

Aussagekraft von OPC UA Testfällen für die innerbetriebliche Dateninfrastruktur

# OPC UA Tests im Kontext einer Dateninfrastruktur

T. Heinemann, S. Ajdinović, A. Lechler, O. Riedel

In der innerbetrieblichen Dateninfrastruktur spielt OPC UA eine maßgebliche Rolle zur Datenerfassung. Dies wird anhand der für OPC UA definierten Testfälle am Beispiel der OPC 40200 erläutert und es werden Vorteile der Tests für verschiedene Nutzergruppen diskutiert. Der Beitrag dieser Testfälle zur Interoperabilität mit anderen Technologien der Dateninfrastruktur wird aufgezeigt.

## Tests for OPC UA as part of data infrastructure

OPC UA plays an important role for data collection in the internal data infrastructure. The test cases defined for OPC UA are exemplified on OPC 40200. The paper points out the benefits of the tests for different user groups and how the test cases contribute to interoperability with other technologies in the data infrastructure.

### STICHWÖRTER

Betriebsdatenerfassung (BDE), Digitalisierung, Industrie 4.0

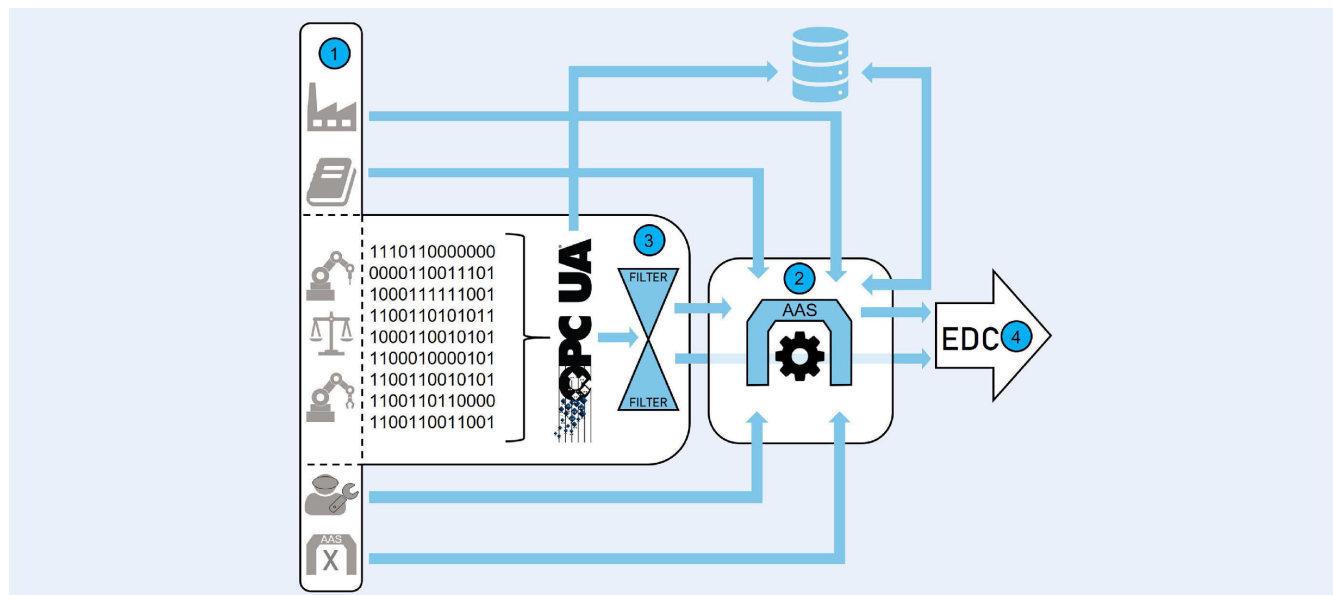


Bild 1. Beispielhafte Dateninfrastruktur eines produzierenden Unternehmens. Grafik: Universität Stuttgart, ISW

## 1 Einführung

Um die in der produzierenden Industrie anfallenden Produktions- und Prozessdaten gewinnbringend nutzen zu können, ist ein systematisches Vorgehen bei der Übertragung, der strukturierten Sammlung und der Weiterverarbeitung der Daten zwischen den einzelnen Akteuren in einer internen Dateninfrastruktur unabdingbar. Dieses Zusammenspiel kann durch die Verwendung von OPC UA (Open Platform Communications Unified Architecture) zum Informationsaustausch und der semantischen Beschreibung

der Daten, der Asset Administration Shell (AAS) und dem Einsatz des Eclipse Dataspace Connectors (EDC) realisiert werden. Bild 1 illustriert beispielhaft dieses Zusammenspiel.

Bild 1 zeigt eine mögliche Zusammenstellung verschiedener Akteure einer unternehmensinternen Dateninfrastruktur. Zur Befüllung der AAS (2) eines beispielhaften Produktes werden verschiedene Datenquellen (1) genutzt. Die vorhandenen Daten werden teilweise über OPC UA (3) in die AAS übertragen, ansonsten werden andere Übertragungsmechanismen genutzt. Um den Austausch der interoperablen AAS oder Daten aus

OPC UA über Unternehmensgrenzen hinweg sicher und vertrauenswürdig zu gestalten, wird der EDC (4) eingesetzt. Dieser zeichnet sich durch einen Datenaustausch mit integrierten Verbindungen der Kooperationspartner in Datenökosystemen aus [1].

### 1.1 Asset Administration Shell

Die AAS wird als zentrales Element zur Sammlung von strukturierten und unstrukturierten Daten von Assets verstanden. Sie kann als digitales Abbild einer Industrie-4.0-Komponente und damit als digitaler Zwilling betrachtet werden [2]. Durch die Verwendung standardisierter Submodelle erlaubt die AAS die sukzessive Anreicherung von Daten über den gesamten Lebenszyklus eines Assets. Fabrikdaten, wie Stromverbrauch während der Produktion, können direkt in die AAS integriert oder referenziert werden, sodass eine Rückverfolgung der verursachten Umweltfaktoren möglich ist, etwa zur Ermittlung des Carbon Footprints auf Produktebene. Weitere Informationen, die bereitgestellt werden können, sind Qualitätsdaten, die während der Produktion manuell oder automatisiert erfasst werden. [2]

Anstelle von Werksdaten können zusätzliche Dokumente wie Betriebsanleitungen oder Gefahrenhinweise in der AAS hinterlegt werden. Durch die Digitalisierung können diese Dokumente papierlos und mehrsprachig dem Produkt als Submodell hinzugefügt werden. Gleiches gilt zum Beispiel für Montageteile, die von externen Zulieferern produziert und geliefert werden: Sie können die eigens erstellte Produkt-AAS des Bauteils zur Integration in die AAS des Endprodukts zur Verfügung stellen, um die gesamte Lieferkette abzubilden. [2]

### 1.2 OPC UA in der betriebsinternen Dateninfrastruktur

Ist der Fokus auf dem Shopfloor die Erfassung und Verarbeitung von Maschinen- und Prozessdaten, so ist es möglich, diese über OPC UA (3) zur Verfügung zu stellen, siehe Bild 1 [3]. In dem Beispiel sind zwei Varianten aufgezeigt. Zum einen besteht die Möglichkeit, alle über OPC UA protokollierten Werte in Zeitreihendatenbanken zu speichern. Zum anderen können die Werte über bestimmte Filterkriterien, wie etwa kritische Temperaturwerte, mit einem Zeitstempel direkt in die AAS übertragen werden [2]. Bei der ersten Option besteht die Möglichkeit, dass die AAS auf die Datenbank verweist oder diese selbst in die AAS schreibt.

Zur Sicherstellung eines optimalen und fehlerfreien Datentransfers sind bereits während der Entwicklung und Implementierung von OPC-UA-Systemen Testfälle im Einsatz [4]. Diese können zum Beispiel verwendet werden, um die Konformität eines OPC-UA-Servers oder -Clients mit dem OPC-UA-Standard sicherzustellen oder um die Interoperabilität zwischen verschiedenen OPC-UA-Produkten zu testen. Außerdem ist ein Performance- und Stabilitätstest möglich. So können Probleme bereits frühzeitig im Entwicklungsprozess erkannt und behoben werden.

Der Fokus dieses Beitrags liegt innerhalb einer unternehmensinternen Dateninfrastruktur sowohl auf der Übertragung von Produktionsdaten mithilfe von OPC UA als auch auf der Verwendung von Testfällen während der Entwicklung und Implementierung von OPC UA Systemen.

### 1.3 Strukturierung des Beitrags

Dieser Beitrag widmet sich der spezifischen Betrachtung von Testfällen im Kontext von OPC UA. Der Fokus liegt dabei auf der näheren Erläuterung des Standards OPC UA in Kapitel 2.1 sowie der detaillierten Definition und Darstellung von Testfällen am Beispiel der Wägetechnik in Kapitel 2.2. Des Weiteren wird in Kapitel 3 die Bedeutung der Interoperabilität anhand von Beispielen erläutert und der Mehrwert sowie die Grenzen der Verwendung von Testfällen aus verschiedenen Rollen diskutiert.

Abschließend wird in Kapitel 4 die Einbettung der Testfälle in die Gesamtarchitektur betrachtet und die Ergebnisse werden zusammengefasst.

## 2 Stand der Technik

In diesem Kapitel wird der Standard OPC UA vorgestellt, um den Kontext der Testfälle zu verdeutlichen. Zunächst werden allgemeine OPC UA Testfälle vorgestellt, bevor auf spezifische Testfälle für die Companion Specifications am Beispiel der „OPC 40200“ eingegangen wird.

### 2.1 Grundlagen von OPC UA

Der OPC UA Standard definiert für die Datenübertragung eine Architektur, die alle notwendigen Aspekte der Kommunikation abdeckt [5]. **Bild 2** stellt diese Architektur mit den Komponenten „Protocol Binding“, „Communication Model“, „Built-in Security“ und „Information Models“ dar.

Über die Protocol Bindings wird festgelegt, mit welchen Protokollen und Encodings die Daten übertragen werden. OPC UA bietet mehrere Alternativen für Anpassungen an unterschiedliche Kommunikationsanforderungen. Die Kommunikationsmodelle in OPC UA sind Server/Client und Publish/Subscribe. Für beide legt der Standard fest, über welche Mechanismen die jeweiligen Kommunikationspartner miteinander interagieren können, um zum Beispiel eine Verbindung aufzubauen oder Datenwerte anzufordern. [6–8]

Built-in Security enthält für die jeweils orthogonalen Komponenten der OPC-UA-Architektur geeignete und aktuelle Mechanismen der Übertragungssicherheit und der Benutzerverwaltung [9]. Das OPC UA Meta Model innerhalb der Informationsmodelle definiert die Regeln, nach denen Informationen dargestellt werden. Das Built-in-Informationsmodell und das DI Model in Bild 2 stellen die grundlegenden Bausteine für die Informationsdarstellung zur Verfügung. Die Companion Specifications (veröffentlicht von der OPC Foundation) und die herstellereinspezifischen Spezifikationen (keine Veröffentlichungsrichtlinien) nutzen diese Bausteine, um domänen- und anwendungsspezifische Datenmodelle zu erstellen. [3, 10]

Diese Datenmodelle stellen zusätzlich zu den übertragenen Daten den Kontext bereit: die Einheit des Datums sowie Informationen, wie es mit anderen Daten derselben Quelle zusammenhängt, etwa von welchem Sensor ein Wert aufgenommen wurde und welchen Wertebereich der Sensor abdeckt. Das DI Model stellt allgemeine Bausteine zur Verfügung, um Geräte und Komponenten und ihre wichtigsten Eigenschaften, wie etwa Identifikationsdaten, zu beschreiben. So kann sichergestellt werden, dass die Identifikationsdaten von Geräten unterschiedlicher Domänen

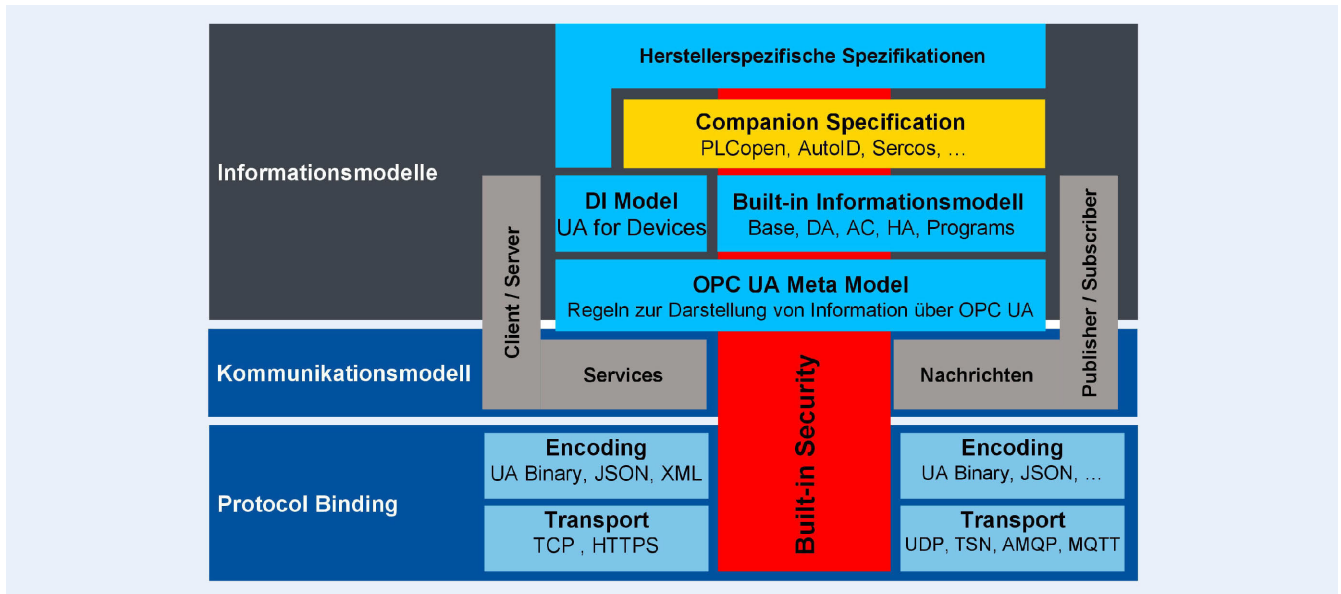


Bild 2. Architekturelle Struktur von OPC UA. Grafik: Stefan Hoppe, VDMA Interoperability Day 2018

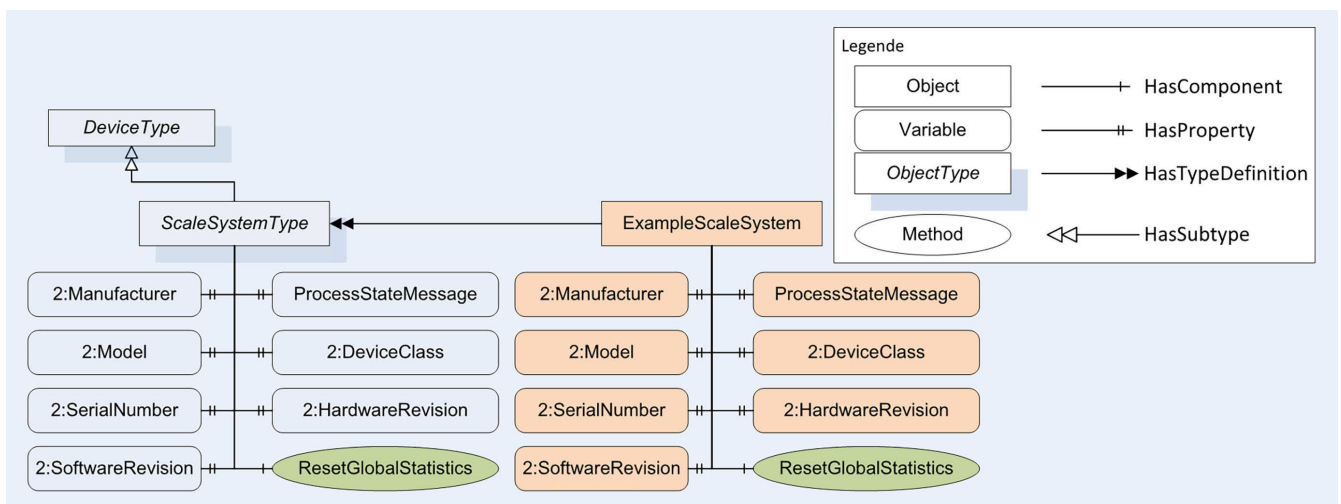


Bild 3. Elemente der OPC UA for Weighing Technology, für welche verschiedene Arten von Testfällen definiert werden. Grafik: Universität Stuttgart, ISW

und Hersteller als semantisch identisch identifiziert werden können. [4]

Ein Beispiel für eine Companion Specification ist OPC 40200: Weighing Technology. Darin wird ein Modell für Waagen und Wägesysteme beschrieben. [11]

## 2.2 Testfälle in OPC UA

Für alle Aspekte der in Bild 2 dargestellten Architektur gibt es Testfälle, die von der OPC Foundation organisiert werden. Dieser Beitrag fokussiert auf Testfälle für die Companion Specifications. Alle Testfälle werden für die Zertifizierung von OPC-UA-Produkten durch von der OPC Foundation akkreditierte Zertifizierungsstellen verwendet [12]. Die Testfälle werden veröffentlicht und stehen so Entwicklern von OPC-UA-Produkten und Integratoren, die die OPC-UA-Kommunikation in die unternehmensinterne Dateninfrastruktur integrieren, zur Verfügung [13].

Im Gegensatz zu statischen Testfällen sind die OPC-UA-Testfälle als dynamische Tests zu verstehen. Sie testen Programme

während der Ausführung. Sie werden auch als Black-Box-Tests bezeichnet: Im Gegensatz zu White-Box-Tests ist der Programmcode dem Tester nicht bekannt. Es wird geprüft, ob die OPC-UA-Schnittstelle in der erwarteten Weise funktioniert. Wie diese Funktionalität im Programm bereitgestellt wird, ist nicht Gegenstand der Testfälle. [14]

Bei Companion Specifications beziehen sich die Testfälle sowohl auf die korrekte Abbildung des Datenmodells in Form von Typ- und Instanztests als auch auf die korrekte Abbildung des Modellverhaltens [15]. Am Beispiel von OPC 40200 in Bild 3 sind diese drei Aspekte farblich hervorgehoben.

## 2.3 Testfälle für OPC UA Companion Specifications am Beispiel der OPC 40200

Der „ScaleSystemType“ ist in Bild 3 blau dargestellt. Dieser Typ muss in OPC-UA-Produkten korrekt abgebildet werden, es müssen also die HasSubtype-Beziehung zum DeviceType und die HasComponent-/HasProperty-Beziehungen zu den sieben Varia-

blen und der Methode „ResetGlobalStatistics“ dargestellt werden. Ähnlich verhält es sich mit den Instanztests. Durch die Typen kann ein OPC-UA-Client die Strukturelemente jedes OPC-UA-Servers kennenlernen, ohne sie vorher programmieren zu müssen.

Bei der rot dargestellten Instanz „ExampleScaleSystem“ muss die Beziehung zum ScaleSystemType abgebildet sein. Es müssen mindestens die sieben dargestellten Variablen mit HasProperty vorhanden sein. Aber es dürfen dem ExampleScaleSystem auch mehr Elemente – auch zusätzlich zur OPC 40200 definiert – zugeordnet werden. Die Methode ResetGlobalStatistics ist als optional definiert, sie muss also der Objektinstanz nicht zugeordnet sein. Wenn sie zugeordnet ist, muss sie mit der Definition in der Spezifikation übereinstimmen. Dadurch kann ein OPC-UA-Client direkt auf die Datenpunkte zugreifen, die aus der Spezifikation bekannt sind.

Bei den Tests zur Abbildung des Datenmodells beziehen sich die Testinhalte direkt auf Elemente des OPC-UA-Metamodells (wie etwa Object, Variable, HasComponent). Bei den Verhaltenstests hingegen ist die zu testende Eigenschaft im beschreibenden Freitext der Spezifikation enthalten. Für ResetGlobalStatistics, grün in Bild 3, wird zum Beispiel als Vorbedingung getestet, ob es im Typ vorhanden ist. Dies ist keine Voraussetzung, da die Spezifikation die Methode nicht für jede Instanz vorschreibt. Ebenso können Nachbedingungen getestet werden, wie etwa die Anforderung, dass die globale Statistik nach dem Aufruf der Methode einen Nullwert enthält. Das Verhalten kann nicht nur für Methoden getestet werden, sondern auch für Variablen, die in einem kausalen Zusammenhang stehen, zum Beispiel Zähler für produzierte Teile und durchgeführte Programmdurchläufe, die gleichzeitig höher zählen.

### 3 Testfälle als Garant für Interoperabilität

Dieses Kapitel untersucht zunächst, wie verschiedene Benutzerrollen mit Testfällen umgehen und von ihnen profitieren. Anschließend wird untersucht, welche Vorteile für die innerbetriebliche Dateninfrastruktur zu erwarten sind.

#### 3.1 Aspekte der Nutzung von OPC UA Testfällen durch verschiedene Benutzerrollen

Für OPC-UA-Datenmodelle ist es möglich die Typen, die Instanzen und das Verhalten zu testen. Zudem kann betrachtet werden, wer die Testfälle ausführt.

Eine Zertifizierungsstelle verwendet die Tests, um die korrekte und vollständige Umsetzung der Spezifikation sicherzustellen. Im Rahmen der Zertifizierung wird auch geprüft, ob die Produkte mit anderen OPC-UA-Geräten im Kontext funktionieren, also ein definiertes Mindestmaß an Interoperabilität bieten.

Entwickler von OPC-UA-Produkten können mit den Testfällen eine falsch konfigurierte Entwicklungsumgebung identifizieren. Die Typen können in vielen Entwicklungsumgebungen aus der Spezifikation importiert werden. Alternativ müssen sie vom Entwickler identisch zur Spezifikation erstellt werden. Die zugehörigen Tests zeigen, ob ein Fehler vorliegt. Ähnlich verhält es sich mit den Instanzen. Diese können meist in der Entwicklungsumgebung automatisch aus den Typen erzeugt und dann vom Entwickler parametrisiert werden. Stellen die Tests Inkonsisten-

zen zum Typ fest, muss entweder die Entwicklungsumgebung angepasst werden oder sie wurde falsch verwendet.

Bei Verhaltenstests ist keine Automatisierung seitens der Entwicklungsumgebung möglich: der Entwickler muss die Spezifikation lesen, verstehen und für sein Produkt umsetzen. Hier können Tests zum benötigten Verständnis beitragen.

Beide Aspekte der korrekten Anwendung der Spezifikation durch die Entwickler von OPC-UA-Produkten sind grundlegend für die spätere Interoperabilität.

Aus Sicht der Integratoren können die Testfälle auch der Qualitätskontrolle aus Kundensicht dienen. Die korrekte Darstellung der Typen gibt den Integratoren die Sicherheit, dass die Werte die gleiche Bedeutung haben. Bild 3 zeigt Variablen, deren Name mit einer 2 beginnt, zum Beispiel „2:Manufacturer“. Alle diese Variablen sind nicht in OPC 40200 definiert, sondern in der generischen OPC 10000–100 und werden entsprechend auch von anderen Domänenspezifikationen verwendet [4]. Ein Integrator sieht also „2:Manufacturer“ sowohl bei der Waage als auch bei Robotern und Fräsmaschinen in seinem Maschinenpark. So kann er sicher sein, dass immer der Herstellername des Gerätes angegeben wird und zusätzlich der Datentyp immer im gleichen Format ausgelesen werden kann. Ähnliches gilt für Instanzen und das Schnittstellenverhalten: Durch die Tests kann sich der Integrator darauf verlassen, dass seine auf der Spezifikation basierenden Erwartungen erfüllt werden. Somit besteht Interoperabilität im Sinne einer stillschweigenden Übereinkunft zwischen Entwickler und Integrator durch die Spezifikation.

#### 3.2 Bedeutung von OPC UA Testfällen

Die Interoperabilität muss im Gesamtkontext der unternehmensinternen Dateninfrastruktur Bestand haben. Für die Praxis im Betrieb vernetzter Produktionsanlagen wurden in [16] Probleme in der Praxis identifiziert. Dazu gehören Komplexität (wie Umgang mit unterschiedlicher Semantik in verschiedenen Systemen), Variabilität (wie etwa Umgang mit Produktvarianten und Überblick über deren Kompatibilität) und Wissensmanagement (Wissen und dessen Weitergabe durch Verknüpfung domänen-spezifischer Informationsquellen).

Hier trägt die Typprüfung zur Handhabung der Komplexität bei. Generell ist die Standardisierung mit den OPC UA Companion Specifications ein wichtiger Schritt in Richtung einer übergreifenden Semantik; OPC 10000–100 ist ein gutes Beispiel [4]. Durch die Tests und die Gewissheit, dass alle Aspekte korrekt umgesetzt sind, kann die Komplexität in der Datenübertragung mit OPC UA effektiv nach außen gekapselt werden. Fehlerfaktoren wie Übertragungs- und Flüchtigkeitsfehler in der Entwicklung können so minimiert werden.

Für Entwickler und Integratoren bieten die Testfälle einen Anhaltspunkt für das Variantenmanagement. Durch das Testen der Instanzen kann überprüft werden, welche Implementierungsoptionen für welche Produktvarianten umgesetzt und verfügbar sind.

Die Verwendung von Verhaltenstests durch die Entwickler trägt zum Wissensmanagement bei: die Tests vermitteln ihnen die Intention der Spezifikation. Daher ist die Bereitstellung von Testfällen für jede Companion Specification ein wichtiger Aspekt der Verschriftlichung von Domänenwissen. Außerdem steht dieses Wissen dem Empfänger in einer Form zur Verfügung, auf die er im Bedarfsfall (bei fehlgeschlagenen Tests) wesentlich interakti-



ver zugreifen kann als etwa auf die Dokumentation durch die Spezifikation.

## 4 Zusammenfassung und Ausblick

Das folgende Kapitel fasst die behandelten Themen zusammen und ordnet den Nutzen von OPC-UA-Testfällen in einer unternehmensinternen Infrastruktur ein.

### 4.1 Zusammenfassende Einordnung OPC UA Testfälle

OPC UA Testfälle bieten für die Zertifizierung sowie für die Entwicklung und den Einsatz von OPC-UA-Produkten eine Möglichkeit, Fehler in den Produkten aufzudecken. Damit legen sie die Grundlage für die Interoperabilität dieser Produkte und tragen zu deren Handhabbarkeit in der vernetzten Produktion bei. Allerdings sind den Testfällen und ihrem Einfluss klare Grenzen gesetzt.

Die Testfälle zu einer OPC-UA-Spezifikation werden von der Spezifikationsgruppe manuell erstellt. Dies führt zu Unterschieden in Vollständigkeit und Art der Testfälle zwischen verschiedenen Spezifikationen. Alle genannten Vorteile gelten nur, wenn auch entsprechende Testfälle definiert wurden.

Spezifikationstestfälle beziehen sich per Definition auf einen begrenzten Horizont. Im Gesamtsystem können die zu testenden Funktionalitäten jedoch weitreichender sein: Ein über OPC UA übertragenes Programm soll nicht nur über die OPC-UA-Schnittstelle als empfangen angezeigt werden, sondern zum Beispiel auch tatsächlich auf der Steuerung des entsprechenden Gerätes zur Ausführung zur Verfügung stehen. Um solche Zusammenhänge zu testen, sind zum Teil produktspezifische Tests nötig.

Bei der Entwicklung von OPC-UA-Spezifikationen und -Produkten kann der Aufwand durch Automatisierung minimiert werden. Entwickler werden entlastet, wenn ihre Umgebungen in der Lage sind, Spezifikationen zu importieren und aus den Typen automatisch korrekte Instanzen zu erzeugen. Die Qualität der OPC-UA-Produkte steigt mit der Zugänglichkeit, Vollständigkeit und Qualität der Testfälle. Deren Vollständigkeit und Qualität wiederum steigt, wenn auch die Testfallerstellung automatisiert ist, insbesondere im Bereich der Typ- und Instanztests.

### 4.2 Zusammenführung in den Gesamtkontext

Durch die genannten Vorteile der Verwendung von OPC UA Testfällen in Verbindung mit der Dateninfrastruktur aus Bild 1 werden Abhängigkeiten deutlich. Durch die genutzten Testfälle kann eine höhere Datenqualität in der Produktion sichergestellt werden. Unabhängig davon, ob die Produktionsdaten zuvor gefiltert wurden oder nicht, wird das Risiko fehlerhafter Daten in den Asset Administration Shells durch die Verwendung von Testfällen minimiert. Auf diese Weise kann eine höhere Sicherheit für die Nutzung und Verwendung der eigenen Daten gewährleistet werden. Für die in Kapitel 1 genannten Beispiele, Carbon Footprint und die Qualitätsdaten in der Produktion, ist es zum Beispiel unerlässlich, dass als Basis von allen Geräten gleichartige Daten zur Verfügung stehen. Auch durch die Bereitstellung von Daten über AAS und EDC können externen Unternehmen über eine qualitative Datenbasis verfügen.

Zusammenfassend bieten Testfälle innerhalb der Kommunikation mit OPC UA sowohl für Entwickler als auch für Integrato-

ren einen deutlichen Nutzen. Sie können im Kontext einer unternehmensinternen Dateninfrastruktur eingesetzt werden, die ihre Produktion in OPC UA abbildet und Asset Administration Shells verwendet. Dazu muss aber eine klare Integration der Testfälle in diese Dateninfrastruktur geschaffen werden, die heute in dieser Form nicht existiert.

### FÖRDERHINWEIS


Im Forschungsprojekt „Entwicklung eines Expertensystems für eine informationstechnische Automatisierung der Erstellung von OPC UA Testfällen und -skripten in Industrial Internet of Things Anwendungen“ wird die automatisierte Erstellung von Testfällen thematisiert. Dieses Projekt wird vom Bundesministerium für Wirtschaft und Klimaschutz (BMWK) aufgrund eines Beschlusses des Deutschen Bundestages gefördert.

### Literatur

- [1] Catena-X Automotive Network: EDC – Einführung. Internet: [catena-x.net/de/angebote/edc-die-zentrale-komponente-fuer-die-navigation](https://catena-x.net/de/angebote/edc-die-zentrale-komponente-fuer-die-navigation). Zugriff am 03.05.2023
- [2] Plattform Industrie 4.0.: Specification. Details of the Asset Administration Shell Part 1 (Version 3.0RC02). Veröffentlicht am 01.05.2022. Internet: [www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details\\_of\\_the\\_Asset\\_Administration\\_Shell\\_Part1\\_V3.pdf?\\_\\_blob=publicationFile&v=1](https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.pdf?__blob=publicationFile&v=1). Zugriff am 03.05.2023
- [3] OPC Foundation: OPC 10000–1: UA Part 1: Overview and Concepts. 1.05.02. Veröffentlicht am 01.11.2022. Internet: [opcfoundation.org/developer-tools/documents/view/158](https://opcfoundation.org/developer-tools/documents/view/158). Zugriff am 03.05.2023
- [4] OPC Foundation: OPC 10000–100: Devices. 1.04. Veröffentlicht am 03.11.2022. Internet: [opcfoundation.org/developer-tools/documents/view/197](https://opcfoundation.org/developer-tools/documents/view/197). Zugriff am 03.05.2023
- [5] IEC 62541–5:2020: OPC Unified Architecture – Part 5: Information Model. Ausgabe Juli 2020
- [6] OPC Foundation: OPC 10000–4: UA Part 4: Services. 1.05.02. Veröffentlicht am 27.10.2021. Internet: [opcfoundation.org/developer-tools/documents/view/161](https://opcfoundation.org/developer-tools/documents/view/161). Zugriff am 03.05.2023
- [7] OPC Foundation: OPC 10000–6: UA Part 6: Mappings. 1.05.01. Veröffentlicht am 01.03.2022. Internet: [opcfoundation.org/developer-tools/documents/view/163](https://opcfoundation.org/developer-tools/documents/view/163). Zugriff am 03.05.2023
- [8] OPC Foundation: OPC 10000–14: UA Part 14: PubSub. 1.05.02. Veröffentlicht am 01.11.2022. Internet: [opcfoundation.org/developer-tools/documents/view/171](https://opcfoundation.org/developer-tools/documents/view/171). Zugriff am 03.05.2023
- [9] OPC Foundation.: OPC 10000–2: UA Part 2: Security. 1.05.02. Veröffentlicht am 01.11.2022. Internet: [opcfoundation.org/developer-tools/documents/view/159](https://opcfoundation.org/developer-tools/documents/view/159). Zugriff am 03.05.2023
- [10] OPC Foundation.: OPC 10000–5: UA Part 5: Information Model. 1.05.02. Veröffentlicht am 01.11.2022. Internet: [opcfoundation.org/developer-tools/documents/view/162](https://opcfoundation.org/developer-tools/documents/view/162). Zugriff am 03.05.2023
- [11] OPC Foundation.: OPC 40200: Weighing Technology. 1.00. Veröffentlicht am 01.06.2020. Internet: [opcfoundation.org/developer-tools/documents/view/237](https://opcfoundation.org/developer-tools/documents/view/237). Zugriff am 03.05.2023
- [12] OPC Foundation.: Overview & Benefits. Internet: [opcfoundation.org/certification/overview-benefits/](https://opcfoundation.org/certification/overview-benefits/). Zugriff am 03.05.2023
- [13] OPC Foundation.: OPC UA Profile Reporting Application. Internet: [profiles.opcfoundation.org/](https://profiles.opcfoundation.org/). Zugriff am 03.05.2023
- [14] Witte, F.: Testmanagement und Softwaretest – Theoretische Grundlagen und praktische Umsetzung. Wiesbaden: Springer Vieweg 2019, S. 84–86
- [15] Friedl, S.; Heinemann T.; Lechler A. et al.: Implicit Templates for Conformance Units in OPC UA Companion Specifications. 2021 22nd IEEE International Conference on Industrial Technology (ICIT), Valencia, Spain, 2021, pp. 730–735
- [16] Feichtinger, K.; Meixner, K.; Rinker, F. et al.: Industry Voices on Software Engineering Challenges in Cyber-Physical Production Systems Engineering. Proceedings of the 2022 IEEE 27th International Confe-

rence on Emerging Technologies and Factory Automation (ETFA),  
Stuttgart, Germany, 2022, pp. 1–8



**Tonja Heinemann**, M.Sc.   
*Foto: Universität Stuttgart, ISW*

**Samed Ajdinović**, MSc. 

Dr.-Ing. **Armin Lechler** 

Prof. Dr.-Ing. **Oliver Riedel** 

Institut für Steuerungstechnik der Werkzeugmaschinen  
und Fertigungseinrichtungen (ISW)  
Universität Stuttgart  
Seidenstr. 36, 70174 Stuttgart  
Tel. +49 711 / 685-84626  
tonja.heinemann@isw.uni-stuttgart.de  
www.isw.uni-stuttgart.de

## LIZENZ



Dieser Fachaufsatz steht unter der Lizenz Creative Commons  
Namensnennung 4.0 International (CC BY 4.0)