Dezentrale Automation

OPC UA Service Sets und Profiles

David Riepl

0625016

5.5.2010

1. Einleitung

OPC Unified Architekturen ist ein Konstrukt welches alle bisherigen Spezifikation von OPC einschließt. Es stellt eine übergeordnete Instanz bzw. ein Modell zu Verfügung wie die Kommunikation in der Automatisierungstechnik ablaufen soll. Der große Vorteil ist das alle vorhanden Standards dieses Modell verwenden können und somit die Probleme die mit verschiedenen Protokollen und Herstellern mit sich kommen gelöst zu sein scheinen. Zumindest solange man sich an das Modell hält. Diese Arbeit enthält eine Detailausführung über die Konzepte von OPC UA Service Sets und POC UA Profiles.

2. OPC UA Services

2.1. Allgemeines

Ein Service beschreibt die Kommunikation zwischen einem OPC UA Client und dem Server.

In OPC UA wird auf Generic Services gesetzt. Eine Generic Service umfasst einen Use Case(Siehe Übersicht) und den für ihn relevanten Methoden. Somit gibt es für jede Aufgabe nur ein Service . Es soll verhindert werden das durch zu viele spezielle Methoden der ServiceKatalog zu unübersichtlich wird. Weiters sind die Services dazu ausgelegt nur größere Datenmengen zurückzugeben um so den Netzwerk Overhead zu minimieren

Sie API besteht aus sogenannten OPC UA Stacks welche es ermöglichen die verschiedenen Services in verschiedenen Programmiersprachen zu implementieren und diese sind somit auch unabhängig vom Kommunikationsmechanismus. Dies ist auch de große Unterschied zu klassischem OPC da hier die Entwicklung auf das Microsoft Component Object Model (COM/DCOM) gebunden ist. Auch die Plattformunabhängigkeit wird dadurch gegeben. Ein weiterer Vorteil von UPC UA ist das hier dank dem HTTP-Protokoll und dem XML Standard die Kommunikation über das Internet möglich ist.

2.2. Übersicht

Use case	Service sets or services
Find servers	Discovery Services Set
Connection management between clients and servers	Secure Channel Service Set Session Service Set
Find information in the Address Space	View Service Set
Read and write data and metadata	Read and Write Service
Subscribe for data changes and Events	Subscription Service Set Monitored Item Service Set
Calling Methods defined by the server	Call Service
Access history of data and Events	HistoryRead and HistoryUpdate Service
Find information in a complex Address Space	Query Service Set
Modify the structure of the server Address Space	Node Management Service Set

Abbildung [1] Services grouped by use cases

2.3. Discovery Services Set

Wie der Name dieses Service Sets schon zu vermuten gibt werden hier die Art und Weise wie OPC UA Server erkannt und verwaltet werden. Die folgende Abbildung beschreibt im Detail den Ablauf des Kommunikationsaufbaus.

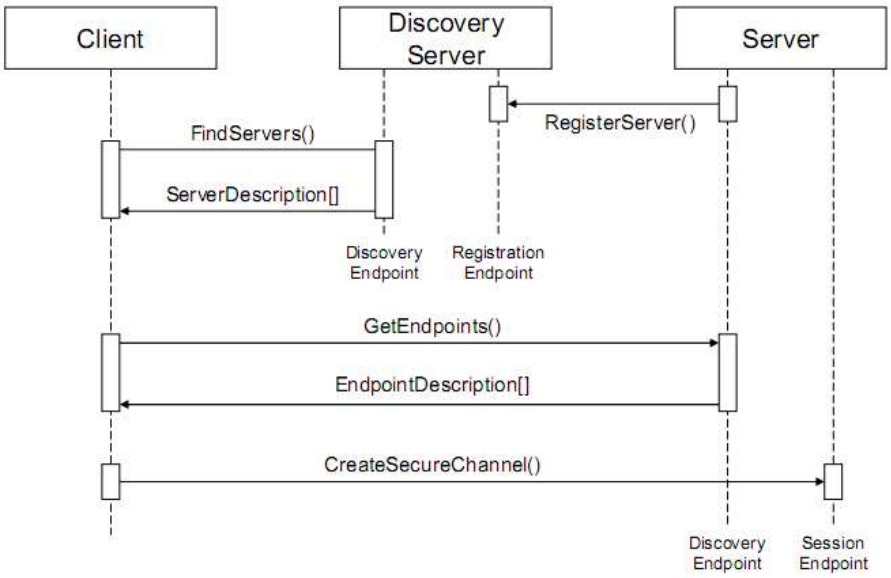

Abbildung [2] Discovery Process

Es gibt eine zentrale Stelle wo alle OPC UA Server erfasst werden. der Discovery Server. Der Client benutzt den Discovery Server um Informationen über die vorhanden und verfügbaren Server zu erhalten(Server Description). Nach erhalt dieser Daten kann der Client direkt mit dem Server in Verbindung treten. Ein Endpoint definiert sämtliche Rahmenbedingungen die nötig sind um mit einem Server in Verbindung zu treten(Netzwerk Protokoll, Security Settings). Die Informationen über alle verfügbaren Endpoints eines Servers werden vom Client angefragt. Es ist natürlich sinnvoll das ein Server möglichst viele Endpoints besitzt, sprich viele unterschiedliche Möglichkeiten mit ihm in Verbindung zu treten, da das OPC UA Netzwerk aus vielen unterschiedlichen Clients mit unterschiedlichen Kommunikationsmöglichkeiten besteht. Wenn der Client einen der Endpoints des Servers unterstützt kann mit der Übertragung begonnen werden.

In Abbildung 2 ist zu erkenn das das Discovery Service Set aus insgesamt 3 Services besteht

FindServers()	Gibt eine Lister der Verfügbaren Server zurück. Diese Service ist auf jedem OPC UA Server implementiert. Sollte ein OPC UA Netzwerk nur aus einem Server bestehen(also kein dezidierter Discovery Server) so muss der Server sich selbst zurückgeben.
GetEndpoints()	Gibt Informationen über das verwendete Protokoll, den Kommunikationskanal, Art der Authentifikation, verwendete Verschlüsselungsalghorithmen, usw. zurück.
RegisterServer()	Dieses Service wird nur von Servern ausgeführt. In periodischen Abständen registriert sich jeder Server am Discovery Server und dieser kann somit die Verfügbarkeit eines Servers abbilden (Heartbeat). Es ist weiters ein sicherer Kanal nötig damit sich nur vertrauenswürdige Server registrieren können. Mittels diesem Service werden keine Detailinformationen über die Endpoints ausgetauscht, sprich der Discovery Server weis nur. Welche Server gibt es? Wie kann man sie erreichen?

2.4. Secure Channel Service Set

Zunächst muss zwischen Secure Channel und Session Establishment unterschieden werden. Ersteres wird ausschließlich vom OPC Communication Stack durchgeführt und beschriebt den logischen Übertragungskanal wobei eine Session sich auf dem Application Level befindet und eben von dieser verwaltet wird.

Auf dieser Grafik ist sehr gut zu erkennen wie die verschiedenen Layer ineinandergreifen. Es wird zwischen low-level und high-level Network Transport unterschieden, wobei low-level die niederen Layer(Transport, Secure Channel) und high-level dien höher gelegenen Session Layer beschreibt

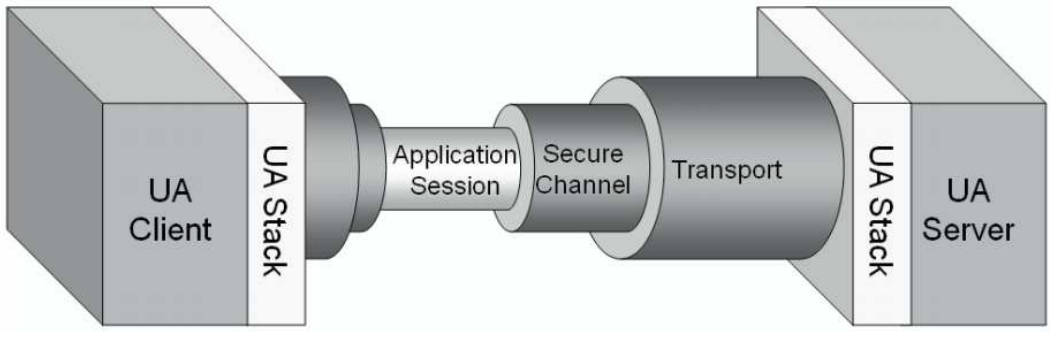

Abbildung [3] Different Levels of Communication channles

Ein Secure Channel ist durch eine eindeutige ID definiert die jeden Channel einem Client Server Paar zuordnet. Weiters besitz ein Secure Channel einen Security Token welcher die zur Verschlüsselung verwendeten Keys enthält. Diese Token werden in regelmäßigen Abständen erneuert

2.5. Session Service Set

Wie bei einem Secure Channel besitz auch eine Application Session eine eindeutige ID und einen autenticationToken. Die Bedeutung ist der beim Secure Channel beinahe ident. Zum Erstellen einer Session ist ein Secure Channel zwischen Client und Server erforderlich. Folgende Services sind hier definiert:

CreateSession	Dieses Service überprüft ob die Erstellung einer Session zwischen 2 Applikationen theoretisch möglich (freie Resourcen) ist noch bevor die Authentifizierungsdaten übermittelt werden.
ActivateSession	Hiermit wird eine Session aktiviert und ist erst dann bereit zur Verwendung. Der Client übermittelt sein ClientSoftwareCertificate
Close Session	Dient zum Beenden einer Session. Die belegten Resourcen werden freigegeben
CancleOutstandingService Requenst	Dieses Service sollte aufgerufen werden bevor die Session beendet wird

2.6. View Service Set

Im View Service Set sind Services implementiert die es erlauben durch den Adressraum zu navigieren. Im Vergleich zu den vorherigen OPC Versionen gibt in OPC UA einen gemeinsamen Adressraum und demnach auch gemeinsame Methoden für alle Funktionalitäten der früheren OPC Versionen. Das Service Set kann grob in 2 Bereiche eingeteilt werden. Browse und Read

2.6.1. Browse Service

Da die Netzstruktur bei OPC UA ein meshed Network ist funktioniert das Prinzip folgendermaßen. Ein Knoten ruft bei einem Nachbarknoten den Browse Service auf und dieser gibt eine Liste der an ihm anhängenden Knoten zurück. So kann sich der erste Knoten mittels durchwandern des Netzwerkes einen Netzwerkplan (Browse Tree) erstellen.

2.6.2. Read Service

Um den Browse Tree zu vervollständigen, also zu den nach dem browsen erhaltenen Daten(Devicename, Devicetype) noch erweiterte Daten zu bekommen(sämtliche Attribute), wird das Read Service verwendet.

2.7. Read and Write Services

Diese beiden Services sind selbstsprechend. Sie dienen dem Datenaustausch zwischen 2 oder mehreren OPC UA Nodes. Die Read/Write Services sind optimiert für größere Datenmengen(Bulk Transfer). Die Lese- und Schreibrechte werden durch spezielle Attribute (Access Level) gesteuert

2.8. Monitored Items and Subscription Service Set

Neben den klassischen Read/Write Operationen gibt es noch eine weitere Möglichkeit um an Informationen zu kommen. Mittel monitored Items und subscriptions kann ein Client die für ihn relevanten Daten(monitored Item) abonnieren. Wenn sich am wert des Monitored Items etwas ändert wird an den Client dieser Wert übermittelt(Notification). Alles Monitored Items haben die selben Attribute(Monitoring Mode, Filter, Queue Size) und unterschieden sich nur in dem Überwachten Variablentyp. Es gibt drei verschiedene Arten wie ein Monitored Item abonniert werden kann(Subscription Methods).

Subscription für Datenänderungen (DataChanges)	**In einem definierten Samplingintervall wird der Wert einer Variable Zyklisch überprüft und an den Client gesendet. Dies wird verwendet wenn der Client über eine kontinuierliche Wertfolge verfügen muss um z.B. Trends berechnen zu können(Bsp.: Temperatur steigt pro Minute um 1 Grad ==> Trend: in ein paar Stunden wird es sehr heiß)**
Subscription für Events	Events Subscriptions werden Ereignisbedingt an den Client gesendet. Dies ist sinnvoll um bei Übersteigung von einem definierten Schwellenwert einen Alarm zu generieren
Subscription für Aggregat Values	Hier wird das Monitored Item mit einer höheren Samplingrate abgetastet und zusätzlich werden die Daten nach einem Schema transformiert und erst dann zum Client gesendet

Die folgende Darstellung zeigt den Zusammenhang zwischen Monitored Items und den Subscription Methods.

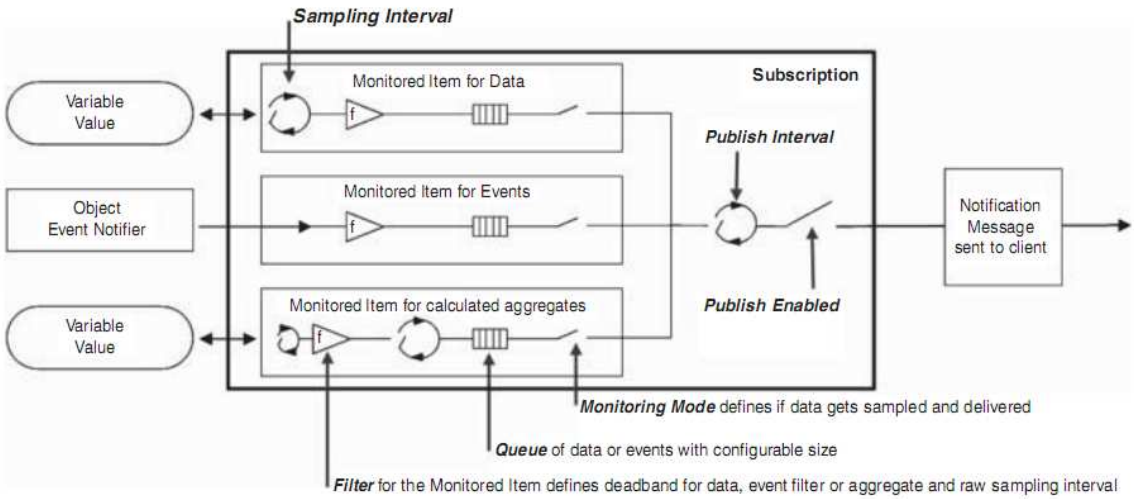

Abbildung [4] Settings for Subscription an Monitored Items

Das Publishing Service wird vom Client genutzt um den Server aufzufordern eine Notification zu den gewünschten Monitored Items zu versenden. Diese Publishing Requests landen in der Publishing-queue des Servers. Mittels des Republish Services kann der Server dazu aufgefordert werden bereits versendetet Notifications erneut zu senden.

2.9. Query Service Set

Dieses Service Set ist eine Erweiterung des View Service Sets. Da es in großen Netzwerken sehr schwierig ist die gewünschten Daten zu finden da das Browse Service sehr viele Knoten zurückgibt wurden im Query Service Set eine Features hinzugefügt. Hier ist es möglich mittels Filtern die Anzahl der von dem Browse Service zurückgegeben Knoten zu reduzieren. Diese Filter basieren auf dem Datentyp der gesucht wird(DataToReturn).

3. OPC UA Profiles

3.1. Allgemeines

Profile in OPC UA sind aus der Anforderung heraus entstanden das verschiedene Applikation auf verschiedenen Endgeräten verschiedener Funktionalität aufweisen. Um mit solche speziellen Applikationen umzugehen werden Profiles verwendet. Profiles können zur Klassifizierung von Applikationen verwendet werden. Wie wird jetzt genau ein Profile einer Applikation zugeordnet? Zu jeder in einem Profile definierten Funktionalität gibt es einen sogenannten Test Case. Von einer Unabhängigen Instanz wird nur die Applikation diesen Test Cases oder Szenarien unterzogen. Bei bestehen entspricht die Applikation dem angewendeten Profile und es wird ein Zertifikat ausgestellt. Weiters stellen Profiles eine Art Secrutity Feature zur Verfügung. Bei dem Aufbau einer Kommunikation zwischen 2 OPC UA Komponenten können die verwendeten Profile als Zugriffskriterium benutzt werden. Ein Client darf somit nur Daten abfragen wenn er dem Profil X und/oder Y entspricht. Hier wird auch ein weiteres Detail angedeutet. Profiles können natürlich mehrfach auf eine Applikation zutreffen, gemeinsame Funktionalitäten haben und andre Profile beinhalten.

Wie Profiles detailiert aufgebaut sind erläuter folgende Darstellung

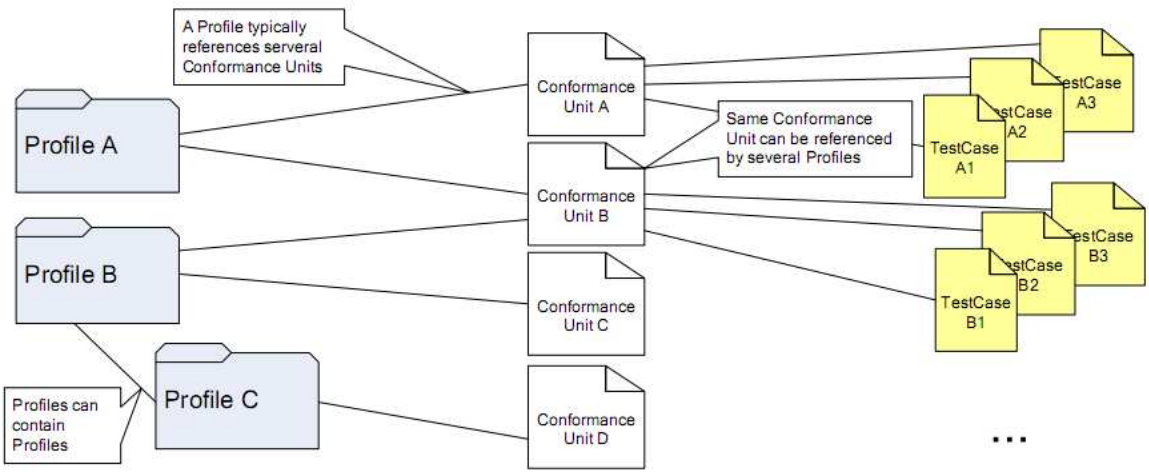

Abbildung [5] Conformance Units and test Cases

Hier ist zusehen das ein Profile aus einem oder mehreren Conformance Units besteht auf welche wiederum mehrere Test Cases angewendet werden. Die Conformance Units sind sozusagen die elementaren Bausteine aus den die Profiles zusammengebaut werden können. Bei der Zertifizierung wird die Funktionalität der Applikation auf die der Conformance Units herunter gebrochen und anhand der Test Cases verifiziert.

3.2. Klassen von Profiles

Ein Profile kann einer der Folgenden Klassen angehören welche im Folgenden im Detail besprochen werden. Zuvor müssen jedoch 2 Begriffe eingeführt werden

full-featured Profiles sind Profiles welche eine sehr große und breite Funktionalität aufweisen, sprich aus vielen Conformance Units bestehen. Diese Profiles werden bei Servern verwendet da diese mit mehreren verschiedenen Clientes (mit verschiedenen Profilen) kommunizieren müssen verwendet

Facets beschreiben im Gegenteil nur eine sehr geringe aber spezielle Funktionalität und werden zusätzlich zu einem full-featured Profile verwendet.

server-related	Einem Server muss mindestens ein full-featured Profile zugeordnet sein. In der Praxis wird das Profile Set eines Serverapplikation aus mehreren full-featured Profilen bestehen, welche durch Facets ergänzt werden.
client-related	Das Profile Set eines Clients besteht nur aus Facets da dessen Funktionalität im Allgemeinen überschaubar ist
security-related	Die security-related beschreiben Art und Weise der Verschlüsselung
transport-related	Die transport-related Profiles beschreiben die Protokolle die zu Kommunikation verwendet werden. Hier ist es wiederum sinnvoll im Server mehrere Transport-related Profiles zuzuweisen(viele verschiedene Clients)

4. Zusammenfassung

Die Services in OPC UA sind als Generic Services definiert. Sie sind sehr allgemein beschrieben und stellen ein Modell für die Datenkommunikation dar. Weiters sind sie abstrakt definiert um die Implementation auf verschiedenen Plattformen zu ermöglichen.

5. Abbildungsverzeichnis

6. Quellen

OPC Unified Architecture, W. Mahnke, Springer Verlag

DIN EN 62541-4

www.**opc**foundation.org

UPC UA, Rene Portmann