

Analyse von Entscheidungsprozessen beim VDI/VDE 2193 Bieterverfahren

Sergej Grunau, Lukasz Wisniewski
Institute Industrial IT - inIT / TH-OWL

Campusallee 6
32657 Lemgo, Germany
sergej.grunau@th-owl.de
lukasz.wisniewski@th-owl.de

Abstract: Das Bieterverfahren nach VDI/VDE 2193 ermöglicht Verhandlungen zwischen Industrie 4.0-Komponenten. Jedoch ist die Umsetzung solcher Industrie 4.0-Komponenten, welche sich durch proaktive Verwaltungsschalen auszeichnen, nicht klar definiert. Besonders die Entscheidungsprozesse im Bieterverfahren, bei denen die Verhandlungsteilnehmer etwa über Angebotszu- oder absage bestimmen sollen, stellen eine Herausforderung in der Umsetzung dar. In dieser Arbeit wird ein konkreter Anwendungsfall aus der auftragsgesteuerten Produktion beschrieben. Anhand des Anwendungsfalls wird analysiert welche Kriterien für die Entscheidungsprozesse notwendig sind und wie diese und die Entscheidungsprozesse in proaktiven Verwaltungsschalen integriert werden können. Dazu zeigen wir eine mögliche Umsetzungsvariante proaktiver Verwaltungsschalen. Diese werden im wesentlichen mit Verwaltungsschalen-Applikationen und reaktiver Verwaltungsschalen realisiert, welche mittels Teilmodellen wie der hier beschriebenen „BiddingprocedureConfig“ miteinander interagieren können.

1 Einleitung

Interoperable I4.0-Komponenten können eine nahtlose Zusammenarbeit von smarten Maschinen und Fabriken innerhalb eines Unternehmens und über die Unternehmensgrenzen hinweg ermöglichen.

Eine Art von Zusammenarbeit sind Verhandlungen zwischen den Maschinen und Produkten um Dienstleistungen auszutauschen die z.B. einen oder mehrere Produktions-

schritte eines Produkts umsetzen sollen. Die Verhandlungen sollen dabei automatisiert ablaufen um eine flexiblere und effizientere Fertigung kundenspezifischer Aufträge zu ermöglichen, so wie in der Auftragsgesteuerten Produktion (siehe [1] [2]). Damit die Verhandlungen automatisiert ablaufen können müssen die Dienstleistungen hinreichend präzise und in einer maschinenlesbaren Form beschrieben sein und die Verhandlungsteilnehmer müssen eine gemeinsame und einheitliche Sprache sprechen. Dabei helfen Modelle und Beschreibungsmittel wie (i) die im Zuge der Plattform Industrie 4.0 standardisierte Verwaltungsschale [3], welche Industrie 4.0-Komponenten merkmalsbasiert beschreibt und (ii) die Sprache für I4.0-Komponenten [4, 5], welche im wesentlichen die Nachrichtenstruktur und Interaktion für ein Ausschreibungsverfahren zwischen I4.0-Komponenten entlang gewisser Regeln und Rollen spezifiziert. Außerdem spielen für die Verhandlungen nicht nur eine präzise Dienstbeschreibung eine Rolle, es müssen auch Kriterien wie Zeit, Kosten, Aufwand oder andere relevante Charakteristika in Betracht genommen werden. Des weiteren müssen die Erwartungen der Verhandlungsteilnehmer erfüllt sein wenn es um die Umsetzung der geforderten Dienstleistungen geht.

Diese Arbeit konzentriert sich auf die Entscheidungsprozesse beim Bieterverfahren während einer Verhandlung zwischen I4.0-Komponenten. Dazu soll ein konkretes Beispiel aus der auftragsgesteuerten Produktion in einer industriellen Umgebung betrachtet werden. Das Beispiel beschreibt die automatisierte Verhandlung um einen Lagerplatz zwischen einem Produkt und einem Lager und wird in Kap. 2 beschrieben. Danach werden in Kap. 3 die Entscheidungsprozesse im Interaktionsprotokoll nach dem Bieterverfahren analysiert und im Anschluss eine mögliche Umsetzungsvariante für eine I4.0-Umgebung beschreibend die im Wesentlichen mit Verwaltungsschalen-Applikationen und reaktiver Verwaltungsschalen realisiert, welche mittels des Teilmodells „BiddingprocedureConfig“ miteinander interagieren können. Im Anschluss wird die Arbeit zusammengefasst und ein Ausblick geschaffen.

2 Anwendungsfall

Beim Anwendungsfall handelt es sich um die Einlagerung eines Produkts in ein Lager. Für die Einlagerung soll zwischen dem Produkt und dem Lager eine Verhandlung stattfinden. Gegenstand der Verhandlung ist die Dienstleistung, die den Einlagerungsvorgang beschreibt. Das Ziel der Verhandlung für das Produkt ist es unter Berücksichtigung vorher bestimmter Kriterien einen Lagerplatz im Lager zu bekommen. Das Ziel des Lagers ist es einen Preis für ein Angebot anhand der Belegung zu bestimmen und eine Zusage für das abgegebene Angebot zu bekommen.

Dafür sollen eine I4.0-Umgebung umgesetzt werden die folgenden Anforderungen erfüllt:

- Das Produkt und das Lager haben ein proaktives Verhalten und führen die Verhandlungen automatisiert mit dem Bieterverfahren durch.
- Das Produkt und das Lager sollen I4.0 Komponenten sein und sich in einer I4.0 Umgebung befinden.

- Entscheidungen bei der Auswahl des Angebots sollen anhand der vorher bestimmten Kriterien getroffen werden (z. B. beim Erwerb des Produktes beim Verkäufer).

Zusätzlich sollen folgende Rahmenbedingungen eingehalten werden:

- Das Produkt hat eine Verwaltungsschale die bereits mit Informationen angereichert worden sind (z. B. durch den Entwickler oder Verkäufer des Produkts). Diese Informationen sind in Teilmodellen in der VWS modelliert.
- Die Dienstleistung ist als Teilmodell beschrieben und wird von beiden Parteien verstanden.
- Die VWS der Produkte werden als passive Verwaltungsschalen dem Produzenten übergeben und von einer VWS-Serverapplikation als reaktive VWS bereitgestellt.
- Es gibt eine Registry in der die VWS auffindbar sind.
- Die VWS des Lagers wird durch eine eigene VWS-Serverapplikation bereitgestellt und kann Logiken implementiert haben.

3 Analyse von Entscheidungsprozessen

In Abb. 1 wird das Interaktionsprotokoll des Bieterverfahrens nach dem das Produkt und das Lager interagieren dargestellt. Das Produkt nimmt die Rolle des Service Requester (SR) ein und das Lager die Rolle des Service Providers (SP).

Um proaktive VWS für den SP und SR umzusetzen, die das Interaktionsprotokoll nach dem Bieterverfahren umsetzen, sollten diese aus einem aktiven und einem passiven Teil bestehen (siehe [6]). Der aktive Teil einer Proaktiven VWS besteht aus Elementen wie einem Interaktionsmanager, einem Messenger und Algorithmen, die für die Entscheidungsprozesse während der Verhandlung benötigt werden.

Um die Rahmenbedingungen aus Kap. 2 einzuhalten werden für die Elemente Applikationen und Operationen implementiert welche die einzelnen Aufgaben während der Verhandlungen übernehmen.

Die Überprüfung der eingegangenen Ausschreibungen und die Berechnung des Preises für ein Angebot auf der SP Seite übernimmt eine Operation, die in der VWS-Serverapplikation des Lagers integriert wird. Sie prüft die Ressourcen über die der SP verfügt und bietet diese, wenn sie nicht belegt sind, als Angebot an.

Für die Entscheidung über die eingegangenen Angebote wird eine Auswahl-Applikation implementiert. Die Auswahl-App wählt eine der eingegangenen Angebote anhand der vorgegebenen Kriterien welche in der reaktiven VWS des SR festgelegt wurden aus und versendet eine Bestätigung an den SP.

Nachdem die Bestätigung eingegangen ist belegt eine Operation die Angebotene Ressource und bestätigt die Angebotsannahme.

Die Aufgabe des Interaktionsmanagers und des Messengers übernimmt die Bidding-Applikation. Sie generiert die Nachrichten für die Ausschreibung und für die Angebote

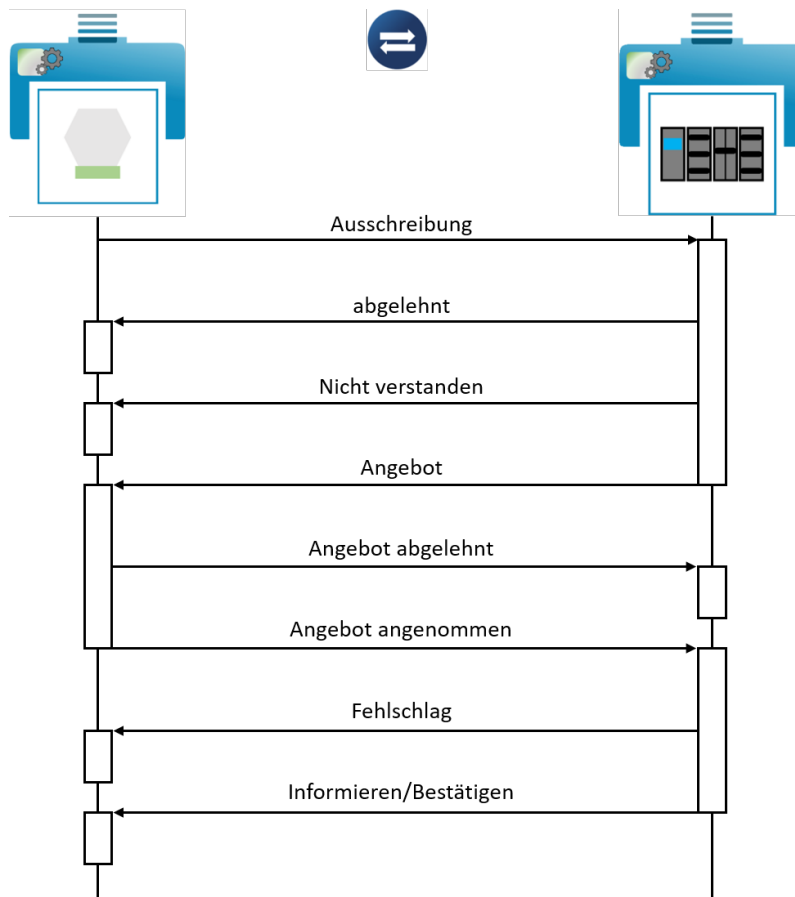


Abbildung 1: Verhandlungen zwischen Produkt und Lager mittels Bieterverfahren nach [4, 2]. Das Produkt (links) nimmt die Rolle des Service Requesters und das Lager die des Service Provider ein.

und tauscht diese mit anderen Verhandlungsteilnehmern aus. Dabei führt sie das Interaktionsprotokoll nach dem Bieterverfahren aus und kann während der Verhandlungen Teilmodelle des Applikationsnutzers (z. B. von der VWS des Produkts) verändern. Zusätzlich kann sie andere VWS-Applikationen und Operationen die oben genannt wurden aufrufen. Dazu soll ein Teilmodell modelliert werden das mit Teilmodellelementen die Interaktion zwischen den Applikationen für die Entscheidungsfindung ermöglicht.

4 Umsetzung der I4.0 Umgebung

Bei der Umsetzung der I4.0 Umgebung werden zunächst alle Komponenten beschrieben, die an der Verhandlung teilhaben. Dazu gehört wie sie umgesetzt wurden, welche Funktionalitäten sie haben und welche Aufgaben sie ausführen müssen. Danach wird beschrieben wie die einzelnen Komponenten miteinander interagieren und welche Informationen dabei ausgetauscht werden.

4.1 Komponenten in der I4.0 Umgebung

Die Umgebung in der die Verwaltungsschalen interagieren besteht aus mehreren Komponenten (siehe Abb. 2) die für die Verhandlung nötig sind.

Das Produkt ist der Service Requester (SR) und das Lager ist der Service Provider (SP) bei der Verhandlung um einen Lagerplatz. Die Verwaltungsschalen werden von verschiedenen VWS-Serverapplikationen bereitgestellt.

Die Serverapplikation für die Produkte greift auf AASX-Dateien zu und stellt diese als reaktive Verwaltungsschalen bereit. Nachdem die Verwaltungsschalen der Produkte bereitgestellt sind, werden sie in der Registry registriert. Die Serverapplikation welche die Verwaltungsschalen der Produkte bereitstellt ist der AASX-Server [7].

Die Verwaltungsschalen des Lagers, der Auswahl-App und der Bidding-App werden jeweils von einer eigenen AAS-Serverapplikation bereitgestellt. Anders als beim Produkt wurden diese in Java unter Verwendung des Eclipse BaSyx SDK [8] implementiert, das das VWS-Metamodell umsetzt und Funktionen wie die (De-)Serialisierung bereitstellt. Mit der Implementierung können bei der Umsetzung der VWS z. B. Operationen implementiert werden die Logik beinhalten. Diese Verwaltungsschalen haben ein proaktives Verhalten oder können eins mittels Interaktion mit einer reaktiven VWS generieren.

4.1.1 Bidding-App und Auswahl-App

Die Auswahl- und die Bidding-App haben als Asset eine Applikation und werden fortan VWS-Applikationen genannt. Mit der Interaktion zwischen einer VWS-Applikation mit einer reaktiven VWS wird ein proaktives Verhalten generiert.

Die Bidding-App implementiert Interaktionen von Verwaltungsschalen nach dem Bieterverfahren (siehe [4, 5]), wobei sie beide Seiten bedienen kann: SR und SP. Als SP bietet sie einem SR eine Dienstleistung an, wenn sein zugehöriges Asset fähig und verfügbar ist, um die Anfrage zu erfüllen. Während sie als SR versucht, einen geeigneten SP für die Dienstleistung zu finden, den sein zugehöriges Asset benötigt. Um die Bidding-App auszuführen wurden für den SP und SR jeweils eine Operation, die über die Rest-API der VWS aufgerufen werden kann, integriert. Die Bidding-App nutzt während der Verhandlungen den MQTT-Broker, um die Nachrichten auszutauschen.

Die Auswahl-App ist für die Auswahl eines geeigneten Angebots zuständig. Diese Auswahl ist ein Entscheidungsprozess im Bieterverfahren und wird auf der SR Seite aus-

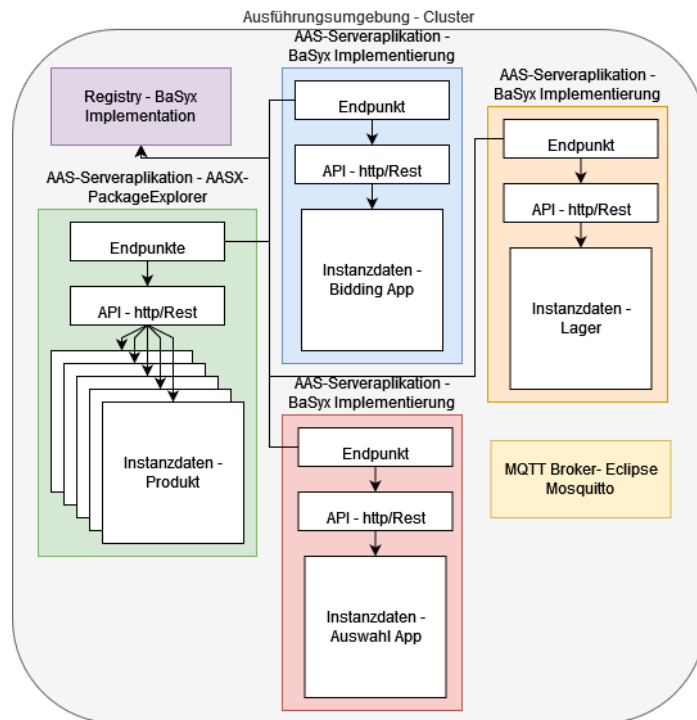


Abbildung 2: Die Abbildung zeigt alle an der Verhandlung beteiligten Komponenten zwischen Produkt (grün) und Lager (orange). Die BiddingApp (blau) und der MQTT-Broker sind für das einhalten des Interaktionsprotokolls und den Nachrichtenaustausch zuständig. Alle Komponenten (außer der MQTT-Broker) besitzen eine VWS und sind in der Registry (lila) mit ihren Endpunkten registriert. (frei nach [9])

geführt. Die Auswahl-App hat die Aufgabe aus allen eingegangenen Angeboten nach einem vorgegebenen Kriterium ein geeignetes auszuwählen. Auch die Auswahl-App kann via Operation in der VWS über die Rest-API aufgerufen werden.

4.1.2 Verwaltungsschalen der Produkte

Die Verwaltungsschalen der Produkte sind reaktive Verwaltungsschalen bei denen über eine Rest-API Teilmodelle und Teilmodellelemente gelesen, geschrieben, gelöscht und aktualisiert werden können. Die Verwaltungsschale eines Produktes hat in diversen Teilmodellen beschrieben, welche Anforderungen das Produkt an den Produktionsprozess hat. So ist u. A. auch die Dienstleistung, die für den Einlagervorgang benötigt wird in einem Teilmodell beschrieben. Dieser wird als Interaktionselement für das Bieterverfahren benötigt, um eine Ausschreibung für einen Lagerplatz auszusenden. Weiterhin ist in einem Teilmodell das Kriterium, nach welchem diese Produziert werden soll

beschrieben. Diese Kriterien können Zeit, Preis und Aufwand sein.

4.1.3 Verwaltungsschale des Lagers

Das Lager stellt eine bestimmte Anzahl von Lagerplätzen zur Verfügung und bietet sie als Dienstleistung mit dem Bieterverfahren an. Die Dienstleistung ist in einem Teilmodell beschrieben und gleicht der Dienstbeschreibung der Dienstleistung für den Einlagervorgang in der VWS des Produkts.

Da es sich bei der Umsetzung der VWS-Serverapplikation für die VWS des Lagers um eine Implementierung mit Java unter Verwendung des Eclipse BaSyx SDK handelt sind zusätzlich für den hier beschriebenen Anwendungsfall zwei Operationen integriert. Zum einen der Entscheidungsprozess (CheckAndPricing) für die Überprüfung der eingehenden Nachrichten und der Preisberechnung und zum anderen der Entscheidungsprozess (OccupyAndInform) in dem die VWS des Lagers die Ressourcen belegt und den SR über den Vertragsabschluss informiert. Diese Entscheidungen werden im Bieterverfahren auf der SP Seite ausgeführt.

4.2 Interaktion zwischen den Komponenten

Die VWS des Lagers und des Produkts sind für sich genommen nicht in der Lage mittels Bieterverfahren miteinander Verhandlungen zu führen. Erst nach Aufruf einer der Operationen der Bidding-App und der Auswahl-App können sie Entweder als SP oder SR an Verhandlungen zwischen I4.0-Komponenten teilnehmen. Für den Aufruf über die Rest-API der VWS-Applikation werden jeweils Registry-URL, AAS-ID und die Teilmodell-ID der Dienstbeschreibung benötigt (siehe Abb. 3).

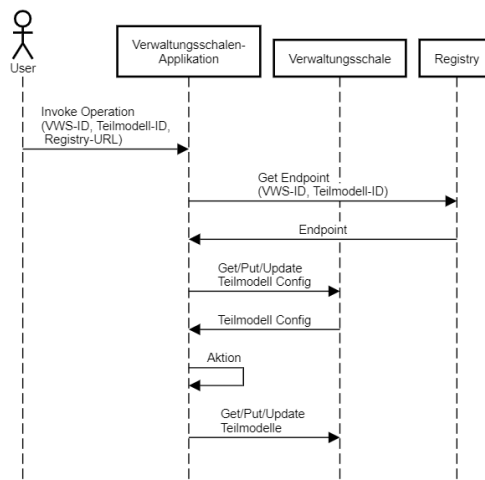


Abbildung 3: Aufruf und Interaktion einer VWS-Applikation mit einer VWS.

Mit diesen Informationen kann sich die VWS-Applikation mit der Verwaltungsschale

des Lagers bzw. des Produktes verbinden und mit ihr interagieren. VWS des Lagers bzw. des Produkts stehen nach Aufruf der Operation mit der VWS-Applikation in einer Server-Client-Beziehung.

Die Interaktion findet wie folgt statt:

1. Zunächst verbindet sich die VWS-Applikation mit der Verwaltungsschale dessen ID als Parameter übergeben wurde. Dies erreicht sie, indem sie den Endpunkt in der Registry anfordert.
2. Dann holt sie sich ein Teilmodell für die Konfiguration der Operation. Z. B. im Fall der Bidding-App das Teilmodell BiddingProcedureConfig welche in Kap. 4.3 näher erläutert wird.
3. Ausführen einer Aktion z. B. des Bieterverfahren bei der Bidding-App oder der Auswahl eines Angebots bei der Auswahl-App.
4. Während der Aktion und danach steht die VWS-Applikation und die VWS in Verbindung und tauschen Daten in Teilmodellen und Teilmodellelementen aus.

Die VWS-Applikation Bidding-App wird sowohl von der Lager als auch von dem Produkt Verwaltungsschalen verwendet. Die Bidding-App generiert die Nachrichten und tauscht diese nach dem Interaktionsprotokoll aus. Sie trifft keine Entscheidungen ob Angebote angenommen werden und berechnet nicht den Preis.

Diese Aufgaben werden von der Auswahl-App auf der SR-Seite und von der in der Verwaltungsschale des Lagers integrierten CheckingAndPricing Operation und der OccupyAndInform Operation auf der SP Seite ausgeführt.

4.3 BiddingProcedureConfig

Das Teilmodell BiddingProcedureConfig (siehe Tab. 1) wird zur Konfiguration der Bidding-App benötigt und wird für die Interaktion mit den VWS-Applikationen verwendet. Dieses Teilmodell muss für das Bieterverfahren in die VWS des Produkts und des Lagers integriert werden.

Im folgenden wird beschrieben wie die Bidding-App, sowohl auf als SR als auch als SP, nach dem Aufruf mit anderen Komponenten interagiert. Diese Beschreibung ergänzt die Interaktion welche in der Abb. 3 durch Aktion abgebildet ist.

Ablauf als SR:

1. Konfiguration des MQTT-Clients mit den Informationen in der Teilmodellelementsammlung „mqttClientConfig“.
2. Generieren und Verschicken des CallForProposal anhand der Teilmodell-ID die während des Aufrufs übergeben wurde. Das Teilmodell beschreibt die Dienstleistung.
3. Warten und Sammeln von eingehenden Angeboten. Diese werden in der Teilmodellelementsammlung „srProposals“ hinterlegt. Die Zeit, die dabei abgewartet wird, ist im Property „srTimeCollect“ festgelegt.

Tabelle 1: Teilmodell BiddingProcedureConfig. (RefEl= Referenzelement, SMEC=Teilmodellelementsammlung.)

idShort	Typ	Beschreibung
mqttClientConfig	SMEC	Konfiguration des MQTT-Client
srTimeCollect	Property	Zeit die für Sammlung
srProposals	SMEC	Sammlung aller eingegangenen Angebote
srSelection	RefEl	Referenz auf Operation der VWS-Applikation Auswahl-App
srCriterion	RefEl	Referenz auf Kriterium die bei der Auswahl berücksichtigt werden soll
srSelected	RefEl	Referenz auf das ausgewählte Angebot
spCallForProposals	SMEC	Sammlung aller anfragen auf Angebote
spProposal	RefEl	Referenz auf das Angebot das unterbreitet wird
spCheckAndPricing	RefEl	Referenz auf Operation CheckAndPricing
spOccupyAndInform	RefEl	Referenz auf Operation OccupyAndInform

4. Aufruf der Operation der Auswahl-App, auf die im Referenzelement „srSelection“ verwiesen wird.
5. Das ausgewählte Angebot auf welches „srSelected“ referenziert annehmen.
6. Nach eingegangener Bestätigung den Produktionsschritt als Abgeschlossen markieren.

Ablauf als SP:

1. Konfiguration des MQTT-Clients mit den Informationen in der Teilmodellelementsammlung „mqttClientConfig“.
2. Eingehende Proposals in der Teilmodellelementsammlung „spCallForProposals“ hinterlegen
3. Aufruf der Operation „CheckAndPricing“, auf die im Referenzelement „srSelection“ verwiesen wird.
4. Generieren eines Proposals anhand der Teilmodell-ID die während des Aufrufs übergeben wurde. Das Teilmodell beschreibt die Dienstleistung.
5. Warten auf Angebotsannahme.
6. Aufruf der Operation „spOccupyAndInform“ um den Lagerplatz als reserviert zu kennzeichnen und dem SR das angenommene Angebot zu bestätigen.

5 Zusammenfassung

Es wurde ein konkretes Beispiel aus der auftragsgesteuerten Produktion in einer industriellen Umgebung betrachtet. Das Beispiel beschrieb die automatisierte Verhandlung um einen Lagerplatz zwischen einem Produkt und einem Lager und wurde in Kap. 2 erläutert. Nachdem in Kap. 3 die Entscheidungsprozesse im Interaktionsprotokoll nach dem Bieterverfahren analysiert wurden, konnte eine mögliche Umsetzungsvariante für eine I4.0-Umgebung beschrieben werden die im Wesentlichen mit Verwaltungsschalen-Applikationen und reaktiver Verwaltungsschalen realisiert wird. Diese interagieren mittels des Teilmodells „BiddingprocedureConfig“ miteinander.

Literaturverzeichnis

- [1] Grunau, Sergej and Redeker, Magnus and Göllner, Denis and Wisniewski, Lukas: The Implementation of Proactive Asset Administration Shells: Evaluation of Possibilities and Realization in an Order Driven Production. Eingereicht in 11. Jahreskolloquium Kommunikation in der Automation (KommA 2020) in Lemgo 2020.
- [2] Plattform Industrie 4.0: Anwendungsszenario trifft Praxis: Auftragsgesteuerte Produktion eines individuellen Fahrradlenkers. 2017.
- [3] Plattform Industrie 4.0: Details of the Administration shell - Part 1: The exchange of information between partners in the value chain of Industrie 4.0. 2020.
- [4] VDI/VDE 2193 Blatt 1: Sprache für I4.0-Komponenten,“ Düsseldorf: VDI, 2019
- [5] VDI/VDE 2193 Blatt 2: Sprache für I4.0-Komponenten. Interaktionsprotokoll für Ausschreibungsverfahren. Düsseldorf: VDI, 2019
- [6] VWS vernetzt: Specification Testbed „AAS networked “: Proactive AAS - interaction according to the VDI/VDE 2193. Berlin: BMWi, 2020. URL: <https://vwsvernetzt.de/projektinhalte/spezifikationen-downloads/>
- [7] AASX-Server. URL: <https://github.com/admin-shell-io/aasx-server>, accessed: 01.07.2021
- [8] Eclipse BaSyx. URL: <https://wiki.eclipse.org/BaSyx>, accessed: 01.07.2021
- [9] Plattform Industrie 4.0: Was ist die Verwaltungsschale aus technischer Sicht?. 2021.