

# Verwaltungsschale: Offene Weiterentwicklung und Evaluation in C#.NET

Juilee Tikekar, Alexander Wollbrink, Marta Fullen, Florian Pethig

In den letzten Jahren hat sich ein lebhaftes digitales Ökosystem rund um die Verwaltungsschale (VWS; engl. Asset Administration Shell, AAS) entwickelt. Software für die Erstellung, Ausführung und Vernetzung von VWS existiert mittlerweile für eine Vielzahl unterschiedlicher Plattformen und Programmiersprachen. Hierbei fokussieren die Lösungen jeweils auf eine Menge von Anwendungsfällen und setzen unterschiedliche Schwerpunkte, bspw. leichtgewichtige Laufzeitumgebungen oder die assistierte Erstellung von VWS auf Webportalen und digitalen Marktplätzen. Insbesondere rund um die Gremienarbeit und Standardisierung der VWS in der Plattform Industrie 4.0 und der Industrial Digital Twin Association (IDTA), haben sich in den letzten Jahren die beiden C#.NET-basierten Projekte AASXPackageExplorer [1] und AASXServer [2] entwickelt.

Bei dem AASXPackageExplorer handelt es sich um einen „Viewer“ und Editor für VWS, der Nutzerinnen und Nutzer insbe-

sondere bei ihren ersten Schritten unterstützen soll. Dagegen stellt der AASXServer eine Laufzeitumgebung für VWS dar, die AASX-Dateien lädt und über das *Application Programming Interface* (API) der VWS via http/REST, OPC UA oder MQTT zugreifbar macht. In Kombination mit einer integrierten Web-Oberfläche, kommt der AASXServer in mehreren Demonstratoren [3, 4] zum Einsatz und wird grundsätzlich gerne für das Prototyping von AAS-as-a-Service-Lösungen verwendet. Bei der Implementierung beider Projekte stand stets die praktische Evaluation und Demonstration im Standard enthaltener Konzepte im Vordergrund.

## Die Verwaltungsschale in der agilen Softwareentwicklung

Um die Treiber der beiden Projekte Prof. Dr. Michael Hoffmeister (FESTO, HS Karlsruhe) und Andreas Orzelski (Phoenix Contact) ist inzwischen eine aktive Community entstanden. Um

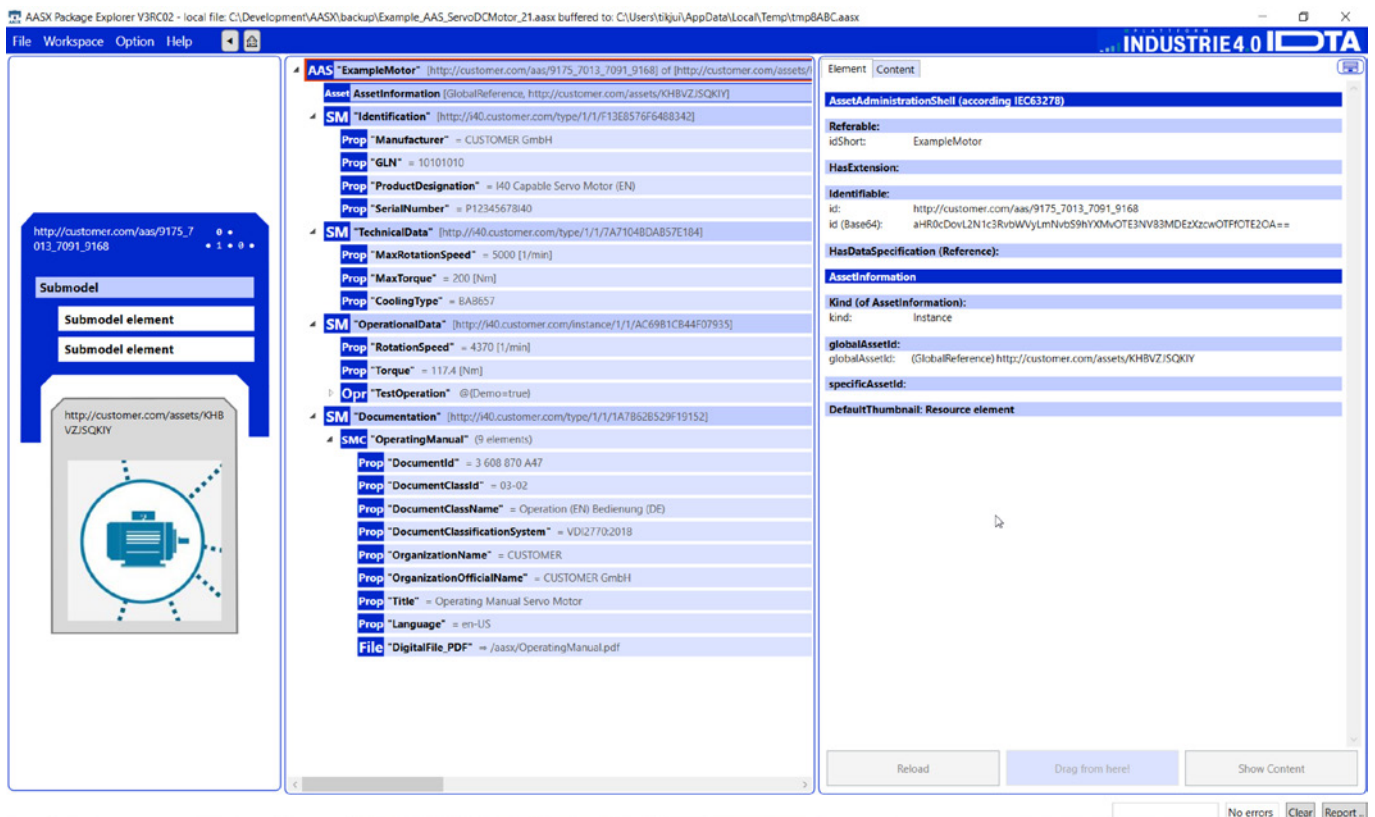


Abbildung 1: Die beispielhafte VWS eines Motors in der neuesten Version des AASXPackageExplorers.

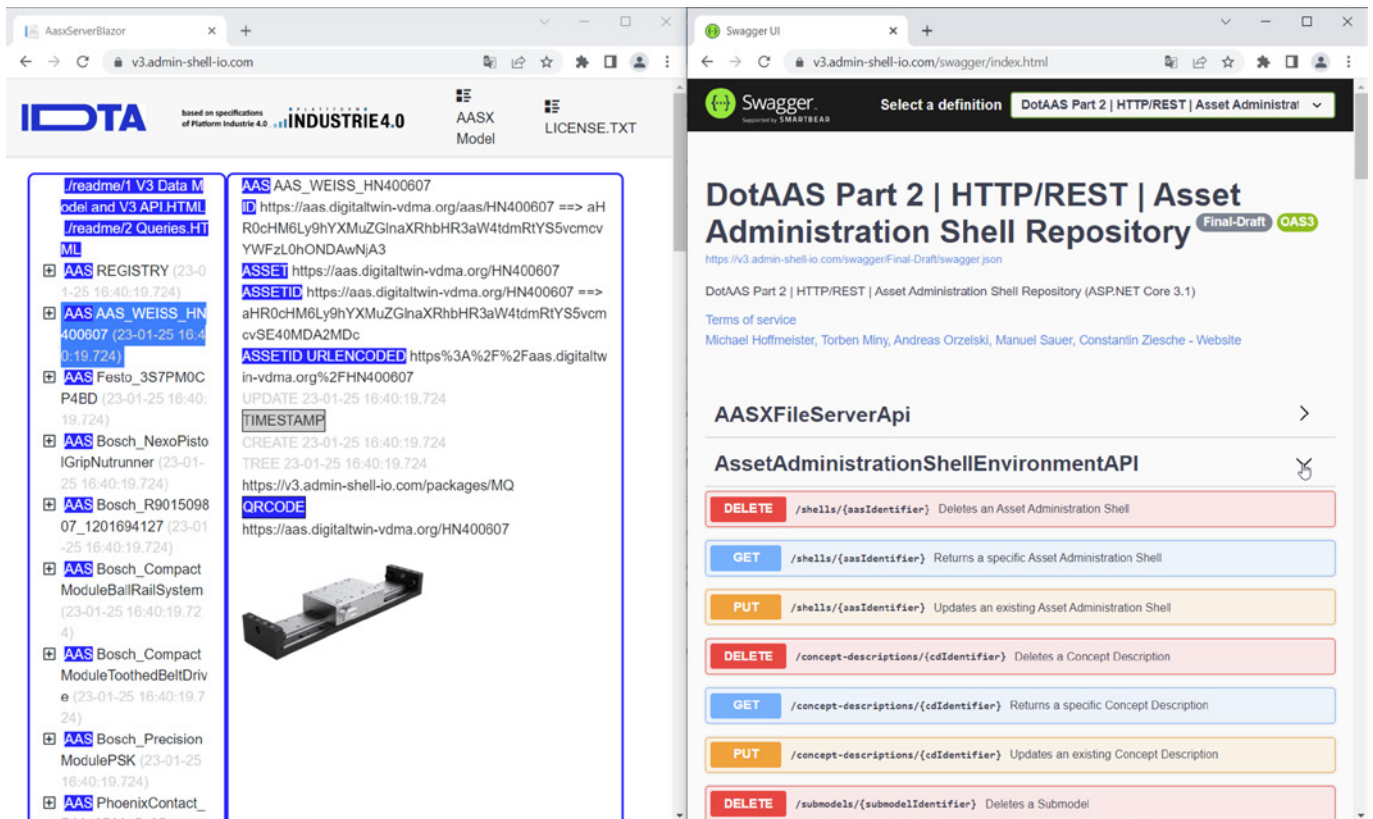


Abbildung 2: Die Weboberfläche (links) und Swagger-Definition der HTTP/REST API (rechts) eines im Internet verfügbaren AASXServers.

diese zu unterstützen und Kernpunkte wie die Aktualisierung des Datenmodells und der API des AASXPackageExplorer und des AASXServer auf den aktuellen Stand der Standardisierung vorzunehmen, wurde das Fraunhofer IOSB-INA (IOSB-INA) aus Lemgo von der IDTA im Rahmen eines gemeinsamen Entwicklungsprojekts beauftragt. Hierbei sollen die beiden Tools auch weiterhin erste Anlaufstelle für die Erprobung von Ideen und Anpassungen der Spezifikationen aus den Arbeitsgruppen der IDTA sein, die im Sinne einer agilen Softwareentwicklung zur Validierung umgesetzt werden. Darüber hinaus stehen die Qualitätssicherung und das *Open Source Management* im Fokus. Hierzu betreut das Fraunhofer IOSB-INA die automatisierten Tests, *Continuous Integration* (CI) Pipelines sowie *Issues* und *Feature Requests* aus der Open Source Community.

### Integration des Meta-Informationsmodells

Bereits im Jahr 2022 konnten das Meta-Informationsmodell („V3 RC02“ [5]) und die REST-API („V1 RC02“ [6]) des AASXServers aktualisiert werden. Als Neuerung wurde das Meta-Informationsmodell mit Hilfe von generiertem Quellcode integriert. Der Quellcode entstammt dem u. a. durch Marco Ristin (ZHAW Zürich) und Nico Braunisch (TU Dresden) vorangetriebenen Projekt *aas-core-codegen* [7], welches die Codegenerierung aus der Spezifikation der VWS ermöglicht. So soll der manuelle Aufwand für Anpassungen des Quellcodes bei neuen Versionen der Spezifikation minimiert werden. Darüber hinaus wurde die Abwärtskompatibilität zu älteren Versionen des Metamodells sichergestellt. Ältere Versionen werden automatisch in V3 RC02 konvertiert, bevor sie im AASXServer geladen werden.

Neben der reinen Aktualisierung der REST-API des AASXServers auf die Version 1 RC 02, wurde die API [8] außerdem auf ASP.NET Core (ein von Microsoft entwickeltes Web-Framework) umgestellt. So können bspw. in Abfragen via REST sogenannte *Output Modifier* verwendet werden, um die erwartete Antwort der VWS gemäß Standardisierung zu spezifizieren, z. B. ob nur der Wert eines Merkmals oder die Metadaten zurückgeliefert werden sollen. In die CI-Pipeline wurden außerdem Tests für die API integriert, die die Entwickler in dem Tool *Postman* [9] auch lokal auf ihren Systemen verwenden können. So werden die aktuellen Versionen beider Projekte stets direkt auf github getestet, gebaut und u. a. auch als Docker Images zur Verfügung gestellt.

Des Weiteren wurde eine Schnittstelle für Verbindungen zwischen AASXServer und AASXPackageExplorer implementiert, über die im Server gehostete VWS geladen und editiert werden können (*AASX File Server Interface* [10]).

### Finale Version 3.0 des Metamodell im Fokus

Seit Herbst 2022 arbeitet das IOSB-INA am AASXPackageExplorer mit und konnte auch diesen bereits auf das AAS-Metamodell V3 RC02 aktualisieren. Auch die Aktualisierung auf die finale Version des Metamodells V 3.0 wird bereits anvisiert und schnellstmöglich umgesetzt. Aufgrund der oben beschriebenen Integration des generierten Quellcodes wird erwartet, dass dies unkompliziert möglich ist. Mit Erscheinen dieser Ausgabe des atp magazins kann es bereits abgeschlossen sein. Auch die finalen Änderungen der REST-APIs werden entsprechend der kommenden Spezifikationen des V3 Release übernommen.



**AASXPackageExplorer:**  
Hier geht's zum kostenfreien  
Download ([www.github.com](http://www.github.com)).



**AASXServer:**  
Hier geht's zum kostenfreien  
Download ([www.github.com](http://www.github.com)).

Darüber hinaus sind in 2023 die Optimierung der Software-Architektur und ein dementsprechendes „Refactoring“ des Quellcodes beider Tools geplant. Hierbei sollen nicht-funktionale Anforderungen wie Nutzbarkeit, Wiederverwendbarkeit, Zuverlässigkeit, Wartbarkeit und die Erweiterung der bereits vorhandenen Security-Features in den Blick genommen werden. Ein Kernziel ist bspw., nur noch ein AAS-Modell als NuGet-Paket zur Verfügung zu stellen, das nicht nur von AASXServer und AASXPackageExplorer, sondern auch von jeder weiteren C#-Anwendung, verwendet werden kann, um VWS zu integrieren.

Das Fraunhofer IOSB-INA wird außerdem die Administration der IDTA-Serverinfrastruktur [4] unterstützen. Die Server-Infrastruktur soll interaktive Demonstrationen von VWS-Showcases ermöglichen, z. B. die Einbindung von Daten via MQTT oder erweiterte Sicherheits- und Autorisierungsmöglichkeiten. Darüber hinaus wird die Verfügbarkeit der Infrastruktur durch ein erweitertes Monitoring sichergestellt, welches u. a. Ressourcenauslastung, Netzwerkverkehr sowie Serviceerreichbarkeit umfasst. Hinzu kommen automatische „Recovery-Mechanismen“ und ein personalisiertes „Alarming“.

### CATENA-X und GAIA-X und die AAS

Des Weiteren sollen im Rahmen der Förderinitiativen CATENA-X und GAIA-X entstandene Konzepte gesichtet und ggf. integriert werden. An dieser Stelle hat das Fraunhofer IOSB-INA z. B. bereits gemeinsam mit dem ZVEI-Führungskreis Industrie 4.0 den Anwendungsfall *Collaborative Condition Monitoring* (CCM) mit *GAIA-X Federation Services* in der SmartFactoryOWL pilotiert [11]. Hierbei konnten bereits wertvolle Erkenntnisse gesammelt werden, z. B. zu *Self Sovereign Identities* (SSI) für die Authentifizierung und dem *Federated Catalogue* für das öffentlich auffindbar machen von VWS.

Getrieben durch die digitale und ökologische Transformation der Industrie erfolgt in den kommenden Jahren der Transfer der Verwaltungsschale und der umgebenden Tools in die Unternehmenspraxis. Konkrete Beispiele sind das digitale Typenschild und der Showcase *Product Carbon Footprint* (PCF) des ZVEI. Mit Hilfe der VWS kann hier eine Menge Papier eingespart und die Transparenz über Lieferketten hinweg erhöht werden. Voraussetzung sind erprobte und einfach einsetzbare Software-Tools. Hieran arbeiten wir gemeinsam mit der IDTA und unterstützen auch Sie und Ihr Unternehmen gerne bei der praktischen Anwendung.

### Referenzen

- [1] Github, Inc. (2022). Admin-shell-io / aasx-package-explorer. Abgerufen von: <https://github.com/admin-shell-io/aasx-package-explorer>
- [2] Github, Inc. (2022). Admin-shell-io / aasx-server. Abgerufen von: <https://github.com/admin-shell-io/aasx-server>
- [3] VDMA e.V. (2022). Industrie 4.0 – Herausforderungen und Chancen. Abgerufen von: <https://fluidtechnik40.de/>
- [4] IDTA e.V. (2022). Home of Asset Administration Shell (AAS). Abgerufen von: <https://admin-shell-io.com/>
- [5] Bundesministerium für Wirtschaft und Klimaschutz (BMWK). (2022). Details of the Asset Administration Shell - Part 1: The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC02). Abgerufen von: [https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details\\_of\\_the\\_Asset\\_Administration\\_Shell\\_Part1\\_V3.html](https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html)
- [6] Bundesministerium für Wirtschaft und Klimaschutz (BMWK). (2021). Details of the Asset Administration Shell - Part 2: Interoperability at Runtime – Exchanging Information via Application Programming Interfaces (Version 1.0RC02). Abgerufen von: [https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details\\_of\\_the\\_Asset\\_Administration\\_Shell\\_Part\\_2\\_V1.html](https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part_2_V1.html)
- [7] Github, Inc. (2022). Aas-core-works / aas-core-codegen. Abgerufen von: <https://github.com/aas-core-works/aas-core-codegen>
- [8] Swaggerhub, Inc. (2022). AssetAdministrationShell-Environment. Abgerufen von: [https://app.swaggerhub.com/apis/Plattform\\_i40/AssetAdministrationShell-Environment/V1.0RC03](https://app.swaggerhub.com/apis/Plattform_i40/AssetAdministrationShell-Environment/V1.0RC03)
- [9] Postman, Inc. (2023). Download Postman. Abgerufen von: <https://www.postman.com/downloads/>
- [10] Swaggerhub, Inc. (2022). AASX-File-Server. Abgerufen von: [https://app.swaggerhub.com/apis/Plattform\\_i40/AASX-File-Server/V1.0RC03](https://app.swaggerhub.com/apis/Plattform_i40/AASX-File-Server/V1.0RC03)
- [11] IDTA e.V. (2023). Demonstrator Collaborative Condition Monitoring – Daten teilen als Win-Win. Abgerufen von: <https://industrialdigitaltwin.org/use-cases/demonstrator-collaborative-condition-monitoring-daten-teilen-als-win-win>

### Juilee Tikekar

Fraunhofer IOSB-INA  
32657 Lemgo

### Alexander Wollbrink

Fraunhofer IOSB-INA  
32657 Lemgo

### Marta Fullen

Fraunhofer IOSB-INA  
32657 Lemgo

### Florian Pethig

Fraunhofer IOSB-INA  
32657 Lemgo