
Wie funktionieren OPC UA Companion Specifications

ACPLT Sommerkolloquium 2015

Julius Pfrommer

open62541



Agenda

Drei Sichten auf OPC UA

- Ein Netz aus Knoten
- Ein objektorientiertes Informationsmodell
- Eine Menge Services

... jeweils mit Beispielen aus der MTConnect Companion Specification.

OPC UA und seine Companion Specifications

OPC UA Spec

Teile

1. Concepts (~30p)
2. Security Model (~35p)
3. Address Space Model (~123p)
4. Services (~197p)
5. Information Model (~126p)
6. Mappings (~87p)
7. Profiles (~98p)
8. Data Access (~29p)
9. Alarms and Conditions (~93p)
10. Programs (~50p)
11. Historical Access (~49p)
12. Discovery (~53p)
13. Aggregates (~89p)

OPC UA Companion Specifications

- ISA-95, PLCOpen, Autold, AutomationML, MTCConnect, ...
- Der Standard definiert den Namespace 0
- Companion Specs und nutzerdefinierte Namespaces sind über eine eindeutige URI gekennzeichnet

OPC UA definiert ...

EIN NETZ AUS KNOTEN

Ontologien und Triple-Stores

Entitäten und ihre Relationen sind als **Triple** gespeichert:

(**Subjekt**, **Relation**, **Objekt**)

Beispiel 1:

- (Bob, hasChild, Alice)
 - (Bob, hasChild, Carol)
 - (Alice, likes, Chocolate)
 - (Alice, hasAge, 8)
- Welche Kinder von Bob mögen Schokolade?

Beispiel 2:

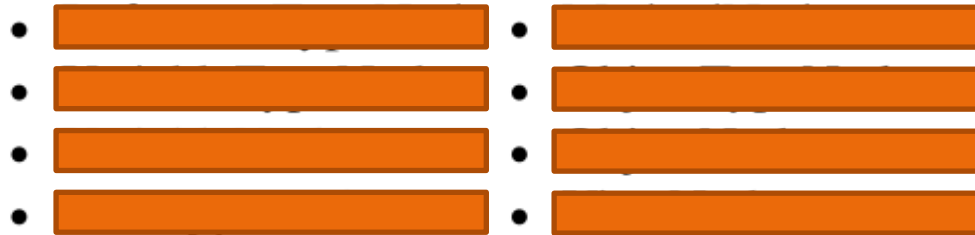
- (Fruit, subclassOf, Food)
 - (Banana, isA, Fruit)
 - (Banana, hasColor, Yellow)
- Welche Lebensmittel sind gelb?



Aber: Inferenz unterstützt keine numerischen Berechnungen

Knoten und Referenzen in OPC UA

8 Arten von Knoten



1. Ein OPC UA Informationsmodell ist ein Netz aus Knoten und Referenzen zwischen den Knoten

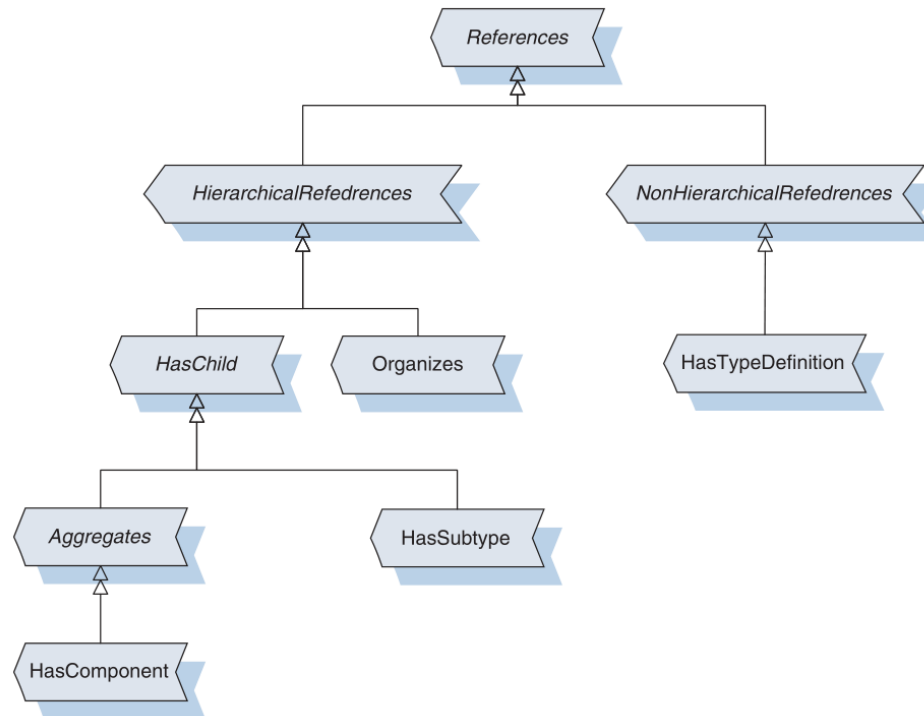
(Quelle, ReferenzTyp, Ziel)

2. Knoten enthalten Standard-Attribute

- Identifier
- BrowseName
- NodeClass (1-8)
- ...

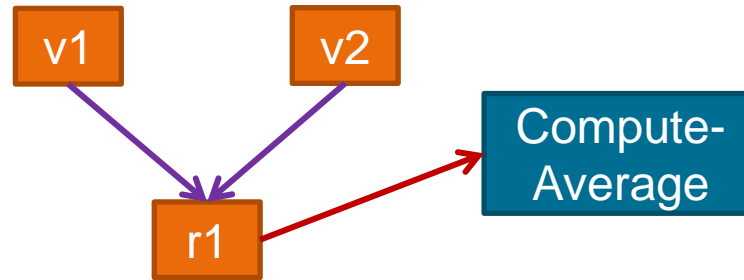
3. ... und spezielle Attribute je Knotenart

Referenz-Typen in OPC UA



- Die Referenzen zwischen den Knoten sind typisiert
- Hierarchische Referenzen dürfen keine Zyklen bilden
 - Jeder Knoten hat eine hierach. Referenz zu einem Elternknoten

Beispiel aus MTConnect: Referenztypen



→ ComputedBy

- The *ComputedBy* reference is a Subtype of *NonHierarchicalReferences*
- This reference is used to indicate that the value of the given variable is calculated using the targeted *AggregateFunctionType*

→ SourceOf

- The *SourceOf* reference is a Subtype of *NonHierarchicalReferences*.
- This reference is used to indicate the actual source associated with the value of the item being reported

Variablen und VariablenTypen

8 Arten von Knoten

- ReferenceTypeNode
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

- Zusätzliche Attribute von Variablen- und VariablenTyp-Knoten
 - DataType
 - ValueRank (Skalar, Array, egal)
 - Value
 - ArrayDimensions
 - ...
- Variablen zeigen mit einer HasTypeDefinition-Referenz auf den Variablentyp

Beispiel aus MTConnect: VariablenTypen

MTSampleDataItem






Attribute	Value				
BrowseName	MTSampleDataItem				
IsAbstract	False				
DataType	Number				
ValueRank	-1 (-1 = Scalar)				
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Inherit the <i>Properties</i> of the <i>AnalogItem</i> which is defined in [UA Part 8]					
HasProperty	Variable	CoordinateSystem	CoordinateSystem TypeEnum	PropertyType	Optional
HasProperty	Variable	NativeUnits	EUInformation	PropertyType	Optional
HasProperty	Variable	NativeScale	Float	PropertyType	Optional
HasProperty	Variable	SampleInterval	Duration	PropertyType	Optional

ElectricalEnergyType (→ Verbrauch in Watt)

Attribute	Value				
BrowseName	ElectricalEnergyType				
IsAbstract	False				
DataType	Number				
ValueRank	-1 (-1 = Scalar)				
Reference	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of MTSampleDataItem					

DatenTypen

8 Arten von Knoten

- ReferenceTypeNode • 
- VariableTypeNode • 
- VariableNode • 
-  • 

- 25 Builtin Types (Int32, String, Nodeld, ...)
- Enumerations
- Komplexe Datentypen (Strukturen)
 - In Namespace 0
 - Nutzerdefiniert

DatenTypen

XML-Snippets mit standardisiertem Schema

```
<UADatatype NodeId="i=12171" BrowseName="ComplexNumberType">
  <DisplayName>ComplexNumberType</DisplayName>
  <References>
    <Reference ReferenceType="HasSubtype" IsForward="false">i=22</Reference>
  </References>
  <Definition Name="ComplexNumberType">
    <Field Name="Real" DataType="i=10" />
    <Field Name="Imaginary" DataType="i=10" />
  </Definition>
</UADatatype>
```

Structure

Float

```
struct ComplexNumberType {
    float real;
    float imaginary;
}
```

Beispiel aus MTConnect: DatenTypen

BufferSizeDataType

Attribute	Value
BrowseName	BufferSizeDataType
IsAbstract	False
Subtype of integer	

DecibelValueDataType

Attribute	Value
BrowseName	DecibelValueDataType
IsAbstract	False
Subtype of string	

DirectionTypeEnum





Enum	Description
CLOCKWISE_1	
COUNTER_CLOCKWISE_2	

OPC UA definiert ...

EIN OBJEKTORIENTIERTES INFORMATIONSMODELL

Methoden




8 Arten von Knoten

- ReferenceTypeNode • 
- VariableTypeNode • 
- VariableNode • 
- DataTypeNode • 

- Einfache Remote Procedure Calls
- Method-Knoten hat Kind-Variablen namens “InputArguments” und “OutputArguments”
- Mehrere Knoten können mit einer *hasComponent* Referenz auf denselben MethodNode zeigen
- Die ID des aufrufenden Knotens wird neben den Argumenten übergeben

Objekte und Objekttypen

8 Arten von Knoten

- ReferenceTypeNode
- VariableTypeNode
- VariableNode
- DataTypeNode
- MethodNode
- 
- 
- 

- Objekttypen definieren Komponenten (Mandatory, Optional)
 - Variablen, Methoden
 - Merkmale (*hasProperty*) und Komponenten (*hasComponent*)
- Objekttypen sind “Kopiervorlagen” für die Instanziierung
- Mehrfachvererbung ist prinzipiell möglich, aber unterspezifiziert

Beispiele aus MTConnect (ObjectTypes)

MTDeviceType

Attribute	Value3				
BrowseName	MTDeviceType				
IsAbstract	False				
Reference	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
Is a Subtype of UA: BaseObjectType					
HasComponent	Variable	Availability	String	DataltemType	Mandatory
HasProperty	Variable	Manufacturer	String	PropertyType	Mandatory
HasProperty	Variable	SerialNumber	String	PropertyType	Mandatory
HasProperty	Variable	Configuration	String	PropertyType	Optional
HasProperty	Variable	SampleInterval	Duration	PropertyType	Optional
HasComponent	Object	Dataltems		FolderType	Mandatory
HasComponent	Object	Components		FolderType	Mandatory
HasComponent	Object	Conditions		FolderType	Optional

MTAxesType

Attribute	Value				
BrowseName	MTAxesType				
IsAbstract	False				
Reference	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
Is a Subtype of MTDeviceType					
HasProperty	Variable	LinearAxesX	String	PropertyType	Optional
HasProperty	Variable	LinearAxesY	String	PropertyType	Optional
HasProperty	Variable	LinearAxesZ	String	PropertyType	Optional
HasProperty	Variable	RotaryAxesA	String	PropertyType	Optional
HasProperty	Variable	RotaryAxesB	String	PropertyType	Optional
HasProperty	Variable	RotaryAxesC	String	PropertyType	Optional

OPC UA definiert ...

EINE MENGE SERVICES

OPC UA Services

```
struct ReadValueId {  
    NodeId nodeId;  
    uint32_t attributeId;  
    String indexRange;  
    QualifiedName dataEncoding;  
};
```

```
struct ReadRequest {  
    RequestHeader requestHeader;  
    double maxAge;  
    enum TimestampsToReturn tsToReturn;  
    int32_t nodesToReadSize;  
    ReadValueId *nodesToRead;  
};
```

- Jeder Service-Aufruf ist eine Nachricht (Request/Response)
 - Wird vom Client ausgelöst (nicht Peer2Peer)
- Nachrichten werden asynchron ausgetauscht
 - Es können vom Client mehrere Nachrichten verschickt werden ohne auf eine Antwort zu warten
 - Der Server kann eingegangene Nachrichten in beliebiger Reihenfolge und mit beliebiger Verzögerung beantworten
- Subscriptions folgen auch dem Request/Response Prinzip
 - Der Server “hortet” Requests und sendet eine Response, sobald ein Ereignis eintritt (oder ein Interval abläuft)

Danke fürs Zuhören

Fragen dürfen gerne gestellt werden

Julius Pfrommer,
Abteilung Informationsmanagement und Leitechnik (ILT)
Fraunhofer IOSB

julius.pfrommer@iosb.fraunhofer.de
www.mes.fraunhofer.de
www.klkblog.de
Tel.: +49-721-6091-286
Fax: +49-721-6091-413

Übersicht aller Knoten-Attribute

```
enum AttributeId {  
    ATTRIBUTEID_NODEID = 1,  
    ATTRIBUTEID_NODECLASS = 2,  
    ATTRIBUTEID_BROWSENAME = 3,  
    ATTRIBUTEID_DISPLAYNAME = 4,  
    ATTRIBUTEID_DESCRIPTION = 5,  
    ATTRIBUTEID_WRITEMASK = 6,  
    ATTRIBUTEID_USERWRITEMASK = 7,  
    ATTRIBUTEID_ISABSTRACT = 8,  
    ATTRIBUTEID_SYMMETRIC = 9,  
    ATTRIBUTEID_INVERSENAME = 10,  
    ATTRIBUTEID_CONTAINSNOLOOPS = 11,  
    ATTRIBUTEID_EVENTNOTIFIER = 12,  
    ATTRIBUTEID_VALUE = 13,  
    ATTRIBUTEID_DATATYPE = 14,  
    ATTRIBUTEID_VALUERANK = 15,  
    ATTRIBUTEID_ARRAYDIMENSIONS = 16,  
    ATTRIBUTEID_ACCESSLEVEL = 17,  
    ATTRIBUTEID_USERACCESSLEVEL = 18,  
    ATTRIBUTEID_MINIMUMSAMPLINGINTERVAL = 19,  
    ATTRIBUTEID_HISTORIZING = 20,  
    ATTRIBUTEID_EXECUTABLE = 21,  
    ATTRIBUTEID_USEREXECUTABLE = 22  
};
```