AAS Engineering

Asset Administration Shell-based Integrated Automation Engineering

Programm / Ausschreibung	BASIS, Basisprogramm, Budgetjahr 2021	**Status**	abgeschlossen
Projektstart	01.09.2021	**Projektende**	30.11.2022
Zeitraum	2021 - 2022	**Projektlaufzeit**	15 Monate
Keywords			

Projektbeschreibung

Problem und Ziel
Bei der Planung und der Entwicklung industrieller Anlagen sind Experten aus mehreren Fachbereichen wie Maschinenbau, Pneumatik & Hydraulik, Elektrotechnik, Steuerungstechnik, Visualisierung usw. beteiligt. Diese arbeiten auf der Entwicklungs- und Konstruktionsebene und benötigen zur Umsetzung ihrer Aufgaben eine Vielzahl unterschiedlicher, spezialisierter Software-Werkzeuge. In der Realisations- und Problemlösungsebene soll schlussendlich die Zusammenführung aller Planungsteile zur Erstellung der fertigen Anlage erfolgen. Damit der Gesamtprozess des parallelen Engineerings erfolgreich ist, ist allerdings eine effektive und effiziente Zusammenarbeit durch direkte bzw. indirekte Kommunikation zwischen den Experten, sowohl auf Entwicklungs- und Konstruktionsebene als auch auf Realisations- und Problemlösungsebene, nötig.
Aber nicht nur die Frage "wie plane ich eine neue Anlage", sondern auch die Frage "Wie kann ich eine Anlage möglichst flexibel und ressourcenschonend einsetzen und an neue Anforderungen anpassen" wurde und wird immer wichtiger und Antworten darauf wurden gesucht. Ziel ist es, möglichst flexibel, mit bis zur Losgröße 1, produzieren zu können. Es soll möglich sein, dass jedes einzelne erstellte Objekt unterschiedliche Ausprägungen hat. Unterstützt soll dieser Ansatz durch das "Industrielle Internet der Dinge" (IIoT, Industrial Internet of Things) werden. Alle Geräte, ob Sensoren, Regler, Motoren, o. ähnliches sollen selbstständig kommunizierende Objekte sein und beim Engineering sollen Metainformationen zu allen beteiligten Artefakten (Geräte zur Produktion, Produktionsgüter, Materialien, Bestellungen, ...) vorhanden sein. In der Produktion entstehen damit sogenannte Cyber-Physical Production Systems (CPPS) mit intelligenten Maschinen, Lagersystemen und Betriebsmitteln, die eigenständig Informationen austauschen, Aktionen auslösen und sich gegenseitig selbstständig steuern. Ein wichtiger zusätzlicher, sich daraus ergebender Vorteil ist die erhöhte Flexibilität im Wartungsfall, wenn Geräte durch andere ("nur ähnliche") ersetzt werden müssen.

In den letzten Jahren wurde das Konzept von Assets und der Asset Administration Shell (AAS) von den Industrie 4.0 Gremien entwickelt. Jedes Objekt, das im Prozess vernetzt verwendet wird, entspricht einem Asset. So ein Asset kann ein physisches Produkt (→ Material-AAS) oder ein Gerät (→ Device-AAS) sein, ein Softwareelement oder Dokumente wie Pläne, Verträge und Bestellungen. Jedes dieser Asset Arten wird auf Softwareseite durch einen Asset Type repräsentiert (z.B. ein konkreter Motortyp eines konkreten Herstellers). Ein Exemplar eines dieser Assets wird auf Softwarekonfigurationsseite durch eine

Asset Instance dargestellt. Zu einem späteren Zeitpunkt (bevor die Applikation erzeugt wird) muss eine Beziehung zwischen dem konkreten (physikalischen) Asset in der Real World und der Asset Instance im Programm hergestellt werden (z.B. ein konkreter Motor innerhalb der Anlage wird einem Softwareelement zugeordnet). Asset Typen und Instanzen erlauben eine generische, flexiblere Entwicklung der Software, die die konkreten Parametrierungen dynamisch zur Laufzeit bezieht. Um diese Informationen von einem Asset beziehen zu können, bekommen Assets im Programm ein digitales Typenschild mit entsprechender Information. Die AAS (Asset Administration Shell) stellt dieses digitale Typenschild zu einem konkreten Asset dar.

Eine AAS besteht aus einer Reihe von Untermodellen, in denen alle Informationen und Funktionen eines bestimmten Assets - einschließlich seiner Merkmale, Eigenschaften, Status, Parameter, Messdaten und Fähigkeiten - beschrieben werden können. Es ermöglicht die Nutzung unterschiedlicher Kommunikationskanäle und -anwendungen und dient als Bindeglied zwischen Objekten und der vernetzten, digitalen und verteilten Welt. Die Struktur der AAS wird über ein technologieunabhängiges Metamodell und mehrere technologie-spezifische Serialisierungszuordnungen wie AutomationML, XML, JSON oder OPC UA definiert. Sein Inhalt wird über domänenspezifische Submodellspezifikationen definiert. Die Interaktion mit der AAS kann unterschiedlichen Mustern folgen, die unterschiedliche technische Anforderungen haben, d. H. Dateiaustausch, Server-Client- und Peer-to-Peer-Interaktion.

Ziel der Forschungs- und Entwicklungsarbeiten in diesem Projekt in den nächsten 2 Jahren (Plan 1.9.2021 - 31.8.2023) ist es, einen Prototypen für das Konzept der AAS im Bereich des Engineerings und der Laufzeit von Anlagen umzusetzen. Es soll ein Prototype dafür basierend auf das Produkt logi.CAD 3 und logi.CLOUD und unseren logi.RTS Laufzeitsystemen realisiert werden. Damit soll es erstmalig möglich sein, AAS im Engineering Prozess selbst und auch im Ergebnis des Engineerings, im Steuerprogramm einer Anlage, verwenden zu können und damit das Ziel der dynamisch rekonfigurierbaren Anlagen umzusetzen. Um die unterschiedlichen Aspekte und Vorteile, die sich für Anwender ergeben, demonstrieren zu können, soll das Projekt gemeinsam mit Forschungs- und Entwicklungspartnern aus verschiedenen Bereichen des Engineerings umgesetzt werden, die eine Anbindung unterschiedlicher Engineering Tools zur Verbesserung und Vereinfachung des Anlagenengineering an logi.CAD 3 mit Hilfe von AAS demonstrieren sollen. logi.CAD 3 selbst soll diese Anbindung mittels AAS und neuer DSLs (Domain Specific Languages) ermöglichen.

Gemeinsam mit Forschungs- und Entwicklungspartnern werden zum Erreichen der obigen Punkte folgende Punkte erforscht und prototypische Lösungen entwickelt:

AAS im Engineeringprozess
Devices (z.B. Sensoren) sollen vom Hersteller gemeinsam mit entsprechenden Beschreibungen (→ Type-AAS mit statischen Parameterinformationen wie mechanische Größe, min/max Werte, Ort, Identifikation, ... und Instance-AAS mit dynamischen Laufzeitwerten wie aktueller Wert ... nach Verdrahtung und Zuordnung) geliefert werden und die AAS müssen vom Administrator in ein AAS Repository eingespielt werden.
Der Entwickler kann nun in seinem Programm Bausteine mit diesen Device-AAS anlegen und zu komplexeren Bausteinen zusammenfügen. Er hat damit Steuerbausteine, die nur über die AAS von der Hardware abhängen (von den Typen, aber nicht von konkreten Instanzen wie bisher). Aus diesen Bausteinen können langfristig CPµServices erstellt werden die unabhängig agieren können. Der Anwendungsentwickler kann nun aus diesen CPµServices Applikationen dynamisch zusammenstellen (konfigurieren und orchestrieren). Damit wird Plug&Play Automatisierung (Software durch Orchestrieren)

erstellt.
Zusätzlich sollen diese AAS auch im erzeugten Programm, das zur Steuerung der Anlage eingesetzt wird, verwendet werden und damit die bisher starre Verdrahtung und Verbindung zwischen Anlagenteilen und Steuerprogrammen aufzubrechen. Die Änderungen in diesem Umfeld sind vergleichbar mit den früher starren Konfigurationen von PC (Personal Computer), bei denen bei jeder Änderung (Tausch einer Festplatte, hinzufügen von Speicher, o.ä.) die Systemkonfiguration geändert werden musste - ein komplexer und fehleranfälliger Prozess. Heute können durch "Plug & Play" Mechanismen die verschiedenen Teile eines PCs sich selbst identifizieren, und können im Betrieb durch einen anderen Teil ersetzt werden (USB Devices enthalten am Device eine Konfigurations-Datei die automatisch verwendet wird). Durch das Konzept der AAS soll es möglich sein, dass sich auch Teile einer Anlage selbst identifizieren und durch geeignete Maßnahmen das Steuerprogramm zur Laufzeit anpassen können. Damit können Teile einer Anlage (z.B. eine Pumpe, Motor, oder eine Temperaturmessgerät) durch ein anderes, nicht 100% identes, Gerät ersetzt werden. Dies ist langfristig für die Wartung, und kurzfristig für dynamische Anlagen (für "Losgröße 1") von zentraler Bedeutung → Plug&Play Automatisierung (Steuerhardware automatisch erkennen).
Wir wollen diese Plug&Play Automatisierung auf 2 Ebenen einführen:
- Bei einer Anlagen-Orchestrierung - Level 3 - durch Zusammenstellen der Software
- Zur Laufzeit, wenn Komponenten durch andere ersetzt, weitere hinzugefügt werden oder Komponenten wegfallen.

Das Ziel ist, dass Änderungen beim Material oder bei verwendeten Geräten einfach, schnell und mittels standardisierter Methoden in die Software übertragen werden können.

Flexible AAS Infrastruktur & Modellierung von Engineering AAS Typen und OPC-UA basierter AAS Zugriff
Wir wollen mit IncQuery einen integrierten OPC-UA Model Server für AAS entwickeln und diesen auch an unser RTS Laufzeitsystem anbinden. Der integrierte Model Server (IMS) soll als single integrated view des model spaces dienen. Im Moment liefern unterschiedliche Server (auf verschiedenen Devices, Servern, verschiedene Technologien ...) unterschiedliche Daten (unterschiedliche Model Teile) von unterschiedlichen Teilen der Anlage. Die Daten können im Moment von OPC-UA Servern, von AML Hubs, oder zum Teil sogar von Laufzeit/Anbieter spezifischen Backends geliefert werden.

DSLs für Konfiguration einer Applikation und Parametrierung von aggregierten AAS & Anlagen Orchestrierung
Gemeinsam mit EclipseSource sollen DSLs für Parametrierung, Konfiguration und Orchestrierung einer Applikation und von aggregierten AAS entwickelt werden. Es ist hier jeweils die Sprache selbst, als auch eine Implementierung und Integration der Sprache in logi.CAD 3 zu entwickeln.

Technisch bedeutet dies in den nächsten 2 Jahren:
- Entwicklung von Device AAS Formaten und eines Device AAS Type Repositories mit OPC-UA Server Umsetzung.
- Kommunikationstechnologie unabhängige Einbindung zu AAS in Engineering Anwendungen: Mit dem in logi.XRA entwickelten Konzept von Objekt Libraries die mittels Library Providern in ein Projekt eingebunden werden können, soll eine Technologieunabhängige Schnittstelle zu AAS Servern geschaffen werden.
-- Es soll ein Library Provider entwickelt werden, der von einem OPC-UA Server eine Library von AAS für ein konkretes Projekt laden und in ein LC3 Projekt installieren kann.
-- Dabei ist lediglich die Existenz eines Configuration Submodels erforderlich, sodass auch mit unvollständigen AAS-Definitionen gearbeitet werden kann.

-- Die in dieser Library vorhandenen Objekte sind AAS Typen und sollen nur im Kontext von AAS-Variablen verwendet werden können.
-- Dazu Erweiterung der IEC 61131-3 ST Sprache um spezielle Konstrukte zur Unterstützung der Asset-Typen.
- Einsatz von AAS im Engineering Bereich inklusive der Entwicklung der Grundlagen einer OPC-UA Companion Spezifikation für AAS im Engineering Bereich [OPC-UA 1]
-- Das IEC-61131-3 Variablen Konzept soll dahingehend erweitert werden, dass eine neue Variablenart (AAS-Variablen) eingeführt wird umso AAS-Typen zu instanziieren.
-- Mit der Companion Spezifikation wird das MetaModel für AAS für Engineering einschränkbar gemacht.
- Aggregation von Device AAS: Für Devices (Level 0 - Sensoren und Aktoren, langfristig existent) werden auf Level 1 (Hardware Abstraktion Layer) Bausteine erstellt. Diese verwenden Device-AAS, um Hardware und Parameter unabhängig eine Beziehung zwischen Baustein und physischen Device herzustellen. Einfache Berechnungen (Validierung der Daten, Normierung der Daten) werden bereits auf diesem Level durchgeführt. Die Bausteine auf diesem Level werden im Level 2 zu Smart Komponenten (Level 2) zusammengefasst die keine unmittelbare Hardware Abhängigkeit haben (aber nachträglich entsprechend parametriert werden können) aber über das Control Interface der Device AAS Steueraufgaben realisieren können. Diese Smart Components können über Material-AAS ("temporär", kurzfristig existent) auf das aktuelle Werkstück zugreifen und damit den Steuercode entsprechend anpassen.
- DSL zur Orchestrierung und Konfiguration von Anlagenteilen (beispielsweise in der Form von VDI/VDE/ NAMUR 2658 Module Type Packages MTP) zu einer vollständigen Anlage am Plant Orchestration Level (POL) (Programme, später auch CPµServices) → Darauf können weitere Tools aufsetzen und z.B. einen graphischen Konfigurator anbieten.
- DSL für Device-AAS Instanz Parametrierung (Zuordnung von AAS Instanzen zu Devices) - Input/Output Konfiguration.
-- Damit kann entweder direkt mit der DSL die Parametrierung durchgeführt werden oder mit externen Tools eine komfortable Parametrierung durchgeführt werden
- AAS im Laufzeitsystem zur dynamischen Anlagen Konfiguration und Parametrierung (Anpassungen zur Laufzeit)
-- Für AAS-Instanzen werden dann im Weiteren automatisch IEC-61131-3 Variablen angelegt und diese dynamisch von einem AAS-System Service mit IO-Daten und Parametern versorgt (dynamische readonly Daten) oder werden als Source für eine Übertragung zum System Service verwendet (readwrite Daten)
- Integrated Model Server basierend auf einem Runtime OPC-UA AAS Server (AAS mit Anlagen Konfiguration, Parametrierung und Echtzeit Laufzeit Daten)
-- Damit können Echtzeit Anlagenparameter von einem Model Server erfragt werden und weitere Laufzeit Tools zur Datenauswertung (→ Big Data Analysis) an die Anlage eingebunden werden → IncQuery

Ziel ist es am Ende des 1. Forschungsjahres die Entwicklungen im Engineering Tool selbst soweit abgeschlossen zu haben, dass prototypisch AAS im Kontext eines IEC 61131-3 Programmes verwendet und vor dem Build der Applikation parametriert werden können.

Am Ende des 2. Forschungsjahres soll es die Erweiterungen des RTS geben, um AAS dynamisch verwenden und über einen OPC-UA Server anderen zur Verfügung stellen zu können. Einfaches Filtering der AAS im OPC-UA Server soll entsprechend der (Basis) Companion Spezifikation erfolgen. Dynamische Konfigs über einen OPC-UA Client am Zielsystem. Im Engineering System soll es da die Orchestrierungsmöglichkeiten für die (emulierten) CPµServices geben.

Projektpartner

- Neuron Engineering Tools GmbH