



OPC Unified Architecture

Die nächste Stufe der Interoperabilität

OPC Unified Architecture

Next Level of Interoperability

Udo Enste, LeiKon GmbH, Herzogenrath,
Wolfgang Mahnke, ABB Forschungszentrum, Industrial Software Systems, Ladenburg

Zusammenfassung OPC Unified Architecture bietet erweiterte Möglichkeiten der Interoperabilität. Auf Basis eines objekt-orientierten Konzeptes stehen moderne Mittel zur Informationsmodellierung zur Verfügung, um den über Schnittstellen zur Verfügung zu stellenden Datenhaushalt eines Systems repräsentieren zu können. Mit Hilfe einer Kommunikationsinfrastruktur werden darüber hinaus eine Systemarchitektur und generische Dienstevorgaben gemacht. ▶▶▶ **Sum-**

mary OPC Unified Architecture offers extended functionalities to realize interoperability between different systems. Based on object-oriented concepts, modern means for information modelling are available to represent the data base of a system, which should be shareable via an OPC UA interface. In addition to the information modelling aspects, OPC UA specifies a communication infrastructure by defining generic services and a system architecture.

Schlagwörter OPC Unified Architecture, OPC UA, IEC 62541 ▶▶▶ **Keywords** OPC Unified Architecture, OPC UA, IEC 62541

1 Einleitung

OPC UA wurde von der OPC Foundation entwickelt. Erste Teile der OPC UA Spezifikation wurden bereits 2006 veröffentlicht und 2009 erfolgte die erste Veröffentlichung der OPC Foundation, die auch die für eine interoperable Implementierung notwendige Technologieabbildung enthielt [1]. Inzwischen ist OPC UA in Teilen als internationale Norm IEC 62541 standardisiert.

Klassisches OPC – der Vorgänger von OPC UA – ist ein akzeptierter und stark verbreiteter Standard im Umfeld der industriellen Automatisierung. Mit den Spezifikationen OPC DA (Data Access – Zugriff auf aktuelle Daten), OPC A&E (Alarms and Events – Alarmer und Ereignisse), sowie OPC HDA (Historical Data Access – Zugriff auf historische Daten) bietet klassisches OPC einen einheitlichen Datenaustausch zwischen produktionsnahen IT-Systemen. Technologisch ist klassisches OPC eine Schnittstellendefinition basierend auf der Microsofttechnologie COM/DCOM und ist damit auf Windows-Umgebungen beschränkt.

OPC UA vereinigt die Funktionalität der bisher getrennten Spezifikationen des klassischen OPC, ermöglicht eine plattformunabhängige Nutzung und bietet erweiterte Formen der Interoperabilität. Als Basistechnologie wird eine Service-orientierte Architektur verwendet.

Dem Aspekt der Informationssicherheit kommt eine immer stärkere Bedeutung zu, da Automatisierungslösungen zunehmend nicht mehr in separierten und isolierten Netzen laufen, sondern mit Büronetzen oder dem Internet verbunden sind. Angreifer fokussieren sich inzwischen vermehrt auf Automatisierungssysteme [2]. Informationssicherheit ist ein integraler Bestandteil von OPC UA. Neben der Authentifizierung von Benutzern ermöglicht es die Authentifizierung von Anwendungen (Software) und Installationen dieser Anwendungen. Nachrichten können signiert und verschlüsselt werden, um ein Ändern und Lesen der Nachrichten durch Dritte zu verhindern.

OPC UA stellt Mechanismen zur Realisierung einer verlässlichen und robusten Kommunikationsinfrastruktur

tur bereit. Darunter zählt z.B. die Handhabung verlorener Nachrichten, Unterstützung der Umschaltung in Redundanzstrukturen (Failover) und die Erkennung von Kommunikationsunterbrechungen (Heartbeat).

Die Nutzung des Funktionsumfangs von OPC UA ist skalierbar. OPC UA kann sowohl auf kleinen eingebetteten Geräten mit stark limitierten Ressourcen als auch auf sehr leistungsstarken Umgebungen wie Mainframes angewendet werden. Aber auch die Nutzung der Informationsmodellierungsmöglichkeiten kann skaliert werden. Je nach Anforderungen kann ein OPC UA Server ein sehr komplexes Modell oder auch ein sehr einfaches Modell anbieten. Ein OPC UA Client kann ein solches Informationsmodell für Datenerkundungen nutzen oder lediglich auf vorkonfigurierte Daten zugreifen.

Die Informationsmodellierungsmöglichkeiten von OPC UA bieten das Potential, dass OPC UA die Kommunikationsinfrastruktur für viele unterschiedliche Informationsmodelle unterschiedlicher Anwendungsdomänen werden kann. Inzwischen gibt es bereits eine Reihe standardisierter Informationsmodelle basierend auf OPC UA, die das generische und mächtige Metamodell von OPC UA verwenden.

2 Überblick

OPC UA definiert die Kommunikationsinfrastruktur und das Metamodell zur Beschreibung von OPC UA basierten Informationsmodellen. Das Metamodell enthält Konstrukte wie Objekte und Variablen. Die Kommunikationsinfrastruktur beschreibt Implementationsvorgaben basierend auf Technologien wie TCP oder Web Services. Damit bilden das Metamodell und die Kommunikationsinfrastruktur die Säulen der OPC UA Spezifikation (siehe Bild 1).

Die OPC UA Services definieren Operationen auf den Konstrukten des Metamodells wie Lesen, Schreiben, und Navigieren, die über die Kommunikationsinfrastruktur übertragen werden. Das Basis-Informationsmodell spezifiziert Basis-Typen, Einstiegspunkte in den Adressraum

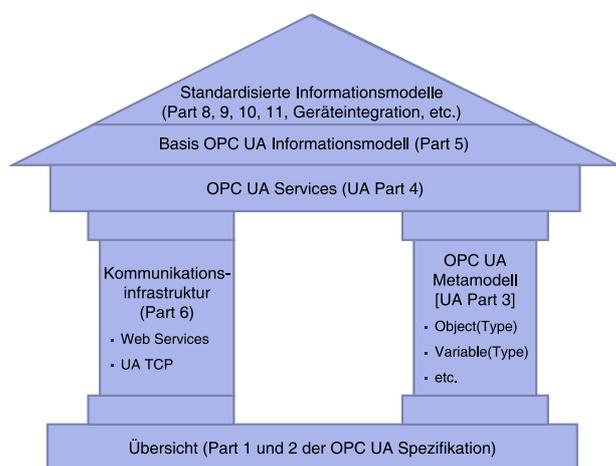


Bild 1 Überblick OPC UA Spezifikation.

sowie Diagnose-Informationen und wird von Informationsmodellen für Datenzugriff (Data Access), Programme (Programs), Alarmer und Bedingungen (Alarms and Conditions) sowie Historischen Zugriff von Daten und Events (Historical Access) verfeinert. Zusätzlich können herstellerspezifische oder standardisierte Informationsmodelle basierend auf den genannten Informationsmodellen definiert werden (s. Abschnitt 4).

Die Kommunikationsarchitektur von OPC UA bietet Technologieabbildungen auf HTTP/SOAP-basierte Web Services sowie ein eigenes Protokoll basierend auf TCP/IP. Im ersten Fall können die Daten als XML Dokumente kodiert werden, um Interoperabilität zu Web Service-basierten Anwendungen sicherzustellen. Das Binärformat kann verwendet werden, um einen schnellen Datenaustausch sicherstellen zu können. Mit dem Konzept abstrakter Service-Definitionen ist es einfach möglich, eine neue Technologieabbildung zu definieren, falls sich in der Zukunft eine neue Kommunikationstechnologie durchsetzen sollte.

3 Informationsmodellierung

OPC UA bietet mächtige Möglichkeiten der Informationsmodellierung. Die verwendeten Prinzipien sind dabei (aus [1]):

- *Verwendung Objekt-orientierter Techniken inklusive Typhierarchien und Vererbung.* Typisierte Instanzen erlauben Clients alle Instanzen desselben Typs auf dieselbe Weise handzuhaben. Typhierarchien ermöglichen dabei lediglich Basistypen zu berücksichtigen und spezialisierte Informationen zu ignorieren.
- *Informationen über Datenobjekttypen werden bereitgestellt.* Auf diese Informationen kann auf die gleiche Weise wie auf Informationen über Instanzen (z. B. Variablenwerte) zugegriffen werden. Damit können Clients Typinformationen auch zur Laufzeit aus dem Server auslesen.
- *Ein vermaschtes Netz von Knoten erlaubt es, Informationen auf verschiedene Weise zu verbinden.* OPC UA unterstützt damit unterschiedliche Informationsansichten, in denen verschiedene Semantiken durch die Verwendung unterschiedlicher Referenztypen differenziert dargestellt werden können. Die gleiche Information wird so in verschiedenen Kontexten, abhängig vom Verwendungszweck, zur Verfügung gestellt.
- *Erweiterbarkeit der Typhierarchien inklusive der Hierarchie von Referenztypen.* Informationsmodelle können verschiedene Erweiterungsmöglichkeiten verwenden. Neben der Definition von Subtypen für Objekte und Variablen können eigene Typen von Referenzen oder Methoden definiert werden.
- *Keine Limitierung wie Informationen modelliert werden sollen, um passende Modelle für die angebotenen Daten bereitstellen zu können.* Damit kann ein OPC UA Server, der auf ein System mit einem eigenen reichhaltigen Informationsmodell zugreift, möglichst natürlich die-

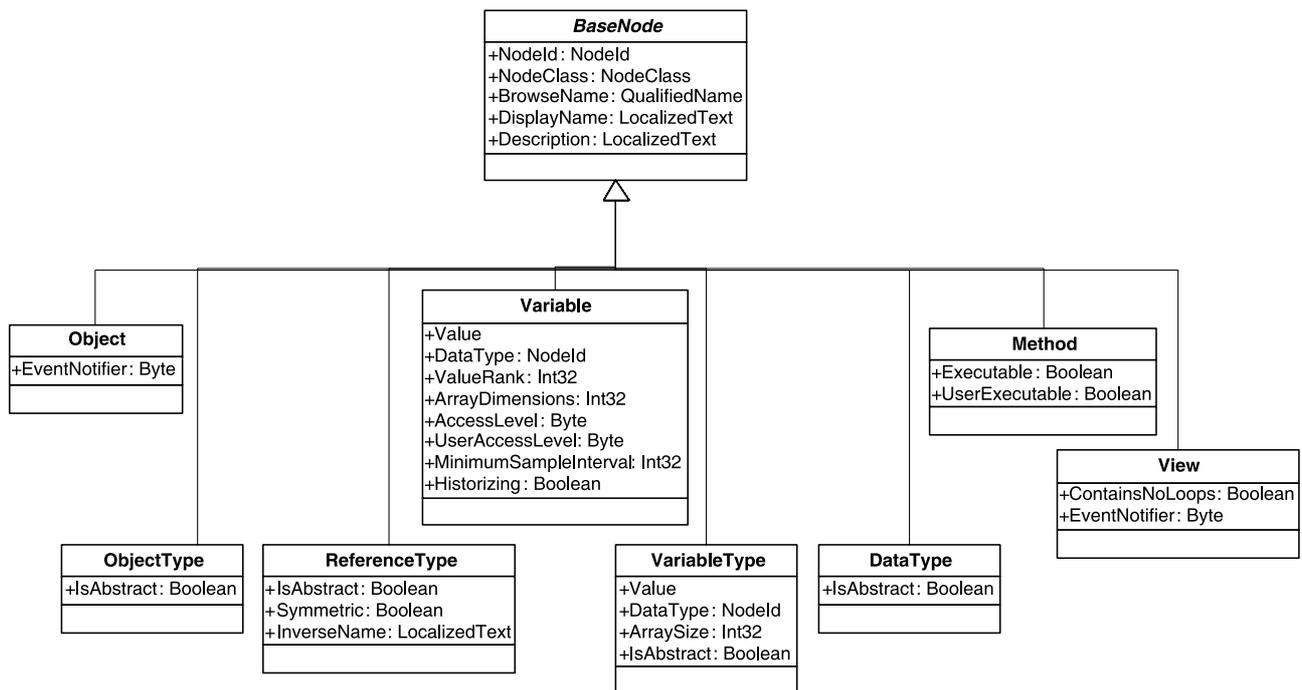


Bild 2 Konzepte des OPC UA Metamodells.

ses Modell anbieten, anstatt es auf ein anderes Modell abbilden zu müssen.

- *Informationsmodellierung findet auf der Serverseite statt.* Ein Informationsmodell existiert auf dem OPC UA Server und wird von Clients genutzt. Ein Client muss intern kein OPC UA Informationsmodell verwalten und auch dem Server kein solches Modell anbieten.

Auf Basis dieser Prinzipien können sowohl sehr einfache als auch sehr komplexe und mächtige Informationsmodelle von OPC UA Servern bereitgestellt und von OPC Clients genutzt werden. Die Konzepte des Metamodells sind in Bild 2 zusammengefasst. Ein Knoten kann über Referenzen mit anderen Knoten verbunden werden. Jeder Knoten hat Attribute wie Namen und Identifikation. Es gibt verschiedene Knotenklassen für verschiedene Zwecke, wie die Repräsentation einer Methode, eines Objekts, um den Adressraum zu strukturieren oder einer Variable, die Daten wie z. B. einen Messwert enthält. Jede Knotenklasse hat spezielle Attribute für ihre Zwecke. Eine Variable enthält beispielsweise das Value Attribut, das den Wert repräsentiert.

OPC UA unterstützt einfache Datentypen wie Integer und Boolean, aber auch Aufzählungen und strukturierte Datentypen, die aus anderen Datentypen zusammengesetzt sind. Jeder Wert kann ein (multidimensionales) Feld sein, wobei es möglich ist, lediglich auf Teile des Feldes zuzugreifen. Datentypen sind erweiterbar. Clients können die Information über neu definierte Datentypen aus dem Server auslesen, um so zur Laufzeit die Daten interpretieren zu können.

Mit Ereignissen kann ein OPC UA Server einem Client Informationen bereitstellen, die nicht als Objekte

und Variablen im Adressraum modelliert wurden wie beispielsweise der Ausfall eines Gerätes. Ereignisse enthalten mehrere Felder wie Identifikation, Zeitstempel und Nachrichtentext. Es gibt eine erweiterbare Ereignistyphierarchie, basierend auf Objekttypen in denen die Felder definiert sind. Durch die Erweiterung der Hierarchie können sowohl Ereigniskategorien als auch neue Ereignisfelder definiert werden. Clients können Filter definieren, um beispielsweise lediglich bestimmte Ereigniskategorien oder bestimmte Ereignisfelder zu empfangen.

4 Standardisierte Informationsmodelle

OPC UA erlaubt die Definition standardisierter Informationsmodelle basierend auf dem Metamodell von OPC UA. Zum einen erfolgt der Datenaustausch damit standardisiert. Jeder OPC UA Client kann auf Daten eines OPC UA Servers zugreifen, auch ohne vorheriges Wissen über das spezifische Informationsmodell, das der OPC UA Server zur Abbildung der Informationen verwendet. Zum anderen wird die Semantik der Datenobjekte über das Informationsmodell standardisiert, so dass sich OPC Clients darauf verlassen können, dass OPC UA Server, die das gleiche, standardisierte Informationsmodell verwenden, Informationen auf die gleiche Weise anbieten und Clients somit die Semantik interpretieren können. Insbesondere dieser zweite Aspekt bietet einen wesentlichen Mehrwert zur Umsetzung herstellerunabhängiger Interoperabilität (s. dazu auch Abschnitt 6 und 7).

Standardisierte Informationsmodelle definieren im Kern eine Reihe an Typen (Variablen-, Objekt-, Daten-, oder Referenztypen). Es können Objekte und Varia-

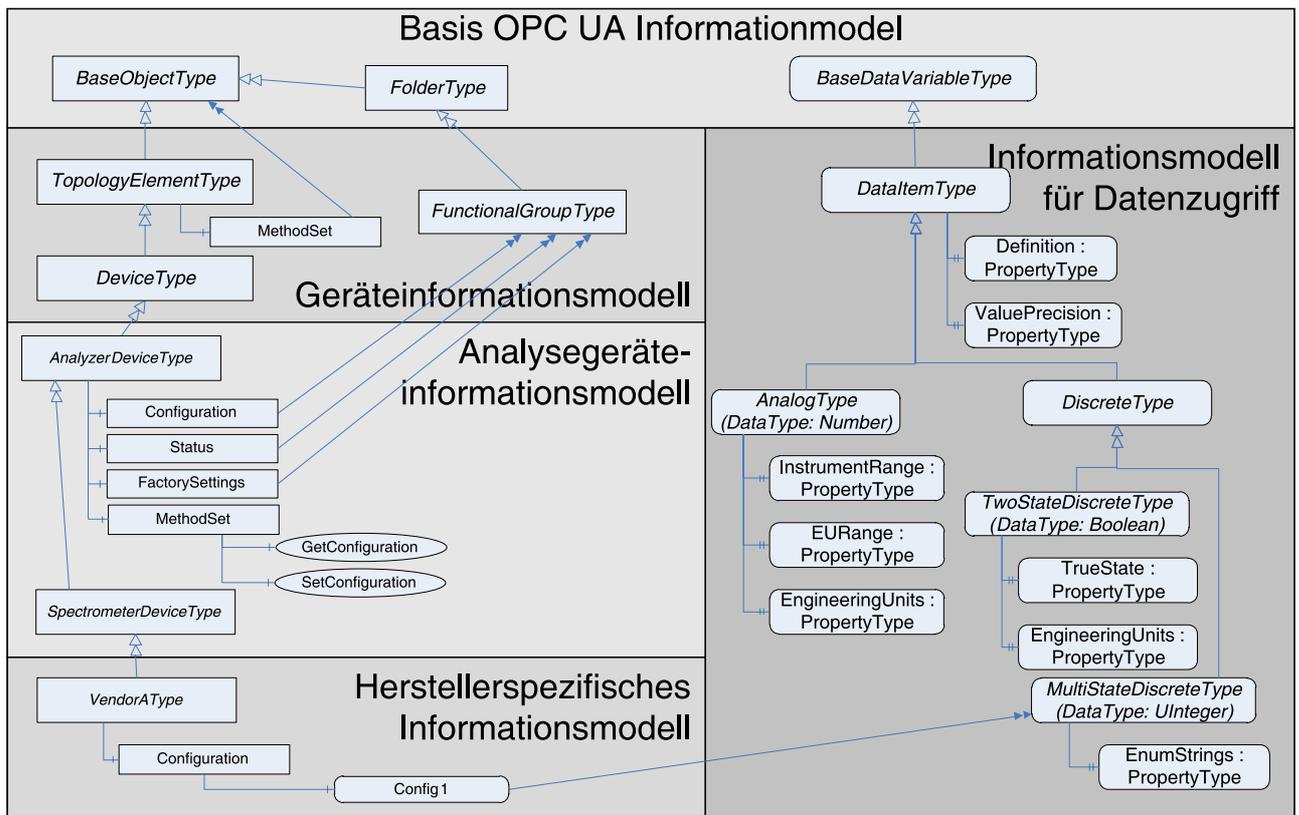


Bild 3 Beispiel standardisierter Informationsmodelle.

blen definiert werden, die spezifische Informationen eines Systems nach außen repräsentieren. Durch die Definition von Referenztypen können charakteristische Beziehungsstrukturen, die transparent gemacht werden sollen, festgelegt werden.

In Bild 3 ist ein Beispiel gezeigt, in dem mehrere standardisierte Informationsmodelle verwendet werden. Das Basis Informationsmodell liefert dabei die Basistypen wie *BaseObjectType* oder *FolderType*, mit dem ein einfacher Typ zur hierarchischen Organisation eines Adressraums gegeben ist. Das Data Access Informationsmodell (s. rechter Teil in Bild 3) verfeinert den *BaseDataVariableType* und bietet die Möglichkeit, Informationen über die Maßeinheit oder Genauigkeit einer Variablen anzugeben. Das Geräteinformationsmodell (s. linker Teil oben in Bild 3) definiert, wie Geräte repräsentiert werden sollen, zum Beispiel, wie Parameter eines Gerätes strukturiert werden. Während das Geräteinformationsmodell noch sehr generisch die Struktur von Geräten jeglicher Art definiert und damit sowohl für Feldgeräte wie Durchflussmesser und Temperatursensor aber auch z.B. für Controller verwendet werden kann, definiert das Analysegerätemodell (s. linker Teil Mitte in Bild 3) ein Informationsmodell für konkrete Klassen von Analysegeräten. Dabei werden spezifische Parameter und Methoden definiert, die ein Gerät einer bestimmten Klasse anbieten muss. Davon abgeleitet, repräsentiert das server- oder herstellerspezifische Informationsmodell (s. linker Teil

unten in Bild 3) spezielle Analysegerätetypen für die zusätzliche, systemspezifische Informationen bereitgestellt werden.

Die Notation, die in Bild 3 verwendet wird, ist eine an UML angelegte und in OPC UA standardisierte Notation eines OPC UA Adressraums, indem z. B. Rechtecke Objekte und Rechtecke mit Schatten Objekttypen darstellen. Die Referenzen zwischen den Knoten haben ebenfalls eine spezifische Semantik und repräsentieren verschiedene Referenztypen wie beispielweise eine Vererbung. Die Semantik der Referenztypen wird von den standardisierten Informationsmodellen festgelegt, im Beispiel bereits vom Basis Informationsmodell.

Das Geräteinformationsmodell [3] wurde initial für FDI (field device integration) [4] entwickelt. Es wurde später sowohl vom Analysegerätemodell verfeinert [5], als auch von einer kombinierten Arbeitsgruppe der PLCopen und der OPC Foundation verwendet, um Basistypen zur Definitionen eines Informationsmodells für IEC 61131-3 Sprachen zu definieren [6]. Alle drei Spezifikationen sind bereits veröffentlicht.

5 Kommunikationsinfrastruktur

Die Kommunikation von OPC UA basiert auf dem Client-Server Modell. Das bedeutet, dass ein Client eine Verbindung initialisiert und danach Information anfordert, die der Server dann als Antwort liefert. Innerhalb dieses Request-Response Modells bietet OPC UA darüber

hinaus die Möglichkeit, über das sogenannte Publishing Modell, Änderungen von Datenwerten oder abnormale Bedingungen wie Alarmerne an den Client zu verschicken [1].

OPC UA ermöglicht den Austausch von aktuellen Daten, Alarmen und Ereignissen, die Historie von beiden, den Aufruf von Methoden sowie den Zugriff auf Metadaten.

Die Kommunikation mit OPC UA bietet keine deterministische Echtzeit, d. h. es wird nicht garantiert, dass innerhalb einer bestimmten Zeitperiode Daten ausgetauscht werden. Mit verschiedenen Mechanismen wird allerdings sichergestellt, dass innerhalb einer konfigurierbaren Zeitperiode, die Daten entweder ausgetauscht wurden oder ein Kommunikationsfehler erkannt wurde. Daneben werden die ausgetauschten Daten mit echtzeitverwandten Informationen ausgestattet, indem ein Status (Qualität) und ein Zeitstempel angeboten werden.

Um auf Metadaten zuzugreifen, kann im Adressraum navigiert, Anfragen gestellt, sowie Attribute gelesen werden. Optional kann ein Client die Attribute auch schreiben oder die Struktur der Adressräume ändern (Hinzufügen und Löschen von Knoten und Referenzen). Dieses passiert auf Basis standardisierter Dienstaufträge. Insbesondere die Möglichkeit der Änderung der Struktur ist im Hinblick auf die Interoperabilität wichtig, da dies Aktivitäten des Engineerings über OPC UA ermöglicht. Es wird jedoch erwartet, dass viele Server diese Möglichkeit nicht oder lediglich eingeschränkt zur Verfügung stellen werden.

Aktuelle Daten können gelesen und geschrieben werden (wie beispielsweise ein Sollwert). Um immer den aktuellen Wert eines Datums zu haben, könnte die Lese-Operation häufig wiederholt werden (Polling). Eine Alternative stellt der Publish-Mechanismus dar, bei dem der Server Änderungen eines Wertes einem Client mitteilt. Dazu kann der Client für jeden ihn interessierten Wert ein Toleranzband und ein Intervall spezifizieren, in dem er Änderungen mitgeteilt bekommen möchte. So kann durch das Toleranzband spezifiziert werden, dass beispielsweise lediglich Änderungen von mehr als 3 Prozent mitgeteilt werden sollen (zum Beispiel bei Temperaturmessungen) und das höchstens alle 500 ms. OPC UA arbeitet ausnahmbasiert, d. h. lediglich Änderungen werden dem Client mitgeteilt. Das reduziert die Anzahl der ausgetauschten Daten. OPC UA unterstützt Überabtastung. Das bedeutet, dass ein Client nicht zwingend jede Änderung sofort mitgeteilt bekommt, sondern mehrere Änderungen auf einmal. Beispielsweise kann ein Server Änderungen jede Millisekunde erfassen aber lediglich alle 500 ms dem Client übertragen.

Alarme und Ereignisse werden ebenfalls über den Publish-Mechanismus übertragen. Mit Filtermechanismen auf Ereigniskategorien und Werten der Ereignisfelder sowie der geforderten Felder kann der Client spezifizieren, welche Daten übertragen werden sollen. Der Alarm-Status kann optional im Adressraum bereitgestellt

werden, so dass er auch direkt aus dem Adressraum gelesen werden kann. Das Quittieren von Alarmen erfolgt über Methodenaufrufe.

OPC UA bietet vielfältige Operationen für den Zugriff auf historische Daten. Neben dem Zugriff auf die rohen Daten, erlaubt es die Verwendung von Aggregationsfunktionen (min, max, usw.) und die Veränderung der gespeicherten Historie. Zugriffe auf historische Ereignisse erfolgen mit denselben Filtermechanismen wie aktuelle Ereignisse, wobei zusätzlich eine Zeitspanne angegeben werden muss.

6 Interoperabilität durch OPC UA

OPC UA bietet, wie oben dargestellt, erweiterte Möglichkeiten der Modellierung austauschbarer Systeminformationen auf Basis elementarer Datenobjekttypen, für die im Standard Kommunikationsdienste festgelegt wurden. Das ist ein wichtiger Schlüssel für die Leistungsfähigkeit von OPC UA zur Realisierung von Interoperabilität. Immer dann, wenn man es schafft, vereinheitlichte Informationsmodelle ausschließlich mit den Sprachmitteln von OPC UA zu definieren, beinhaltet das zugleich die Spezifikation des nutzbaren Nachrichtenaustauschs und die Festlegung der Kommunikationsinfrastruktur. In diesem Fall ist nämlich jeder Objekttyp, der im Informationsmodell benutzt wird von einem Kommunikationsobjekttyp wie z. B. „Nodetype“, „Eventtype“ oder auch „Variabletype“ abgeleitet, für die OPC UA bereits Vorgaben für den nutzbaren Nachrichtenaustausch vorsieht.

OPC UA zeigt im Gegensatz zu vielen anderen Kommunikationsstandards (u. a. das heutige, klassische OPC) kaum Restriktionen zur Informationsmodellierung.

Für jedes System, das als Informationsanbieter fungiert, muss festgelegt werden, welche Arten der Informationen und welche Informationssichten für den Zugriff von Client-Applikationen angeboten werden sollen. Dazu muss der Informationshaushalt auf einen eindeutigen Adressraum abgebildet werden. Die Festlegung systemübergreifender, gleichartiger Bedeutungen von Daten, erzielt man durch Typisierung von Datenobjekten und durch die Festlegung eindeutiger Attributbezeichnungen pro Datenobjekttyp. Eine Transparenz von Datenzusammenhängen erhält man durch das Konzept der Objektbeziehungen (References). Bei dieser Art der Systemintegration spricht man auch von „Whitebox“-Kommunikation (s. [7] und [8]).

6.1 Funktionaler Nutzen

Dadurch, dass mit Hilfe von OPC UA insbesondere auch Querbeziehungen zwischen Informationen transparent gemacht werden können und einzelne Informationselemente eindeutig typisiert werden können, eröffnen sich neue Möglichkeiten der Interoperabilität. Zudem bieten die Objektmanagementdienste (Objekte anlegen, löschen, ...) erweiterte Formen der Interoperabilität, die

über das klassische Lesen und Schreiben von Informationen hinausgehen. Die wesentlichen funktionalen Vorteile sind:

- Externe Anwendungen können Datenhaushalte von Fremdsystemen erkunden und somit gezielte Informationsrecherchen/-analysen durchführen, ohne dass Vorabinformationen explizit konfiguriert werden müssen. Das stellt die Voraussetzung zur Entwicklung systemunabhängiger Erkundungs- und Analyse-Werkzeuge dar. So ist es denkbar, eine systemunabhängige Anwendung zu nutzen, die ohne Konfigurationsaufwand auflisten kann, welchen Ventile zurzeit offen oder welche Regler aktuell auf Handbetrieb sind.
- Datenhaushalte externer Systeme können mit dem Wissen über die Bedeutung einzelner Datenobjekte verändert werden. Das ist die Voraussetzung zur Entwicklung offener, systemunabhängiger Engineeringwerkzeuge und auch moderner Konzepte wie z. B. Agentensysteme. Dabei ist jedoch zu beachten, dass die Bedeutungsinhalte einzelner Datenobjekte außerhalb von OPC UA in eindeutig beschriebenen Profildefinitionen hinterlegt und veröffentlicht sein müssen.
- Informationen externer Systeme können auf Grund des Zugangs zu deren Semantik auf eigene Daten abgebildet werden. Das ist die Voraussetzung zur Realisierung selbstkonfigurierender Systeme sowie autark interagierender Systemverbünde.

Um diese Funktionalitäten zu erreichen, genügen die Festlegungen von OPC UA jedoch nicht aus. Vielmehr müssen darauf aufbauend spezifische Informationsmodelle, so genannte „Profile“, für einzelne Anwendungsdomänen entwickelt werden.

6.2 Profilbildung

Eine wichtige Herausforderung besteht darin, Anwendungsdomänen zu identifizieren und für jede dieser Domänen ein „OPC UA Profil“ zu definieren. Diese Profildefinitionen sollte jeweils von einer repräsentativen Menge an Vertretern einer Community bzw. von deren Interessensgemeinschaften unterstützt werden.

Der Prozess einer Profildefinition kann auf zwei Ansätzen beruhen:

1. Bestehende, bereits standardisierte Informationsmodelle auf die grundlegenden Objekte und Datenstrukturen von OPC UA abbilden.
2. Mit dem Wissen um die Möglichkeiten von OPC UA neue Profile definieren.

Für den ersten Weg wurden in Abschnitt 4 Beispiele für bereits erarbeitete Profilbildungen aufgezeigt. Im Bereich der Prozessleittechnik und MES existiert darüber hinaus eine Vielzahl an akzeptierten Standards, deren Informationsmodelle in OPC UA Profile abgebildet werden können. Dazu zählen u. a. die IEC 62264 (bekannt als S95) [9], die IEC 61512 (bekannt als S88) [10], IEC 624242 (bekannt als CAEX) [11] oder auch das Pumpenprofil des VDMA [12].

Eine 1:1 Umsetzung der dort beschriebenen Informationsmodelle in OPC UA Strukturen ist jedoch nicht immer möglich oder sinnvoll, da entweder die Modelle noch nicht formalisiert sind oder die Modelle zu eng bzw. zu weit gefasst wurden. Hier müssen zunächst mögliche Anwendungsszenarien untersucht werden, um darauf aufbauend geeignete Abbildungen zu finden.

7 Anwendungsbeispiele gehobener Interoperabilität

7.1 Von offener Scada-Visualisierung bis zu neutralen SPS Engineeringwerkzeugen

Das Profil der Nutzerorganisation PLCopen verfolgt das Ziel offener, herstellerunabhängiger SPS-Anbindungen. Gemäß einer Visionsbetrachtung der PLCopen, stellt sich ein SPS-Projekt auf unterschiedlichen Steuerungsplattformen verschiedener Hersteller nach außen zur Visualisierung und den MES/ERP Ebenen vom Zugriff exakt identisch dar [13]. Damit entfallen nicht nur protokollspezifische Unterschiede bei der Kommunikation mit SPSen unterschiedlicher Hersteller, sondern es ergeben sich auch erweiterte Möglichkeiten der Wiederverwendung von Software und der Reduzierung von Konfigurationsaktivitäten. Durch eine Erkundung von CtrlFunctionBlockTypes können Visualisierungssysteme z. B. automatisch passende Faceplates instantiiieren und an die zugehörigen CtrlFunctionBlock Instanzen anbinden. Für andere typische Methoden, wie z. B. Zähler, können automatisch Subscriptions eingerichtet werden, um aktuelle Zählerwerte selbstkonfigurierend visualisieren zu können [14].

In einem zweiten Schritt sollen auf Basis des PLCopen Profils Funktionsbausteine mit den Mitteln von OPC UA herstellerunabhängig erstellt und in verschiedene SPSen geladen werden können. Auch Updates von Programmen sollen herstellerunabhängig erfolgen können. Ein erster Schritt in Richtung systemunabhängiger Engineeringwerkzeuge für speicherprogrammierbare Steuerungen.

7.2 Asset Management

Zur Konfiguration von Feldgeräten werden heute zwei Standards eingesetzt: Electronic Device Description Language (EDDL) und Field Device Tool (FDT). In einer gemeinsamen Arbeitsgruppe zwischen OPC Foundation, Profibus Nutzerorganisation (PNO), HART Foundation, Fieldbus Foundation (FF) und FDT Group wird derzeit ein allgemeingültiges Modell für die Konfiguration von Hardware- und Softwarekomponenten entwickelt. EDDL und FDT sollen zukünftig über den gemeinsamen Standard FDI unter der Verwendung von OPC UA zusammengeführt werden. In FDI soll ein Feldgerät über ein sogenanntes Gerätepaket beschrieben werden. Dieses setzt sich aus einer Parameterbeschreibung und Oberflächenelementen zusammen. Der FDI Standard sieht vor, dass der Konfigurationsserver ein OPC UA Server sein soll, der seinen Adressraum auf Basis der Gerätepakete

füllt. Die Konfigurationsoberflächen sind OPC UA Clients, die auf die Geräteparameter über OPC UA zugreifen und zur Anzeige die spezifischen Oberflächenelemente aus den Gerätepaketen verwenden.

Innerhalb der FDI Arbeitsgruppe wurden im Rahmen der Spezifikation Anwendungsfälle identifiziert, die verschiedenartige Aspekte des Asset Managements beim Betreiben von Anlagen umfassen: von der Inbetriebnahme und Übergabe bis hin zur laufenden Instandhaltung und Anlagenbedienung. Auf Basis eines im FDI Standard verankerten Gerätemodells können für diese Aufgaben Automatismen zur Konfiguration, Kalibrierung und auch für Zustandsdatenauslesungen und -verarbeitungen entwickelt und mit Hilfe von OPC UA umgesetzt werden.

7.3 Selbstkonfigurierende Systeme

Der Trend zur Gestaltung von IT-Systemverbänden produktionsnaher Anwendungen bringt es mit sich, dass erhöhte Pflegeaufwände auf Anlagenbetreiber zukommen. Insbesondere die Problematik der Konsistenzsicherung von Datenbeständen und Konfigurationsparametern ist dabei eine alltägliche Herausforderung.

In vielen Produktionsbetrieben werden aus diesem Grund eine enge Integration von Systemen und auch die Einführung erweiterter Anwendungen gemieden. Der Pflegeaufwand wird oftmals höher eingeschätzt als der Nutzen. Selbst bei Annahme eines langfristig positiven Aufwand/Nutzenverhältnisses wird der laufende Pflegeaufwand als momentan nicht leistbar betrachtet. Potentiale der Wertschöpfungserhöhung von Produktionsanlagen liegen damit brach.

Ein Ansatz zur Lösung dieser Aufgabe ist die Entwicklung selbstkonfigurierender Systeme. In [15] wurden bereits industriell eingesetzte Anwendungen aufgezeigt. Voraussetzung zur Umsetzung derartiger Anwendungen ist die Verwendung offener Systemschnittstellen, die neben dem Auslesen von Datenwerten auch das Erkunden von Engineeringdaten ohne a priori Wissen in systemunabhängiger Form bieten. Diese Voraussetzung ist mit den Möglichkeiten von OPC UA gegeben.

Beispiele für selbstkonfigurierende Systeme und Applikationen sind:

- **Controller Performance Monitoring (CPM):** Diese Anwendungen dienen der kontinuierlichen Performanzbewertung von Regelkreisen. Dazu lesen sie über Leitsystemschnittstellen in zyklischen Abständen aus dem PLS Signalsequenzen von Reglern ein und bewerten damit die Regelkreise anhand regelungstechnischer Leistungskennzahlen (z. B. Regelgüten, Oszillationsverhalten, etc). Zur Konfiguration von CPM Anwendungen müssen eine Vielzahl an Parametern für jeden zu betrachtenden Regelkreis konfiguriert werden. Fast alle dieser Konfigurationsdaten stehen aber zur Laufzeit in den angekoppelten Leitsystemen zur Verfügung. Bei Verwendung einer geeigneten OPC UA Serverschnittstelle im Prozessleitsystem können

vom externen CPM-Werkzeugen alle Regelkreise automatisch gefunden und die zur CPM-Konfiguration benötigten Daten ausgelesen werden.

- **Zentralisierte Langzeitarchive und Analyse-Werkzeuge** für Produktionsdaten: OPC UA sieht ein gemeinsames Datenmodell sowohl für online als auch historische Daten vor. Das ermöglicht die Entwicklung von Schnittstellen, über die eine konfigurationslose Archiv-Archivkopplung realisiert werden kann. Archive und Archivstrukturen inklusive aller zugehörigen Metainformationen verteilter Systeme können erkundet werden und in ein einheitliches Format eines zentralen Archivsystems ausfallsicher überführt werden. Damit können heutige Kopplungen zwischen Prozessleitsystemen und PIMS (Process Information Management Systems) selbstkonfigurierend umgesetzt werden.

8 Fazit

OPC UA bietet einen deutlichen Mehrwert im Hinblick auf eine Interoperabilität zwischen Systemen unterschiedlicher Hersteller und verschiedenartiger Anwendungsdomänen. Interoperabilität bedeutet mehr, als Daten von A nach B zu transportieren. Möglichkeiten der transparenten Informationsabbildung inklusive der Repräsentation von netzartigen Datenbeziehungen sowie von Metadaten einzelner Datenobjekte stellen die Grundvoraussetzungen dar, um semantische, systemunabhängige Interpretationen von Daten und damit einen höheren Grad der Informationsverarbeitung leisten zu können. Für verschiedenartige Anwendungsdomänen wurden dazu bereits spezifische Informationsmodelle, sogenannte Profile, entwickelt.

Moderne Konzepte der objekt-orientierten Informationsmodellierung, wie die Bildung komplexer Datenstrukturen oder hierarchisierbare Klassenkonzepte zur Typisierung von Datenobjekten sind in den OPC UA Standard eingeflossen. Ergänzt werden diese Mittel durch erweiterte Konzepte zur Spezifikation von Ablaufstrukturen und Rahmenvorgaben zur Definition von Programmen, die von externen Clients gesteuert werden können.

Durch die Möglichkeit der Spezifikation integraler Schnittstellen, über die sowohl Online-, Archiv- als auch Ereignisdaten innerhalb eines vereinigten Adressraumes verfügbar gemacht werden können, wird eine grundlegende Restriktion der bisherigen, klassischen OPC Spezifikationen aufgelöst.

Neben den Vorgaben zur Informationsmodellierung spezifiziert OPC UA Dienste zur Kommunikation zwischen OPC UA Clients und OPC UA Servern. Dazu zählen Dienste, die der IT-Security dienen, als auch Dienste der Anwendungsebene. Auf der Ebene der Anwendung bieten insbesondere die Objektmanagementdienste, über die mit Hilfe neutral formulierter Dienste, Datenobjekte in Fremdsystemen angelegt oder gelöscht werden können, erweiterte Möglichkeiten der Interoperabilität.



Der Standard OPC UA bietet weitaus mehr als das klassische OPC und stellt damit eine neue Stufe der Interoperabilität dar. Es liegt nun in der Verantwortung von Systemherstellern und Nutzergruppen, diese Möglichkeiten zu nutzen. Eine Gefahr besteht lediglich darin, dass durch die Vielfältigkeit der Konzepte innerhalb von OPC UA, immer die richtige Balance zwischen einfacher Handhabung und Eröffnung von Anwendungsmöglichkeiten bei der Entwicklung von Profilen gefunden werden muss.

Literatur

- [1] W. Mahnke, S.-H. Leitner, M. Damm: *OPC Unified Architecture*, ISBN: 978-3-540-68898-3, New York: Springer, 2009.
- [2] A. Ginter: *The Stuxnet Worm and Options for Remediation*, Industrial Defender, Inc., 2010.
- [3] OPC Foundation: *OPC Unified Architecture for Devices (DI)*, Version 1.00, December 2009.
- [4] D. Grossmann, D. John, A. Laubenstein: *EDDL Harmonisierung, atp – Automatisierungstechnische Praxis* 10-11 2009.
- [5] OPC Foundation: *OPC Unified Architecture Companion Specification for Analyser Devices*, Version 1.00, October 2009.
- [6] PLCopen and OPC Foundation: *OPC UA Information Model for IEC 61131-3*, Version 1.00, March 2010.
- [7] H. Albrecht: *On Meta-Modeling for Communication in Operational Process Control Engineering*. Dissertation. VDI Fortschritt-Bericht, Series 8, No. 975, VDI-Verlag, Düsseldorf, Germany.
- [8] U. Enste, J. Müller: *Datenkommunikation in der Prozessindustrie*. Oldenbourg Industrieverlag, München, 2007.
- [9] IEC: *Enterprise Control System Integration*, IEC 62264, 2003 (als Deutsche Norm: DIN EN 62264 Integration von Unternehmensführungs- und Leitsystemen, 2008).
- [10] IEC: *Batch Control*, IEC 61512, 2008.
- [11] IEC: *Specification for Representation of process control engineering requests in P&I Diagrams and for data exchange between P&ID tools and PCE-CAE tools*. IEC CDV 62424, 2007.
- [12] VDMA: *Device Profile for Liquid and Vacuum Pumps – Part 1: Device Information Model – Universal Profile Elements – Generic Pumps*. VDMA Einheitsblatt 24223-1. 2008.
- [13] PLCopen: *Introduction in the PLCopen and OPC UA Communications Model*. http://www.plcopen.org/pages/tc4_communication/downloads/intro_plcopen_opc_v3.pdf.
- [14] PLCopen: *PLCopen – OPC UA – Use Cases*. http://www.plcopen.org/pages/tc4_communication/downloads/plcopen_opcua_usecases.pdf.
- [15] Bukva, S., Enste, U., Uecker, F.: *Selbstkonfiguration und automatisiertes Änderungsmanagement von MES-Systemen*. atp-Automatisierungstechnische Praxis 08/2009.

Manuskripteingang: 2. Februar 2011



Dr.-Ing. Udo Enste ist Geschäftsführer der LeiKon GmbH, Herzogenrath. Seine Hauptarbeitsgebiete liegen in den Bereichen Planung und Konzeption von Leit- und MES-Systemen, industrielle Datenkommunikation und betriebliches Informationsmanagement. Herr Dr. Enste ist Mitglied in Gremien der DIN/DKE, VDI/GMA und der NAMUR. Die LeiKon GmbH ist Mitglied der OPC Foundation.

Adresse: LeiKon GmbH, Kaiserstr. 100, 52134 Herzogenrath, E-Mail: udo.enste@leikon.de



Dr.-Ing. Wolfgang Mahnke arbeitet am ABB Forschungszentrum in Ladenburg im Bereich industrieller Softwaresysteme. Er ist Mitglied der OPC UA Arbeitsgruppe und Editor von Teil 3 und Teil 5 der OPC UA Spezifikation. An der Universität Stuttgart erhielt Dr. Mahnke sein Diplom in Informatik und an der Technischen Universität Kaiserslautern seinen Dr. Ing. im Bereich von Datenbanken und Informationssystemen.

Adresse: ABB Forschungszentrum, Wallstadter Str. 59, 68526 Ladenburg, E-Mail: wolfgang.mahnke@de.abb.com



Oldenbourg
Verlag

Ein Wissenschaftsverlag der
Oldenbourg Gruppe

Die *at* der Zukunft

Jetzt an der Leserumfrage teilnehmen und iPod nano gewinnen!

Nehmen Sie sich fünf Minuten Zeit und gestalten Sie die zukünftige *at* – *Automatisierungstechnik* aktiv mit!

Wir möchten Ihnen ein paar kurze Fragen stellen, damit wir die *at* weiterentwickeln und an Ihre Anforderungen anpassen können!

Die Umfrage nimmt maximal fünf Minuten Ihrer Zeit in Anspruch.



Unter allen Teilnehmern
verlosen wir einen
iPod nano mit Multi-Touch!



Mehr Informationen und Link zur Teilnahme unter



www.at-automatisierungstechnik.de