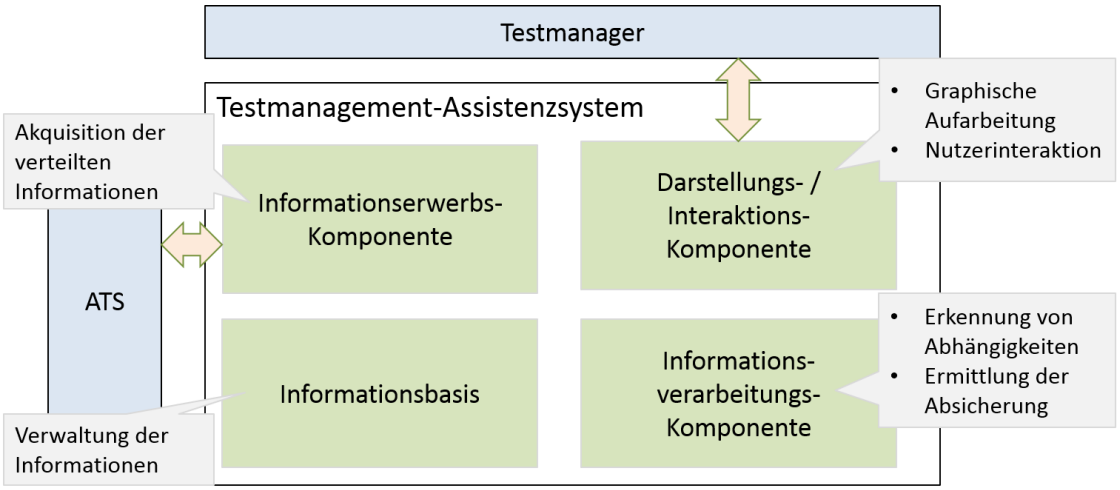

Bild 3: Schematischer Aufbau des Testmanagement-Assistenzsystems

4.1 Informationserwerbskomponente

Die Informationserwerbskomponente stellt die Schnittstelle des Assistenzsystems zum IT-Netzwerk des Produktionssystems dar. Dabei fungiert sie als Client, welcher das Produktionsnetzwerk nach verfügbaren Diensten und deren Beschreibungsdaten durchsucht. Dies beinhaltet die verfügbaren Dienste und die IDs der Server, welche diese Dienste anbieten. Diese Information kann beispielsweise über das Auslesen des Diensteverzeichnisses akquiriert werden. Falls die Dienste darüber hinaus selbstbeschreibende Daten, beispielsweiße in Form einer Verwaltungsschale bereitstellen, können diese über explizite Anfrage an den jeweiligen Server ausgelesen werden. Diese Informationen können unter anderem eine Versionsnummer, Prozessmodelle oder Metadaten über Testfälle, welche zur Absicherung des Dienstes notwendig sind, beinhalten. Um den Datenverkehr gering zu halten, wird dies ausschließlich nach Beauftragung durch die Informationsverarbeitungskomponente durchgeführt.

4.2 Informationsbasis

Die Informationsbasis verwaltet einerseits die von der Informationserwerbskomponente erhaltenen Rohinformationen, andererseits auch die von der Informationsverarbeitungskomponente prozessierte Informationen, welche zusätzliche Attribute über die Abhängigkeiten und Kritikalität besitzen.

4.3 Informationsverarbeitungskomponente

Die Informationsverarbeitungskomponente stellt das Gehirn des Testmanagement-Assistenzsystems dar. Wie in Bild 4 dargestellt werden die partiellen Informationen der

Dienste, welche in der Informationsbasis gespeichert sind, zu einem globalen Abhängigkeitsgraph fusioniert. Dies kann über Reasoner umgesetzt werden. Dabei entspricht die Aussage „ruft auf“ in Gegenrichtung der Aussage „wird aufgerufen von“. Die daraus generierten Attribute werden in der Informationsbasis strukturiert hinterlegt. Besitzt der Dienst C die Informationen, dass zu dessen Ausführung der Aufruf von Dienst A und B notwendig ist, gilt ebenso für Dienst A und B, dass diese durch Dienst C aufgerufen werden können.

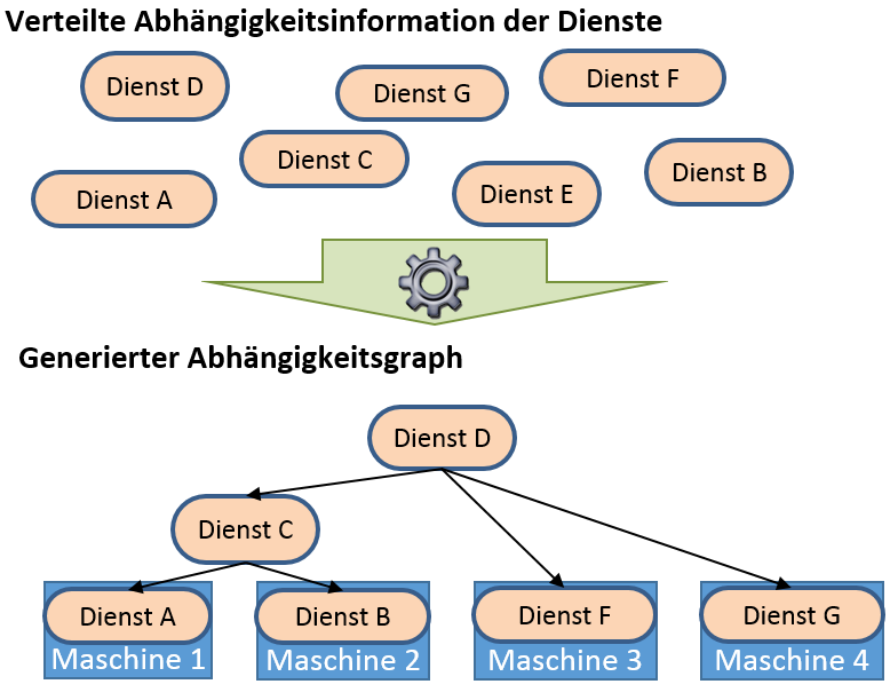

Bild 4: Generierung eines globalen Abhängigkeitsgraphs. Dazu werden die partiellen Abhängigkeitsinformationen der Dienste genutzt, um eine globale Ansicht zu generieren.

Zur Ermittlung der Auswirkungen von Änderungen eines Teilsystems auf andere Teilsysteme wird eine Sicht generiert, die aus den Teilsystemen besteht, welche von den Änderungen direkt oder indirekt über funktionale Abhängigkeiten betroffen sind. Dazu ist die Information von der Darstellungs- / Interaktionskomponente notwendig, welches Teilsystem geändert werden soll. Durch Kenntnis der Testfälle, die zur Absicherung des jeweiligen Dienstes notwendig sind, können die Testfälle, welche ein kritisches Teilsystem betreffen, vorgeschlagen werden. Die Kenntnis, welche Testfälle zur Absicherung eines Services notwendig sind, kann entweder über eine Cloud-Datenbank bereitgestellt oder abhängig von der jeweiligen Technologie in den beschreibenden Daten des Dienstes gespeichert werden. Die ermittelten Ergebnisse werden an die Darstellungskomponente weitergegeben.

4.4 Darstellungs- / Interaktionskomponente

Um dem Testmanager assistieren zu können, müssen relevante Abhängigkeiten grafisch aufgearbeitet werden. Dabei werden zum einen die Abhängigkeiten der von der Informationsverarbeitungskomponente generierten Sicht grafisch dargestellt (siehe Bild 4 unten), zum anderen die zur Absicherung der betroffenen Dienste relevanten Testfälle vorgeschlagen. Dazu wird eine Liste generiert, welche relevante Testfälle der Priorität entsprechend sortiert. Des Weiteren bietet die Interaktionskomponente die Möglichkeit, Informationen zu geplanten Änderungen über eine Nutzereingabe zu erfassen.

4.5 Validierung

Um den Nutzen und die Realisierbarkeit des Konzepts zu evaluieren, existiert noch ein Validierungsbedarf. Dabei sollen hauptsächlich folgende Thesen betrachtet werden:

- Der Grad der Absicherung kann bei gleichbleibendem Testaufwand erhöht werden.
- Es ist möglich einen Abhängigkeitsgraph aus einem verteilten IT-System automatisiert zu generieren.
- Es ist möglich, Sichten aus dem Abhängigkeitsgraphen zu generieren, die für den Testmanager einfach zu handhaben sind.

5 Fazit und Ausblick

Ein flexibles Produktionssystem bedarf eines flexiblen IT-Systems, welches den Produktionsablauf koordiniert und steuert. Dabei besitzen verteilte IT-Systeme Vorteile bezüglich Flexibilität und Skalierbarkeit. Verteilte IT-Systeme, die lose gekoppelt sind, stellen aber auch neuartige Anforderungen an den Testmanager. Zum einen sind Informationen über den Systemzustand auf zahlreiche Teilsysteme verteilt, zum andern bewirken Rekonfigurationen Änderungen des Systemmodells, welches die logischen Abhängigkeiten des Produktionssystems beschreibt. Zur Handhabung dieser Anforderungen kann angenommen werden, dass Assistenzsysteme unterstützend wirken können. Dabei wird sich zunutze gemacht, dass Assistenzsysteme aufgrund der durchgehenden Vernetzung von IT-Systemen in der Lage sind, verteilte Informationen aus dem IT-System zu akquirieren und diese zu interpretieren.

Da sich weder das Paradigma zur Koordination der verteilten IT-Systeme oder gar ein konkretes Protokoll durchgesetzt haben, wurde hierzu ein generisches Konzept zur Unterstützung der Absicherung entworfen. Dieses Konzept soll in weiteren Arbeiten anhand eines OPC-UA-Szenarios umgesetzt und evaluiert werden.

6 Literatur

[1] Kagermann, H. et. al.: *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. Abschlussbericht des Arbeitskreises Industrie 4.0, 2013.* https://www.bmbf.de/files/Umsetzungsempfehlungen_Industrie4_0.pdf abgerufen am: 23.03.2016

[2] Huffman D.: *Benefits of State Based Control. Isa.org*, White Paper, 2009.

[3] VDI/VDE; ZVEI: *Referenzarchitekturmodell Industrie 4.0* (RAMI4.0), 2015

[4] Göhner, P.: *Softwareagenten für die flexible Kopplung von Automatisierungssystemen*. 6. VDI-Expertenforum "Agenten im Umfeld von Industrie 4.0", München, 2014.

[5] Josuttis, N.: *SOA in der Praxis. System-Desgin für verteilte Geschäftsprozesse*, ISBN-13: 978-3898644761, 2008.

[6] ZVEI: *White Paper Modulbasierte Produktion in der Prozessindustrie - Auswirkungen auf die Automation im Umfeld von Industrie 4.0,* 2015. http://www.zvei.org/Publikationen/ZVEI-White-Paper-Modulare-Automation.pdf abgerufen am 24.03.2016

[7] Verein zu Weiterbildung des V-Modell XT e.V.: *V-Modell XT Das deutsche Referenzmodell für Systementwicklungsprojekte Version 2.0*, 2006.

[8] Zeller, A.; Weyrich, M: *Test Case Selection for Networked Production Systems*. In: 20th IEEE International Conference on Emerging Technologies and Factory Automation, Luxembourg, 2015.

[9] Zeller, A.; Weyrich, M: *Herausforderung Test verteilter Systeme – Wie Industrie 4.0 das Testen verändert*. In: atp edition - Automatisierungstechnische Praxis, 2015.