

# Vorgehensweise zur Implementierung der VDMA Spezifikation „OPC UA for Machinery (OPC 40 001)“ in Maschinen und IT-Systemen erarbeiten

Bachelorarbeit

Angewandte Informatik

Duale Hochschule Baden-Württemberg Mosbach

Studienpartner

AZO GmbH & Co. KG



von

Rico Kursidem

betreut von

Steffen Michalowski

und

Prof. Dr. Mielebacher

23. Juni 2023

# Abkürzungsverzeichnis

**API** Advanced Programming Interface

**COM** Component Object Model

**DCOM** Distrebuted Component Object Model

**HMI** Human Maschine Interface

**IoT** Internet of Things

**M2M** Maschine-zu-Maschine

**MES** Manufacturing Execution System

**OPC UA** Open Platform Communications Unified Architecture

**SOA** Service orientierte Architektur

**TCP/IP** Transmission Control Protocol/Internet Protocol

**UMATI** Universal machien technology interface

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>2</b>
1.1	Problemstellung und Ziel . . . . .	3
1.2	Unternehmen . . . . .	4
1.3	Forschungsfragen . . . . .	4
1.4	Aufbau der Arbeit . . . . .	5
<b>2</b>	<b>Grundlagen</b>	<b>6</b>
2.1	OPC UA . . . . .	6
2.1.1	Struktur . . . . .	8
2.1.2	Kommunikationsaufbau . . . . .	9
2.1.3	Sicherheit . . . . .	10
2.2	Umati . . . . .	10
2.2.1	VDMA . . . . .	10
2.2.2	Anforderungen . . . . .	10
2.2.3	OPC UA 400 01 . . . . .	10
<b>3</b>	<b>Methodik</b>	<b>11</b>
3.1	Literaturrecherche . . . . .	11
3.2	Evaluation . . . . .	11
3.3	Experteninterview . . . . .	11
3.4	Zeitplanung . . . . .	11
<b>4</b>	<b>Ergebnisse</b>	<b>12</b>
4.1	Anforderungsanalyse . . . . .	12
4.2	Lösungsansätze . . . . .	12
4.3	Marktanalyse . . . . .	12
4.4	Wirtschaftlichkeit und Projektanalyse . . . . .	12
4.4.1	Kostenplanung . . . . .	12

## Inhaltsverzeichnis

4.4.2	Fallbeispiel . . . . .	12
4.5	Implementierung eines Prototyps . . . . .	12
4.5.1	Aufbau der Umgebung . . . . .	12
4.5.2	Implementierung . . . . .	12
4.5.3	Integration in bestehende Infrastruktur . . . . .	12
<b>5</b>	<b>Diskussion und Fazit</b>	<b>13</b>

# Abbildungsverzeichnis

2.1	Die vier Ebenen der Interoperabilität . . . . .	7
2.2	Aufbau des OPC UA Standards, Angelehnt an: Abbildung 1.7 in [2] . . .	9
2.3	Kommunikationsstruktur OPC UA . . . . .	10

# Tabellenverzeichnis

# **Abstract**

**Deutsch**

**English**

# 1 Einführung

Zunehmend leben wir in einer immer stärker vernetzten Welt. Globalisierte Unternehmen mit Lieferketten, die den ganzen Planeten umspannen. Unternehmen mit Kunden und Zulieferern in verschiedensten Ländern. In der Industrie ist es dabei besonders wichtig, dass alle Teilnehmer an Wertschöpfungsketten, Arbeitsgruppen und Verträgen dasselbe Verständnis von Daten haben. Wir versuchen möglichst viele Bereiche der Produktion und Kommunikation zu standardisieren und somit weltweite Verständigung mit minimalen Missverständnissen zu erreichen. In Zeiten der Industrie 4.0 nehmen Standardisierungen einen noch wichtigeren Platz ein. Die Vernetzung von Fertigungsanlagen über Unternehmensgrenzen hinaus und zwischen verschiedenen Anlagen von unterschiedlichen Herstellern.

Standardisierung wird verwendet, um effizienteren Austausch und effektivere Prozesse zu ermöglichen. Ein sehr altes Beispiel sind Sprachen. Eine Menge an Regeln, welche zur Kommunikation genutzt werden, auf die sich alle Teilnehmenden geeinigt haben. In dieser Arbeit soll es um die Standardisierung im Bereich der Maschine-zu-Maschine (M2M) Kommunikation gehen. Durch die Herausforderungen von Industrie 4.0 kann ein guter Standard die Integration mehrerer Systeme vereinfachen und Kosten sparen.

In der M2M Kommunikation existieren bereits Standards, die schon vor Industrie 4.0 entworfen wurden. Dazu gehört auch Open Platform Communications Unified Architecture (OPC UA). Umati baut auf diesem Standard auf und verspricht weitere Aspekte wie beispielsweise eine Umgebung in der neue Maschinen durch *Plug & Play* integriert werden können. Das Ziel dieser Arbeit ist es, die Versprechen von Umati zu überprüfen und in seinem derzeitigen Stand zu bewerten, ob ein Mehrwert aus diesem Standard entspringt, der in der Industrie 4.0 Tragweite hat.



### 1.1 Problemstellung und Ziel

Durch die zunehmende Automatisierung der Produktion und Fertigung stehen viele Produzenten vor der Aufgabe, zahlreiche Maschinen verschiedener Hersteller in ihren Fertigungsprozess zu integrieren. Viele dieser Maschinen nutzen verschiedene Kommunikationsprotokolle, Datenformate und Schnittstellen, was die Integration dieser diversen Systeme erschwert. Durch eine komplizierte Integration kann Vendor Locking entstehen. Die Bindung eines Kunden an einen Anbieter, da dieser eine Technologie integriert, die andere Hersteller nicht oder nur extrem schwer integrieren können. Dies kann zu einer Atmosphäre führen, welche für den Kunden weniger Flexibilität und Kontrolle über seine eigene Produktion gibt und gleichzeitig die Kosten steigen lässt. Um dieses Problem zu lösen, müssen sich die Produzenten zu einem Standard zusammenfinden und die Interoperabilität ihrer Systeme gewährleisten.

Für AZO als Anlagenproduzent ist es von großem Interesse, mit Anlagen und Maschinen anderer Hersteller kommunizieren zu können und dabei die Integration möglichst einfach zu realisieren. Auch für Entwickler von (! (!)MES) bedeutet die Standardisierung von Kommunikationsprotokollen eine schnellere und einfachere Entwicklung, was zu niedrigeren Entwicklungskosten führt und die Robustheit der Systeme erhöhen kann. AZO hat, um diese Standardisierung zu erreichen, an der Ausarbeitung von Universal machine technology interface (UMATI) zugesagt. Dieser Standard möchte die Kommunikation zwischen Maschinen und der darüber liegenden Datenverarbeitung definieren und basiert auf dem offenen Standard OPC UA.

Das Ziel dieser Arbeit ist es, UMATI in einem Testumfeld aufzubauen und eine Proof of Konzept System aufzubauen. Um dies zu erreichen, soll zunächst eine Marktanalyse durchgeführt werden und mögliche Lösungen zum Versenden und Empfangen von Daten zwischen Maschine und MES zu finden. Diese sollen anhand von selbst entwickelten Kriterien abgewogen und bewertet werden. Die vielversprechendsten Umsetzungen sollen in einem Testsystem implementiert werden und für zukünftige Demonstrationen auf Messen für AZO zur Verfügung gestellt werden. Außerdem soll evaluiert werden, ob UMATI auch in der Low-Code Plattform Node Red angewendet werden kann und wenn möglich, das entstehende System als Open-Source Lösung der Öffentlichkeit zur Verfügung zu stellen.

Der Prototyp soll ebenfalls eine Integration in ACAS umfassen. Damit kann gezeigt werden, dass UMATI eine Möglichkeit ist, die Kommunikation von AZO Anlagen und

ACAS zu realisieren. Diese Integration muss im Nachgang weiter evaluiert werden, um festzustellen, ob UMATI als Standardisierung eine gute Lösung ist.

## 1.2 Unternehmen

Diese Arbeit wird mit dem Dualen Partner *AZO GmbH & Co. KG* durchgeführt. AZO bietet maßgeschneiderte Lösungen für die automatisierte Förderung, Lagerung und Dosierung von Rohstoffen weltweit an. Dabei werden Anlagen für die Bereiche der Chemie-, Nahrungsmittel-, Pharma-, Kosmetik und Kunststoffindustrie gefertigt. Diese Projekte umfassen die Planung, Fertigung und Montage, sowie die Automatisierung der Anlagen.

AZO entwickelt ein eigenes MES-System ACAS, welches die Kommunikation zwischen Anlagen von AZO, aber auch Anlagen von anderen Herstellern, mit den Kunden vereinheitlichen und verbessern soll. Dabei soll ein umfassendes System entstehen, welches Steuerung, Visualisierung und Überwachung der Produktion übernehmen und in einer zentralen Software bündeln soll.

ACAS entsteht in der Entwicklungsabteilung, welche ebenfalls an dieser Arbeit beteiligt ist. Sie fokussiert sich auf die Automatisierung von AZO Anlagen im Bereich von SPS, Steuerungen bis zur Datenverarbeitung und Erhebung auf abstrahierter Ebene.

## 1.3 Forschungsfragen

Die Standardisierung der Integration von verschiedensten Maschinen in eine Umgebung ist nötig und kann zu verstärkten Effizienzsteigerungen und Aufwandsersparnis führen. Für Anlagenbauer verspricht die Nutzung von Umati, diese Integration in einen schnellen und effizienten Prozess zu wandeln. Um das Potenzial von Umati zu evaluieren, wurden folgende Schwerpunkte betrachtet:

- **Q1:** Welche Herausforderungen ergeben sich bei der Umsetzung von Umati?
- **Q2:** Welche Lösungen zur Umsetzung von Umati sind am Wirtschaftlichsten?
- **Q3:** Wie kann die Datenkommunikation von Anlage und ACAS bestmöglich implementiert werden?

### **1.4 Aufbau der Arbeit**

Nachfolgend werden Thematiken beschrieben die zum grundlegenden Verständnis der Arbeit beitragen (Kapitel 2). Daraufhin werden die Methodiken beschrieben, mit denen die Informationen für diese Arbeit erhoben wurden (Kapitel 3). Im Hauptteil der Arbeit (Kapitel 4) wird der Ablauf beschrieben. Es werden zunächst Informationen zur Entscheidungsfindung eingeholt und in Analysen in Relation gesetzt und daraufhin wird die Entwicklung des Prototyps beschrieben. Die Arbeit schließt mit einer Diskussion der gefundenen Ergebnisse und einem Fazit (Kapitel 5).

## 2 Grundlagen

### 2.1 OPC UA

OPC UA ist ein unternehmensunabhängiger, offener Standard zur Kommunikation von Informationen und Daten zwischen Maschinen im industriellen Umfeld. Das Protokoll wurde 2008 von der OPC Foundation veröffentlicht und nimmt sich zur Aufgabe, die Interoperabilität von Systemen zu fördern. Es verwendet eine Service orientierte Architektur (SOA) und kann auf verschiedensten Betriebssystemen verwendet werden. OPC UA ist sehr gut skalierbar und kann deshalb als Kommunikationsprotokoll bei kleinen Internet of Things (IoT) Systemen als auch bei komplexen Cloud-Systemen zur Anwendung kommen.

OPC UA baut auf einem Server-Client-Modell auf, wobei der Client die Anfragen an den Server senden muss. Der Client fragt den Server nach Daten und analysiert diese während der Server stellt die Daten zur Verfügung. Diese Struktur ermöglicht eine einfache Skalierung von verschiedensten Geräten.

Das Ziel von OPC UA und auch im weitergetragenen Sinne von Umati ist das Erreichen von Interoperabilität im Industriellen Bereich. Interoperabilität beschreibt die Standardisierung von Kommunikation zwischen Systemen. Allgemein kann diese in 4 Stufen eingeteilt werden, wobei jede Ebene auf die darunterliegende aufbaut. Im Schaubild 2.1 sind die vier Ebenen Strukturelle, Syntaktische, semantische und Organisatorische Interoperabilität abgebildet.

Die strukturelle Interoperabilität beschreibt Verbindungen zwischen den Systemen, weshalb sie auch oft als Konnektivität bezeichnet wird. Der Datenaustausch kann nur erfolgen, wenn die beteiligten Systeme miteinander verbunden sind und ein geeignetes Advanced Programming Interface (API) implementieren. Diese Verbundenheit kann über ein Netzwerk mit TCP/IP oder HTTP erreicht werden oder kann ein Übertragungsmedium wie USB festlegen.

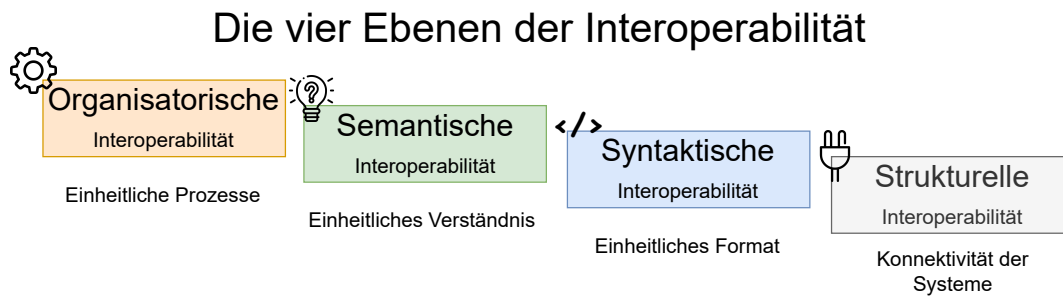


Abbildung 2.1: Die vier Ebenen der Interoperabilität

Sind Systeme syntaktisch interoperabel, nutzen und verstehen sie die Struktur der Daten. Hierfür können festgelegte Standards wie XML oder Json verwendet werden.

Die Semantische Interoperabilität beschreibt das vereinheitlichte Verständnis der Daten. Kommunizieren zwei Systeme miteinander, so müssen Sie unter denselben Daten auch dasselbe verstehen. Ein Beispiel für einen Standard der Semantische Interoperabilität umsetzt, ist der "International Classification of Diseases". Bei diesem geht es um die Kommunikation von Krankheiten zwischen Ärzten und wurde von der WHO eingeführt. Jede Krankheit hat einen festgelegten Code, wodurch eine eindeutige Kommunikation gewährleistet ist.

OPC UA legt anhand des Information-Models die Syntax und Semantik der Daten fest, wodurch die syntaktische Interoperabilität gewährleistet ist. Die semantische Interoperabilität wird anhand der Companion Standards erweitert. Diese beschreiben Technologiespezifische Informationsmodelle. Diese Standards entstanden durch intensive Kooperation und Zusammenarbeit mit Organisationen, um die Standards möglichst umfassend aufzustellen.

OPC UA entstand aus dem zuvor entwickelten Standard OPC Classics. Der alte Standard setzte die Kommunikation im Bereich der Automatisierung bereits sehr gut um. Der Nachteil an Classics war jedoch die fehlende Plattformunabhängigkeit. OPC UA unterstützt heute jegliche Betriebssysteme für Clients als auch für die Server. Der Vorgänger musste auf einem Windowssystem betreiben werden. Auch die Kommunikation war auf die von Microsoft entwickelte Technologien Component Object Model (COM) und Distrebuted Component Object Model (DCOM) basiert. Durch die Ablösung dieses Standards durch OPC UA wird über TCP/IP und SOAP kommuniziert, welche beide Plattformunabhängig sind und somit kann zunehmend OPC UA auch auf anderen Plattformen wie Linux, Mac und Android angewandt werden.

### 2.1.1 Struktur

OPC UA ist eine Sammlung von Standards, welche in unterschiedlichen Spezifikationen definiert sind. Diese gliedern sich in eine hierarchische Struktur, welche dann bei der Implementierung aufeinander aufbauen. In Abbildung 2.2 ist die Struktur des UA Framework abgebildet. Im unteren Teil und somit der Grundbaustein des Modells stehen Spezifikationen zum Transport und der Service Discovery. Diese beschreiben über welche Medien Daten transportiert wird und wie erkannt werden kann, welche Services angeboten existieren, die den OPC UA Standard implementieren. [2]

Zur Infrastruktur, welche im Allgemeinen die Konnektivität gewährleistet, ist auch die Spezifikation für den Datenzugang (Information Access) definiert. Außerdem werden auch weitere Standards beschrieben, um die Sicherheit und Robustheit des Systems zu gewährleisten.

Auf den Standards der Infrastruktur bauen die Spezifikationen der Informationsmodelle auf. Diese bestehen aus den offiziellen OPC UA Spezifikationen und den darauf aufbauenden Companion und Vendor Informationsmodellen. [2]

Die offiziellen Standards befassen sich mit dem allgemeinen Austausch von Daten. Hierzu gehören der Austausch von Echtzeitdaten (DA), historischen Daten (HA) und Alarmen (AC). Neben diesen Hauptfaktoren werden auch andere Standards festgelegt, die das offizielle OPC UA Information Modell bilden. [2]

Auf dem OPC UA Standard setzen die Industriestandardinformation Modelle auf. Diese sind von mehreren Teilhabern entwickelte Standards, die beispielsweise innerhalb einer bestimmten Branche oder Industrie-Gruppen verwendet werden. Diese Standards stehen in den Companion Spezifikationen. Beispielsweise gibt es einzelnen Spezifikationen für AutoID Systeme oder Spritzgussmaschinen, die Abwandlungen und Spezialisierungen haben, die auf ihren Bereich angepasst sind. [2]

Auf der obersten Ebene des Schaubilds in Abbildung 2.2 sind die Vendor Informationsmodelle dargestellt. Diese sind weitere Standards, die die Companion Standards erweitern und innerhalb einer Maschinenumgebung eines Erstellers umgesetzt werden. In diesen Standards sind Veränderungen individueller Hersteller definiert, damit eine weitere Spezialisierung über die Companion Spezifikationen hinaus stattfinden kann. [2]

## 2.1. OPC UA

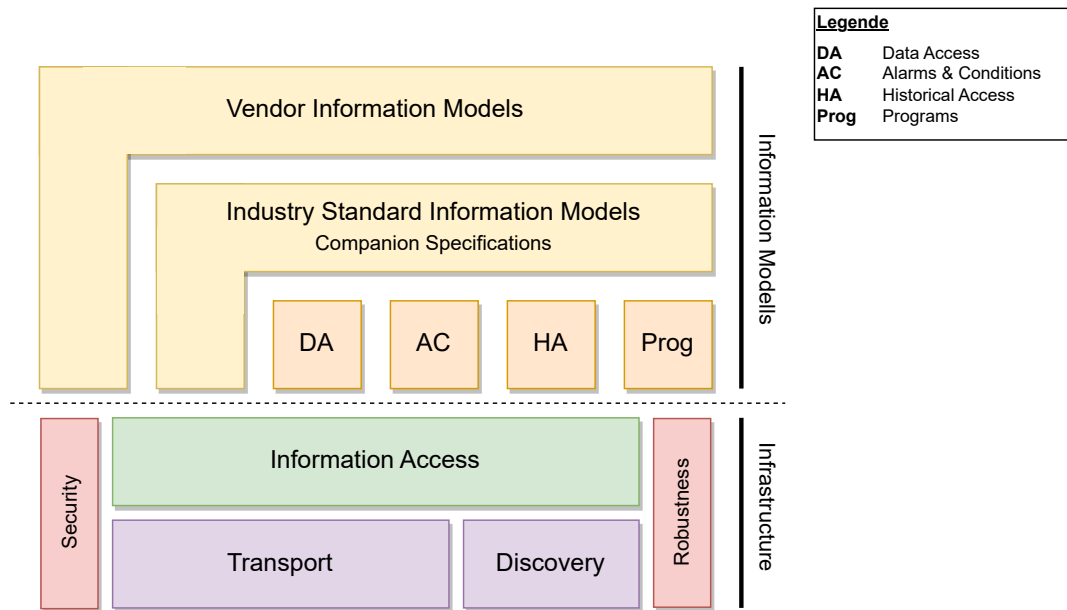


Abbildung 2.2: Aufbau des OPC UA Standards,  
Angelehnt an: Abbildung 1.7 in [2]

### 2.1.2 Kommunikationsaufbau

OPC UA besteht aus drei Komponenten, deren Zusammenwirkung in Abbildung 2.3 abgebildet ist. Mehrere Field Connections, die die Sensorik oder SPS darstellen, mit welchen kommuniziert werden soll. Die Daten sollen zu den OPC UA Clients geleitet werden. Dies können Human Maschine Interface (HMI) oder MES-Systeme sein, die die Daten der Maschinen auswerten oder Steuerungsbefehle absenden möchten. OPC UA standardisiert diese Kommunikation über den OPC UA Server. Dieser setzt den OPC UA Standard um und stellt die notwendigen APIs zur Verfügung.

Der OPC UA Server implementiert die proprietären Kommunikationsprotokolle, die vom Maschinenhersteller entwickelt wurden, um mit den Anlagen und den Field Connections zu kommunizieren. Je nach Art der Daten, die die Maschine weiter gibt, werden die Daten auf dem OPC UA Server abgespeichert oder an den Client weiter gegeben. Ein OPC UA Server kann in zwei Formen entstehen. Der Hersteller kann seiner Maschine die OPC UA Standards bereits bei der Entwicklung einprogrammieren. Damit ist der Server in der Maschine integriert und die Maschine ist so von Beginn an OPC UA fähig. Sollte der Hersteller die Spezifikationen jedoch nicht implementieren, kann der Server auch zusätzlich zwischen die Maschine und den Client geschaltet werden. Durch die Unabhängigkeit des Herstellers sind diese Server oft mit mehr Kommunikationsprotokollen

## 2.2. UMATI

ausgestattet, was deren Möglichkeiten, mit der Maschine zu interagieren erhöht.

Die Kommunikation mit dem Client läuft dabei 1 zu 1 ab. Allerdings kann auch ein 1 zu n Kommunikationsmodell über Publish and Subscribe implementiert werden. Um diese Funktion jedoch zu nutzen, muss der Server mit der OPC UA Pub/Sub Spezifikation erweitert werden.

Der Client kann die Daten über die standardisierte API des OPC UA Servers abfragen. Es können Echtzeitdaten, historische Daten abgerufen werden. Außerdem gibt es eine Spezifikation, wie Alarmierungslogik umgesetzt werden soll. Dies wird anhand der Implementation auf Serverebene deutlich vereinfacht, da sie dann Hersteller unabhängig ist.

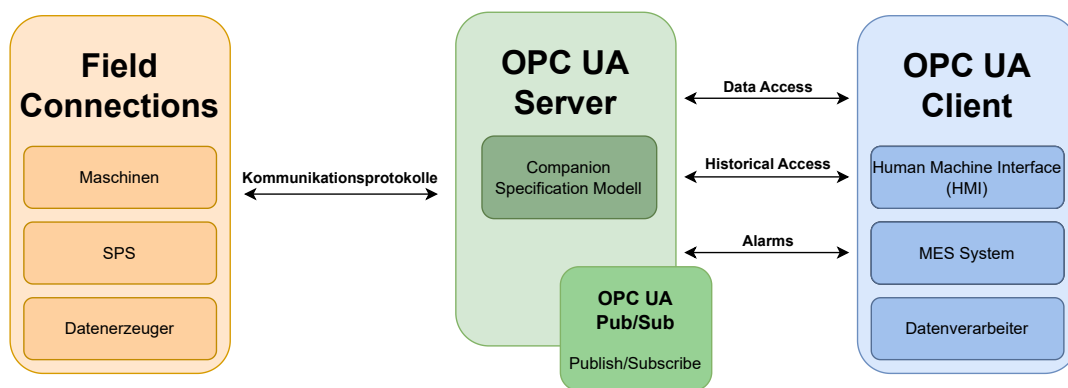


Abbildung 2.3: Kommunikationsstruktur OPC UA

### 2.1.3 Sicherheit

## 2.2 Umati

### 2.2.1 VDMA

### 2.2.2 Anforderungen

### 2.2.3 OPC UA 400 01



## **3 Methodik**

### **3.1 Literaturrecherche**

### **3.2 Evaluation**

### **3.3 Experteninterview**

### **3.4 Zeitplanung**

## **4 Ergebnisse**

### **4.1 Anforderungsanalyse**

### **4.2 Lösungsansätze**

### **4.3 Marktanalyse**

### **4.4 Wirtschaftlichkeit und Projektanalyse**

#### **4.4.1 Kostenplanung**

#### **4.4.2 Fallbeispiel**

### **4.5 Implementierung eines Prototyps**

#### **4.5.1 Aufbau der Umgebung**

#### **4.5.2 Implementierung**

#### **4.5.3 Integration in bestehende Infrastruktur**

## **5 Diskussion und Fazit**

# Literatur

- [1] Yinzhe Bao, Weimin Zhang und Ziwei Jia. *5G-based Cloud Learning Factory in AMTC*. en. SSRN Scholarly Paper. Rochester, NY, Feb. 2022. DOI: 10 . 2139 / ssrn.4072330. URL: <https://papers.ssrn.com/abstract=4072330> (besucht am 21.06.2023).
- [2] Wolfgang Mahnke, Stefan-Helmut Leitner und Matthias Damm. *OPC unified architecture*. eng. Berlin Heidelberg: Springer, 2009. ISBN: 978-3-540-68898-3 978-3-642-08842-1.
- [3] Jürgen Lange, Frank Iwanitz und Thomas J. Burke. *OPC: von Data Access bis Unified Architecture*. ger. 4., völlig neu bearb. und erw. Aufl. Berlin Offenbach: VDE Verlag, 2010. ISBN: 978-3-8007-3217-3.