

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA  
ELEKTROTECHNIKY A INFORMATIKY

# Sémantická interoperabilita v OPC Unified Architecture

Učebný text

Oblasť informačné, komunikačné a riadiace systémy

Ing. Rudolf Pribiš (2020)

## Obsah

1. Úvod.....	3
2. Úvod do problematiky .....	3
3. Sémantická interoperabilita v priemysle .....	6
4. Sémantická interoperabilita v OPC Unified Architecture .....	9
4.1. Adresný priestor (Address space) .....	11
4.2. Informačné modely (Information models) .....	12
4.3. Sady servisov (Service sets).....	14
4.4. Profily .....	16
5. Praktická implementácia OPC Unified Architecture servera a klienta.....	17
6. Záver.....	19
Zoznam literatúry.....	20

OBRÁZOK 1 - AVEVA INSIGHT .....	4
OBRÁZOK 2 – JSON OBJEKT AKO ZDROJ DÁT.....	4
OBRÁZOK 3 - IO-LINK AKO ZDROJ DÁT .....	5
OBRÁZOK 4 - OPC SERVER AKO ZDROJ DÁT.....	5
OBRÁZOK 5 - DÁTA BEZ DODATOČNÝCH INFORMÁCIÍ .....	6
OBRÁZOK 6 - DÁTA S DODATOČNÝMI INFORMÁCIAMI.....	6
OBRÁZOK 7 - MQTT –OBOJSMERNÁ KOMUNIKÁCIA (BEŇO, 2019) .....	7
OBRÁZOK 8 - PRÍKLAD DÁT V JSON (DAGUE, 2018).....	8
OBRÁZOK 9 – PRÍSTUP K JSON DÁTAM (DAGUE, 2018).....	8
OBRÁZOK 10 – AUTOMATIZAČNÁ PYRAMÍDA.....	10
OBRÁZOK 11 – OPC UA UNIFIED ARCHITECTURE - OD SENZORA DO CLOUDU (OPCFONDATION.ORG, 2019) ....	10
OBRÁZOK 12 – GRAFICKÉ ANOTÁCIE UZLOV REFERENCIÍ.....	11
OBRÁZOK 13 – ZOBRAZENIE INFORMAČNÉHO MODELU V ADRESNOM PRIESTORE .....	13
OBRÁZOK 14 – OBJEKTY OPC UA SERVERA ZOBRAZENÉ V UA EXPERT .....	17
OBRÁZOK 15 –GRAFICKÁ ANOTÁCIA OPC UA INFORMAČNÉHO MODELU PRE VÁHU .....	18
OBRÁZOK 15 –PRÍKLAD POUŽITIA OPC UA SERVERA A KLIENTA NA VLOŽENOM ZARIADENÍ .....	18

## 1. Úvod

Dnešná doba v oblasti automatizácie je charakteristická digitalizáciou a informatizáciou procesov na všetkých úrovniach. Jedným z cieľov takejto digitalizácie a informatizácie je vytvorenie tzv. *insight* (náhľadu, nahliadnutiu), ktorý umožňuje akési vnútorné nahliadnutie do procesov, napríklad v priemyselnej výrobe, a porozumenie vzťahov na základe získaných dát. Sledovanie a analýza spotreby energie umožňuje jej optimalizáciu a úsporu nákladov a v neposlednom rade aj zelenší priemysel.

Obrovský pokrok v oblasti integrovaných obvodov zdostupnil a zlacnil pripojenie snímačov a akčných členov k sieti internetu, prípadne intranetu, vďaka čomu nastal boom internetu vecí z a.j. Internet of Things (IoT) i priemyselného internetu vecí z a.j. Industrial Internet of Things (IIoT). S veľkým množstvom týchto zariadení vzniklo aj veľké množstvo interpretácií dát, ktoré poskytujú sieti.

Tento text vysvetľuje potrebu, priam až nevyhnutnosť, dodatočných dát, ktoré poskytujú významové porozumenie dátam (sémantickú interoperabilitu) získaných z IIoT a IoT. Pre ľahšie ozrejmienie tejto problematiky, text obsahuje príklad z praxe a niekoľko možných spôsobov prenosov dát zo zariadenia do miesta spracovania.

Jadrom tohto textu je vysvetlenie sémantickej interoperability v OPC Unified Architecture (OPC UA) ako aj základných častí OPC UA, ktoré túto interoperabilitu umožňujú. Ďalšie informácie ohľadom OPC UA sú dostupné v *Návrh a tvorba OPC UA adresného priestoru (Pribiš, 2019a)*, *OPC UA – história a špecifikácia – dočasný učebný text (Pribiš, 2018)*, *OPC UA – Interoperabilita pre priemysel 4.0 a internet vecí (IoT) (KEGA, 2018)*, *An Industrial Communication Platform for Industry 4.0 - case study (Pribiš et al, 2020)*.

Posledná časť textu je venovaná prehľadu textov a publikácií praktických implementácií OPC UA servera, klienta.

## 2. Úvod do problematiky

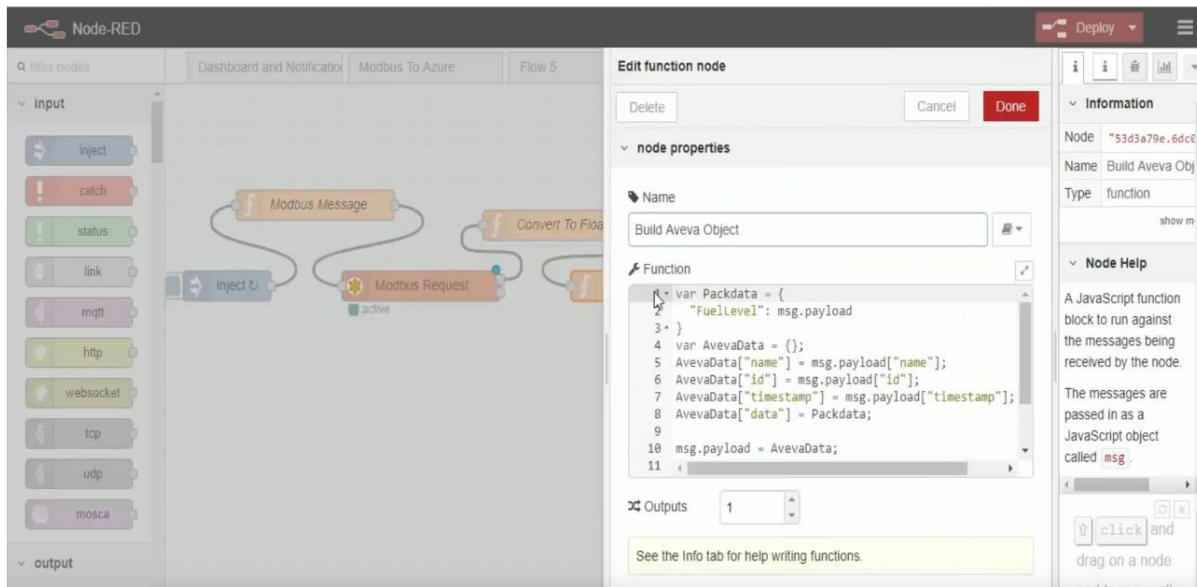
Predstavme si napríklad takú výrobu sušeného mlieka pre dojčatá. Výroba sa dá veľmi zjednodušene rozdeliť na mokrú a suchú časť. V mokrej časti výroby je medziprodukt prepravovaný rozpustený vo vode. Prechod z mokrej do suchej časti sa deje v tzv. *spray dryer* (sprejovom sušiči), ktorý si možno predstaviť ako trojposchodovú nádrž do ktorej sa vstreávajú rozpustené ingrediencie vo vode pomocou vodnej pary. V suchej časti to je zas vzduch, ktorý presúva ingrediencie a medziprodukt. Čiže voda, vodná para a vzduch sú infraštruktúrou výroby. Na fungovanie tejto infraštruktúry sú potrebné čerpadlá, turbíny, kompresory atď., ktoré spotrebujú veľké množstvo elektrickej energie a aj samotného média – vody. Preto je dôležité mať spotrebu infraštruktúry pod kontrolou.

Napríklad AVEVA (pred tým Wonderware) ponúka svoje riešenie InSight ktoré zahŕňa rôzne náhľady na dáta kvázi v reálnom čase, plánovanie servisných prác atď. AVEVA InSight je vybraný iba ako príklad, keďže AVEVA (predtým Schneider Electric, pred tým Wonderware) patrí medzi svetových lídrov v oblasti automatizácie, takže technologicky aj konkurenčné riešenia sú založené na veľmi podobných technológiách (napr. miesto Microsoft Azure môže byť použitý Amazon aws).



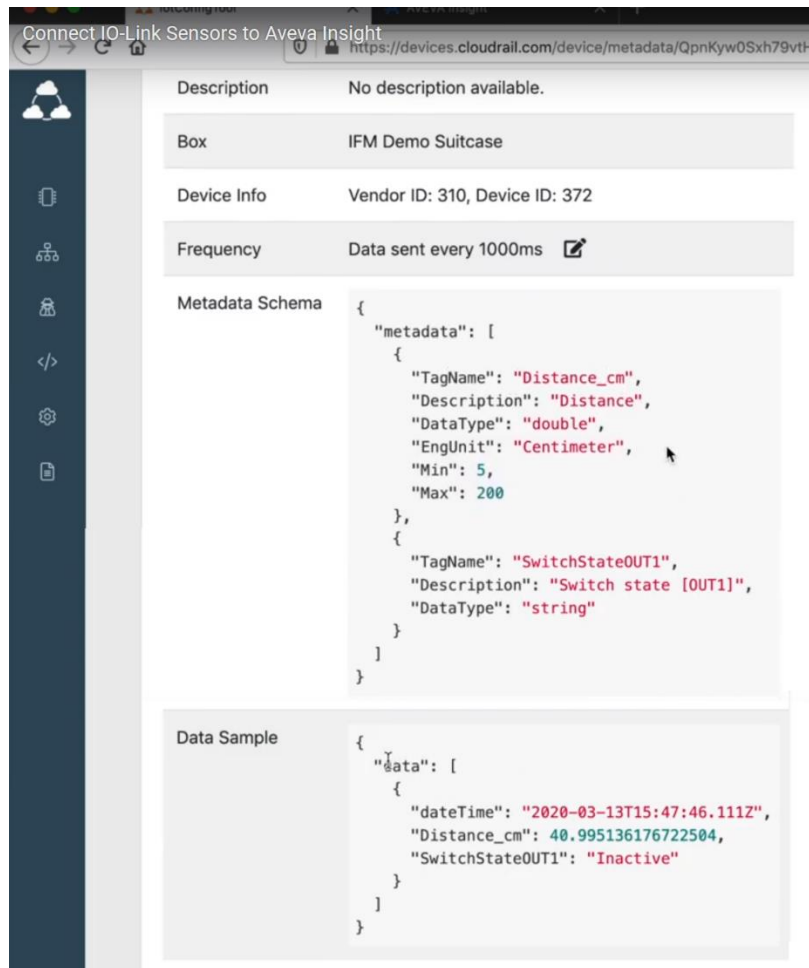
Obrázok 1 - AVEVA InSight

Zdrojom dát môže byť OPC server, AVEVA Historian (databáza pre procesné dáta), .json alebo .csv.



Obrázok 2 – json objekt ako zdroj dát

Na obrázku vyššie je ako zdroj dát json objekt cez *https*. Tu je vidno, ako sa zostavuje *payload* ktorý obsahuje dáta name, id, timestamp, data (Manditereza K., 2019).



Obrázok 3 - IO-Link ako zdroj dát

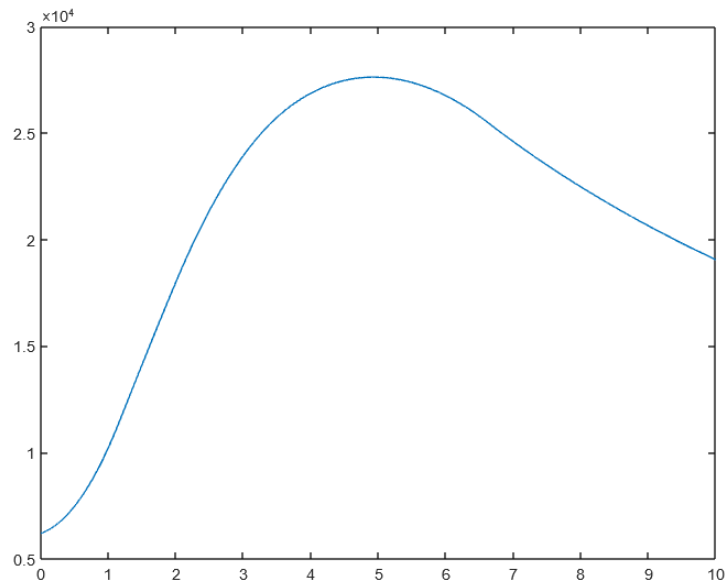
Obrázok vyššie zas zobrazuje ako zdroj dát IO-Link. IO-Link je priemyselný komunikačný štandard IEC 61131-9. Na rozdiel od predošlého zdroja dát, IO-Link posiela aj tzv. *meta dáta*, ktoré poskytujú rozšírené informácie (dátový typ, inžiniersku jednotku, rozsah) o posielanom údaji – vzdialenosti (CloudRail, 2020).

	A	B	C	D	E	F	G	H	I	J
1	ns1:tagName	ns1:description	ns1:tagType	ns1:engineeringUnit	ns1:minEU	ns1:maxEU	ns1:minRaw	ns1:maxRaw	ns1:messageOff	ns1:messageOn
2	Line2.Units	Accumulated units produced counter	uint2	Units	0	65535				
3	Reactor3.Level	Level in reactor on line 3	float	liters	0	100				
4	Reactor3.Level.SP	Setpoint for level in reactor on line 3	float	liters	0	100				
5	Temperature	Ambient temperature	float	DegC	-20	40	0	20		
6	Reactor3.OutletValve	Position of outlet valve on Reactor 3	boolean		0	1			Closed	Open
7	Reactor3.Step	Recipe step in Reactor 3	int2	Step	-2	10				
8	Line2.Product	Product code for packaging line 2	string		0	0				
9	Evaporator.Temp	Temperature in evaporator	float	DegC	0	250				
10	PLC1.Status	Connection status for PLC #4 (S7)	boolean		0	1			Offline	Online
11										
12										
13										
14										
15										
16										
17										
18										
19										

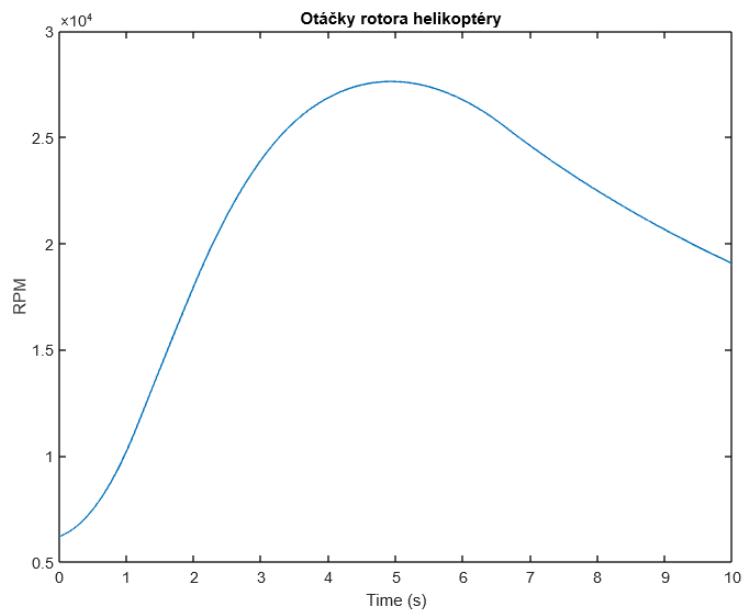
Obrázok 4 - OPC server ako zdroj dát

Asi najbežnejším zdrojom dát je OPC server. Konfigurácia zdroja dát sa deje pomocou tzv. *tag-listov*. Tag-list tiež môže a spravidla obsahuje dodatočné informácie (popis, dátový typ, inžiniersku jednotku, rozsah).

Na ukážku tu máme vykreslené dva priebehy:



Obrázok 5 - Dáta bez dodatočných informácií



Obrázok 6 - Dáta s dodatočnými informáciami

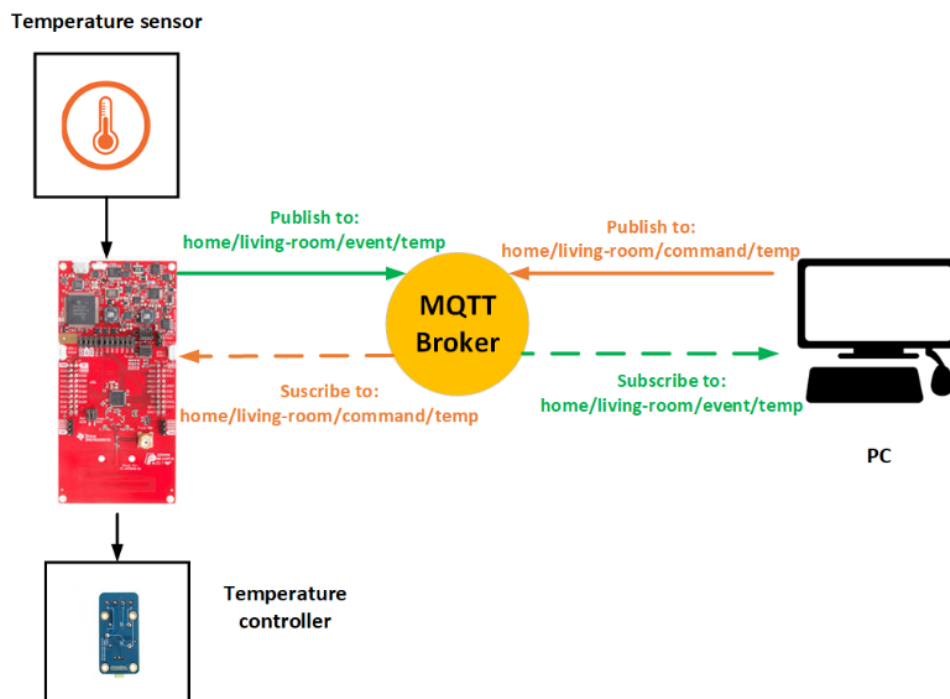
Je zrejmé, že z prvej sady dát nie je možné spraviť žiadnu analýzu, lebo nie je jasné čo je na osi  $x$  a  $y$  a čoho sa dáta týkajú. Takéto dáta sú v podstate bezcenné. Z toho vyplýva, že význam dát je rovnako dôležitý ako dáta samotné.

### 3. Sémantická interoperabilita v priemysle

Na úvod si vysvetlíme pojem *interoperabilita*. Interoperabilita je schopnosť rôznych systémov vzájomne spolupracovať (z a.j. inter-operate); schopnosť dvoch alebo viacerých systémov vymieňať si medzi sebou informácie a využívať tieto informácie (IGI Global, 2020). *Sémantická*, alebo významová interoperabilita v komunikácii znamená schopnosť spoločne komunikovať na úrovni komunikačného protokolu ako aj na úrovni porozumenia vymenených dát.

Pre porovnanie, povedzme že vďaka interoperabilite na úrovni komunikačného protokolu vieme získavať dáta ako sú zobrazené na *Obrázok 5 - Dáta bez dodatočných informácií* a pomocou sémantickej interoperability vieme získať dáta ako na *Obrázok 6 - Dáta s dodatočnými informáciami*.

Interoperabilita je horúcou témou aj v momentálne najrýchlejšie sa rozvíjajúcom odvetví priemyslu – priemyselného internetu vecí. Interoperabilita na úrovni komunikačného protokolu môže byť zabezpečená napríklad pomocou protokolu Message Queuing Telemetry Transport (MQTT), štandard ISO/IEC 20922:2016. Tento protokol je určený na komunikáciu stroj – stroj z a.j. Machine to Machine (M2M) a IoT. Jednou z hlavných myšlienok IIoT je možnosť pripojenia nejakého zariadenia, napríklad snímača, na jeho určené miesto a následné pripojenie do siete internetu, alebo intranetu. Zariadenie by malo byť automaticky schopné poselať do siete informácie – *data publishing*. Konzument dát sa môže prihlásiť na odber – *data subscribing*. Prihlásenie na odber sa deje pomocou reťazca – *topic* (Beňo, 2019, Bielik et al, 2017).



Obrázok 7 - MQTT –obojsmerná komunikácia (Beňo, 2019)

Samotné dáta sa v MQTT posielajú cez *payload*, ktorý z pravidla býva vo formáte JSON:

```
ny-power/computed/co2 {"units": "g / kWh", "value": 152.9486, "ts": "05/09/2018 00:05:00"}
```

Obrázok 8 - Príklad dát v JSON (Dague, 2018)

K dátam sa dá prístupíť priamo z web stránky napríklad takto:

```
// called when the client connects
function onConnect() {
  // Once a connection has been made, make a subscription and send a message
  console.log("onConnect");
  client.subscribe("ny-power/computed/co2");
  client.subscribe("ny-power/archive/co2/24h");
  client.subscribe("ny-power/upstream/fuel-mix/#");
}

// called when a message arrives
function onMessageArrived(message) {
  console.log("onMessageArrived:" + message.destinationName + message.payload);
  if (message.destinationName === "ny-power/computed/co2") {
    var data = JSON.parse(message.payloadString);
    $("#co2-per-kwh").html(Math.round(data.value));
    $("#co2-units").html(data.units);
    $("#co2-updated").html(data.ts);
  }
  if (message.destinationName.startsWith("ny-power/upstream/fuel-mix")) {
    fuel_mix_graph(message);
  }
  if (message.destinationName === "ny-power/archive/co2/24h") {
    var data = JSON.parse(message.payloadString);
    var plot = [
      {
        x: data.ts,
        y: data.values,
        type: 'scatter'
      }
    ];
    var layout = {
      yaxis: {
        title: "g CO2 / kWh",
      }
    };
    Plotly.newPlot('co2_graph', plot, layout);
  }
}
```

Obrázok 9 – Prístup k JSON dátam (Dague, 2018)

Takýto prístup k dátam nie je práve najšťastnejší, lebo prístup k dátam je natvrdo naprogramovaný v aplikácii a tým pádom má aplikácia využitie iba len pre jeden daný prípad, kedy sú dáta v špecifickom formáte ako je ukázané na *Obrázok 8 - Príklad dát v JSON (Dague, 2018)*.

Pozrime sa ešte raz na *Obrázok 3 - IO-Link ako zdroj dát*. Ak by náš JSON reťazec obsahoval dáta a aj meta-dáta a my by sme vedeli že sú v štandardizovanom formáte *IO-Link*, tak by sa k dátam dalo pristupovať generickým spôsobom, čo by jednoznačne zjednodušilo ich použitie.



Takže je zrejmé, že aj v komunikácii IIoT je nevyhnuté to aby konzument dát (*subscriber*) mal vedomosť o štruktúre prijímaných dát, teda aby rozumel aj významu (*sémantike*) prijímaných dát.

#### 4. Sémantická interoperabilita v OPC Unified Architecture

OPC UA je medzinárodným štandardom pre horizontálnu a vertikálnu komunikáciu v priemyselnej výrobe a automatizácii, poskytuje sémantickú (významovú) interoperabilitu vo svete vzájomne spojených systémov. Predstavuje základ pre konektivitu internetu vecí a pre Priemysel 4.0 (Gerrikagoitia et al, 2019).

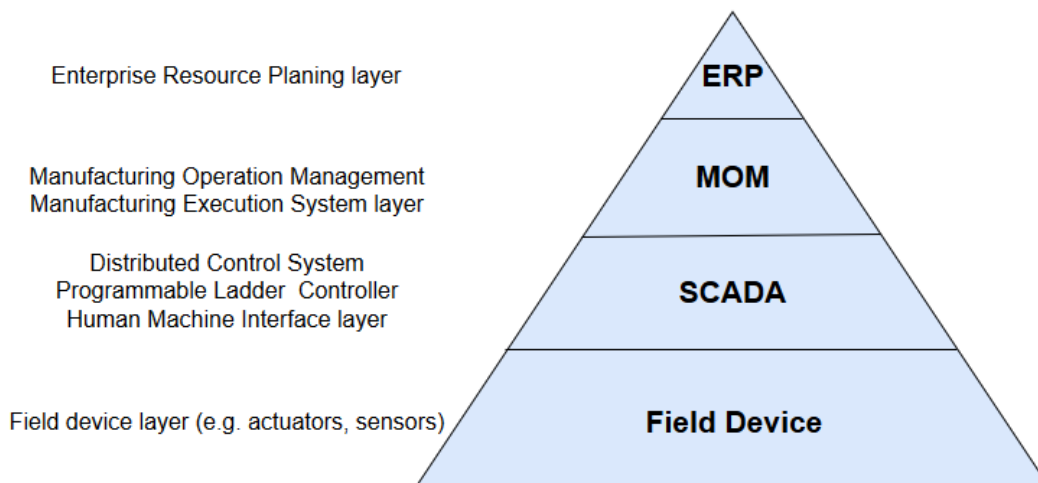
IEC Štandard	OPC UA špecifikácia
IEC TR 62541-1:2016	Part 1: Overview and concepts
IEC TR 62541-2:2016	Part 2: Security Model
IEC TR 62541-3:2015	Part 3: Address Space Model
IEC TR 62541-4:2015	Part 4: Services
IEC TR 62541-5:2015	Part 5: Information Model
IEC TR 62541-6:2015	Part 6: Mappings
IEC TR 62541-7:2015	Part 7: Profiles
IEC TR 62541-8:2015	Part 8: Data Access
IEC TR 62541-9:2015	Part 9: Alarms and conditions
IEC TR 62541-10:2015	Part 10: Programs
IEC TR 62541-11:2015	Part 11: Historical Access
(nie je súčasťou IEC)	Part 12: Discovery and Global Services
IEC TR 62541-13:2015	Part 13: Aggregates
(nie je súčasťou IEC)	Part 14: PubSub
IEC TR 62541-100:2015	Part 100: Device Interface

Tabuľka 1 IEC62541 a OPC UA špecifikácie

Aby bola dosiahnutá sémantická interoperabilita, prenesené dáta musia obsahovať ucelenú informáciu o informačnom modeli, ktorý tieto dáta reprezentujú. Koncept reprezentácie informačného modelu OPC UA je založený na objektovo orientovanom programovaní (OOP), čo je programovacia paradigma (súbor princípov, na ktorých je založený programovací jazyk), ktorá sa skladá z objektov, z väzieb medzi objektami, z vlastností i metód objektov, a aj z udalostí. V súlade s konceptom OOP, informačný model OPC UA zahŕňa mechanizmy ako abstrakcia, zapuzdrenie, dedičnosť a polymorfizmus. Aby bola zabezpečená interoperabilita medzi stranami ktoré spolu komunikujú, je potrebné oddeliť prenášanú informáciu od technológie ktorá ju transportuje (reprezentácia informácia napr. po kábli, komunikačný protokol) (Lange et al, 2010, pp. 100-101). *OPC UA mapping* definuje kódovanie dát nezávisle od technológie komunikačného protokolu.

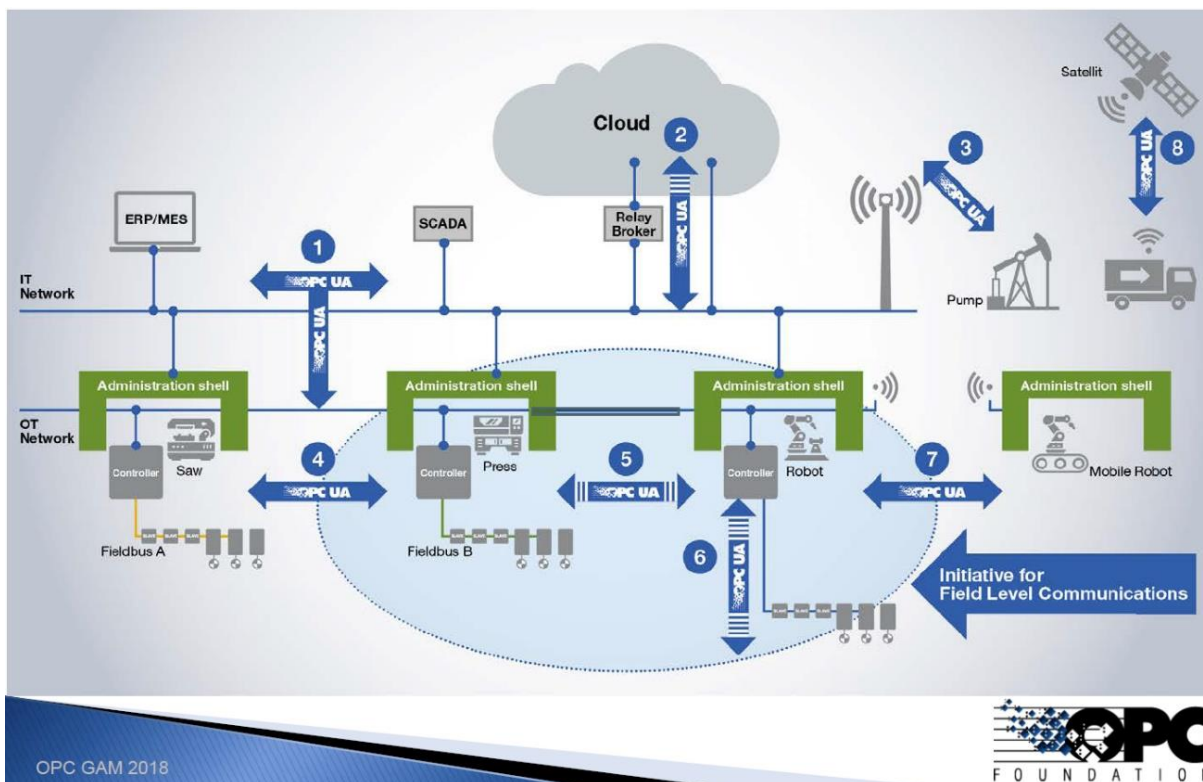
Interoperabilita robí z OPC UA platformovo nezávislý komunikačný protokol a vďaka svojmu flexibilnému informačnému modelu je dostatočne škálovateľný aby mohol byť implementovaný v rôznych zariadeniach, od vstavaných (embedded) systémov, cez zariadenia v prevádzke, riadiace systémy, PLC, brány (gateways), alebo operátorské panely. Taktiež čo sa týka operačného systému, OPC UA podporuje Unix, Windows, ale aj RTOS (Real Time Operating System).

Kaskádové zapojenie OPC UA klientov a serverov počnúc výrobnou halou až po plánovanie produkcie, ponúka automatizácii nové možnosti vo vertikálnej integrácii. Stratégia transformácie tradičného konceptu automatizácie (automatizačná pyramída) Obrázok 1. na Priemysel 4.0 nevyhnutne neznamená nahradenie OPC Classic za OPC UA. Obidve technológie vedia spolu koexistovať, čo dáva príležitosť k postupnej migrácii výroby na Priemysel 4.0.



Obrázok 10 – Automatizačná pyramída

I keď sa zdá nevyhnutným, že hierarchia automatizačnej pyramídy bude porušená vertikálnou integráciou.



Obrázok 11 – OPC UA Unified Architecture - od senzora do cloudu (opcfoundation.org, 2019)

Komunikačné rozhranie		Príklad
1	Komunikácia informačnej a operačnej technológie (Information / Operation Technology).	Komunikácia PLC a MES.
2	Integrácia do clodu.	Ukladanie veľkých dát.
3	Zabezpečený vzdialený prístup.	Prístup do vnútornej siete z detašovaného pracoviska.
4	Lokálna komunikácia operačnej technológie.	Komunikácia rôznych riadiacich systémov napr. DCS a PLC.
5	Riadiaci prístroj a riadiaci prístroj.	Komunikácia medzi PLC.
6	Riadiaci prístroj a zariadenie vo výrobe.	Komunikácia medzi PLC a pumpou, senzorom.
7	Bezdrôtová integrácia (5G).	Komunikácia s laserom navádzaným vozidlom v sklade.
8	Príprava na budúcnosť.	Na budúce rozšírenia.

Tabuľka 2 Popis komunikačných rozhraní v Priemysle 4.0 s OPC UA

#### 4.1. Adresný priestor (Address space)

Adresný priestor je neoddeliteľnou súčasťou OPC UA serveru. Zjednodušenie, OPC UA server môže poskytnúť klientovi iba tie dáta, ktoré sa nachádzajú v jeho adresnom priestore. Ako už napovedá samotný názov, každý jeden element tohto priestoru je adresovateľný, a to jednoznačne. Detailné informácie o koncepte a modelovaní v adresnom priestore poskytuje špecifikácia OPC UA Part 3: Address Space Model (OPC Foundation, 2017a).

OPC UA Address Space Model je definovaný pomocou *Base Node Class* (základná trieda uzlov) od ktorých sú odvodené všetky ostatné triedy uzlov v súlade s OOP.

V texte sú často použité grafické anotácie OPC UA dát, ktoré sú popísané v OPC UA špecifikácii (OPC Foundation, 2017a).

NodeClass	Graphical Representation	ReferenceType	Graphical Representation
Object		Any symmetric ReferenceType	
ObjectType		Any asymmetric ReferenceType	
Variable		Any hierarchical ReferenceType	
VariableType		HasComponent	
DataType		HasProperty	
ReferenceType		HasTypeDefinition	
Method		HasSubtype	
View		HasEventSource	

Obrázok 12 – Grafické anotácie uzlov referencií

Ako už bolo spomenuté, OPC UA adresný priestor je navrhnutý na základe OOP pričom samotný adresný priestor sa skladá z tzv. menných priestorov (*name spaces*). Menný priestor sa používa predovšetkým v programovacích jazykoch, v ktorých sa rovnaký názov môže používať pre rôzne

objekty. Je vytvorený na zoskupenie tých mien, ktoré by sa mohli opakovať kdekoľvek inde v rámci rovnakých alebo vzájomne prepojených programov, objektov a prvkov. Využitím menných priestorov je zabezpečená jednoznačná adresovateľnosť v rámci celého adresného priestoru. Menný priestor je v podstate unikátny identifikátor informačného modelu, ktorý definuje typy a objekty, premenné z nich vytvorené a ostatné uzly (viď. obrázok vyššie). Štandardizované informačné modely sú v OPC Foundation nazývané *companion specifications*.

#### 4.2. Informačné modely (Information models)

Informačný model sa môže skladať z jedného alebo viacerých menných priestorov a je súčasťou adresného priestoru. Základný menný priestor označovaný ako *ns0* je rodičom všetkých menných priestorov a je obsiahnutý v každom informačnom modeli. Ďalším informačným modelom, ktorý bol povýšený na špecifikáciu OPC UA Part 100, je *Device Information Model* (OPC Foundation, 2019). Z neho sú odvodené informačné modely, ktoré popisujú rôzne zariadenia, napríklad aj PLC.

Organizácia OPC Foundation v spolupráci s jej členmi (komerčnými subjektmi alebo organizáciami) vytvorila tzv. *companion specifications*, spoločné špecifikácie, ktorými sa štandardizuje opis vecí reálneho sveta. Jednotné modelovanie umožňuje v komunikácii jednoduchšiu sémantickú interoperabilitu a vďaka nemu sú modely transparentné pre klienta aj server a to aj naprieč rôznymi platformami a výrobcami. Tieto informačné modely sú voľne dostupné a na ich základe je možné vytvoriť aj špecifický (užívateľský) informačný model. Takýto špecifický model dedí atribúty informačných modelov z ktorých je odvodený. OPC UA špecifikácia Part 5 (OPC Foundation, 2017b) popisuje koncept informačného modelu OPC UA a štandardné uzly, z ktorých je informačný model vytvorený. Pre lepšiu predstavu, nižšie je zoznam vybraných informačných modelov. Momentálne je ich 21.

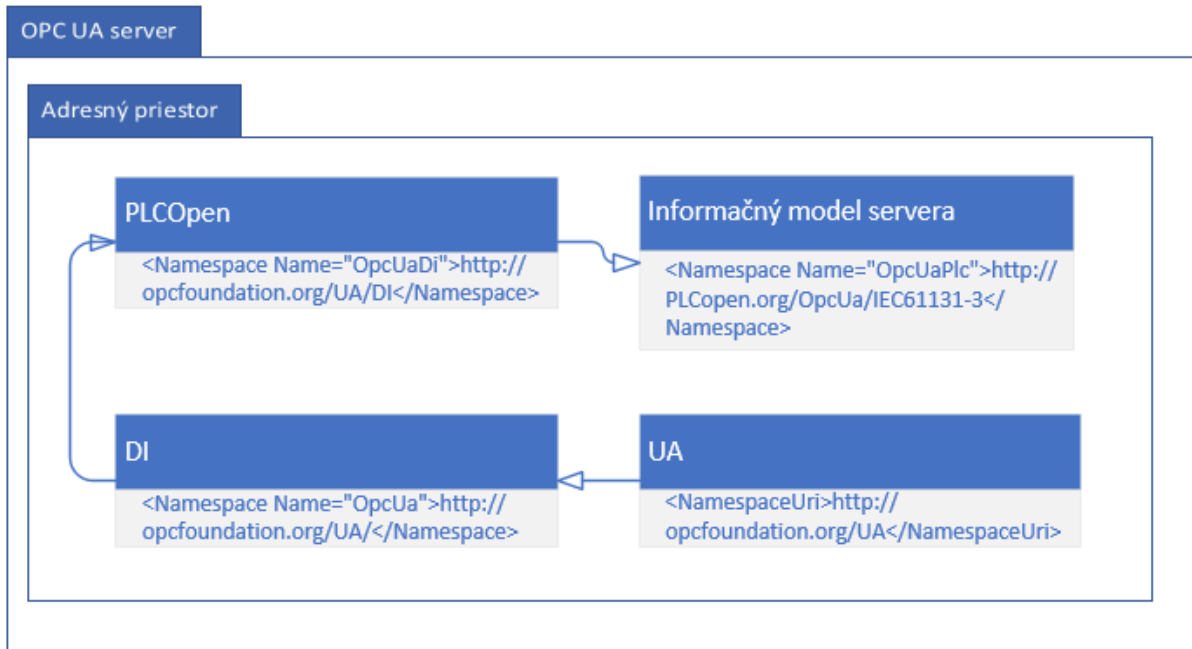
- OPC Unified Architecture for PROFINET
- OPC UA for IO Link Devices and IO-Link Masters
- OPC Unified Architecture for FDI
- OPC UA for PLCopen
- OPC Unified Architecture for CNC Systems
- OPC Unified Architecture for AutoID
- OPC Unified Architecture for MTConnect
- OPC Unified Architecture for AutomationML
- ISA-95 Common Object Model

#### Príklad:

Obrázok nižšie opisuje prípad OPC UA servera. Tento server má modelovať PLC objekt, presnejšie inštanciu PLC objektu, čiže fyzické zariadenie. Takýto model sa musí nachádzať v adresnom priestore servera. Všetky špecifické uzly OPC UA servera sú definované v „*Informačnom modeli servera*“. Prototyp (*ObjectType*) inštancie PLC objektu je definovaný v informačnom modeli *PLCOpen*. V rámci „*Informačného modelu servera*“ pristupujeme k definíciám z *PLCOpen* cez menný priestor *OpcUaPlc*.

Analogicky, prototypy uzlov (objektov, premenných, atď.) informačného modelu *PLCOpen* sú definované v informačnom modeli *UA*. K týmto uzlom pristupujeme cez menný priestor *OpcUaDi*. Podobne sa dostaneme až k základnému informačnému modelu *UA*. „*Informačný model servera*“ vie adresovať akýkoľvek uzol z adresného priestoru pomocou príslušného menného priestoru (*OPCUaPlc*, *OpcUaDi*, *OpcUa*) a názvu objektu. Menné priestory sú v adresnom priestore indexované číslami. Kde

index 0 prináleží práve základnému informačnému modelu UA – preto *ns0* je základným menným priestorom adresného priestoru.



Obrázok 13 – Zobrazenie informačného modelu v adresnom priestore

### 4.3. Sady servisov (Service sets)

V rámci OPC UA sa funkcionality zoskupujú do sád servisov. Týmto sadám sa venuje OPC UA špecifikácia Part 4 (OPC Foundation, 2017c). Táto kapitola sa venuje týmto sadám iba informatívne:

- Discovery Services Set
- Secure Chanel Service Set
- Session Service Set
- Node Management Service Set
- View Service Set
- Query Service Set
- Attribute Service Set
- Method Service Set
- Monitored Item Service Set
- Subscribtion Service Set
- 

#### **Discovery Service Set**

Dôležité je podotknúť, že na jednom hostiteľovi (počítači) môže bežať viac serverov. Ďalej je treba zdefinovať tzv. *discovery server* – server ktorý „objavuje“ ostatné servery. Server v rámci svojho hostiteľa môže použiť servis *RegisterServer* voči *discovery serveru*. To umožní jeho objavenie klientom, ktorý zavolajú servis *FindServers*. Podobne fungujú aj servisy *RegisterServer2* a *FindServersOnNetwork*, ale v rámci siete. Servis *GetEndpoints* zobrazí klientovi tzv. koncové body (*endpoints*) s ich bezpečnostnými nastaveniami. Koncové body poskytuje server a slúžia klientovi k pripojeniu k serveru.

#### **Secure Chanel Service Set**

Na rozdiel od ostatných servisov, servisy v tejto sade nie sú implementované v rámci aplikácie OPC UA, ale v tzv. komunikačnom stohu (*communication stack*). Sada obsahuje servisy ako *OpenSecureChannel* a *CloseSecureChannel*. Pred komunikáciou klient – server je nevyhnutné otvorenie zabezpečeného kanálu.

#### **Session Service Set**

Táto sada v sebe zahŕňa servisy ktoré spravujú vytvorenie komunikácie medzi klientom a serverom: *CreateSession*, *ActivateSession*, *CloseSession* a *CancelSession*.

#### **Node Management Service Set**

Servisy tejto sady slúžia na spravovanie uzlov adresného priestoru servera. Sú to *AddNodes*, *AddReferences*, *DeleteNodes* a *DeleteReferences*.

#### **View Service Set**

Táto sada v sebe zahŕňa servisy, ktoré umožňujú tzv. prechádzanie (*browsing*) adresného priestoru servera alebo jeho časti náhľadu (*view*). Sú to servisy *Browse* a *BrowseNext*. Servis *TranslateBrowsePathsToNodeIds* môže byť použitý na preloženie cesty, napríklad získanej zo stromu uzlov, na identifikátor uzla. Dvojica servisov *RegisterNodes* a *UnregisterNodes* slúži na optimalizovaný

zápis / čítanie uzlov. Ak je napríklad potrebné často zapisovať alebo čítať atribúty uzla, tak sa klient na tento uzol prihlási (*RegisterNodes*) a po ukončení odhlási (*UnregisterNodes*).

### **Query Service Set**

Pomocou tejto sady servisov je možné dopytovať sa na uzly adresného priestoru, prípadne podmnožiny adresného priestoru – pohľadu (*view*). Napríklad pomocou dotazu (*query*) je možné získať všetky inštancie daného prototypu. Servisy tejto sady sú *QueryFirst* a *QueryNext*.

### **Attribute Service Set**

Pomocou tejto sady servisov je možné pristupovať k atribútom uzlov, čiže k jeho hodnotám. Podporované servisy sú *Read*, *Write*, *HistoryRead* a *HistoryUpdate*.

### **Method Service Set**

Táto sada obsahuje servis *Call*, pomocou ktorého je možné zavolať metódu.

### **Monitored Item Service Set**

Klient používa servisy tejto sady na definovanie monitorovaného elementu. Monitorovaným elementom môže byť akýkoľvek atribút uzla. Klient následne odoberá (*subscribe*) jeho dáta a udalosti. Sada obsahuje servisy: *CreateMonitoredItems*, *ModifyMonitoredItems*, *SetMonitoringMode*, *SetTriggering* a *DeleteMonitoredItems*.

### **Subscription Service Set**

Sada odoberania (*subscription*) popisuje servisy spravujúce odoberanie notifikácií monitorovaných elementov. Vzťah medzi monitorovaným elementom a odoberateľom je nasledovný: klient vytvorí monitorovaný element, ktorý monitoruje atribút uzla v adresnom priestore servera. Tento monitorovaný element posiela notifikácie, ktoré klient odoberá cez servisy sady *subscription*. Sada *subscription* obsahuje tieto servisy: *CreateSubscription*, *ModifySubscription*, *SetPublishingMode*, *Publish*, *Republish*, *TransferSubscriptions* and *DeleteSubscriptions*.

#### 4.4. Profily

Servisy a informačné modely klientov a serverov možno považovať sa ich vlastnosti, črty (*features*). Tieto vlastnosti (*features*) sú v rámci OPC UA zoskupené do tzv. *ConformanceUnits* a tie sa zas združujú do profilov (*profiles*). Špeciálnym druhom profilu je tzv. *facet*, ktorý združuje servisy iba pre určitú, čiastkovú funkcionálnu. Napríklad Core 2017 Server Facet zoskupuje niektoré funkcie poskytované servisnými sadami *Discovery Services, Session Services, View Services*, atď.

Detailne sú tieto pojmy a ich vzťahy popísané v špecifikácii Part 7 (OPC Foundation, 2017d). Táto špecifikácia sa venuje profilom a *conformance units* hlavne z pohľadu testovania a získania certifikácie pre komerčné použitie klientov a serverov. Stránka OPC UA Profile (OPC Foundation, 2017e) popisuje profily viac z hľadiska škálovateľnosti:

Pre servery sú k dispozícii *plnohodnotné* profily:

- Nano Embedded Device 2017 Server Profile
- Micro Embedded Device 2017 Server Profile
- Embedded 2017 UA Server Profile
- Standard 2017 UA Server Profile

Podobne aj pre klienty:

- Minimum UA Client Profile
- Standard UA Client 2017 Profile



## 5. Praktická implementácia OPC Unified Architecture servera a klienta

Pre priblíženie čitateľovi OPC UA, táto kapitola obsahuje prehľad implementácií OPC UA servera, klienta pomocou voľných a otvorených vývojových softvérových sád.

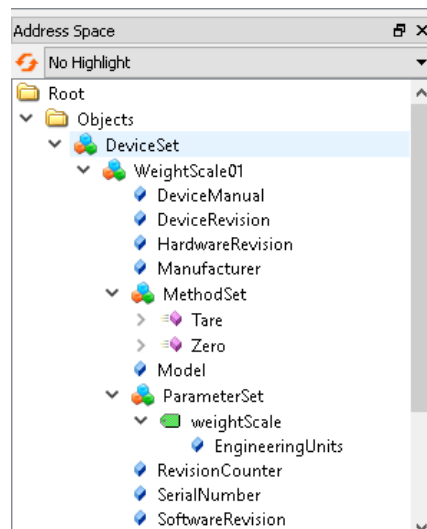
Asi najprirodzenejším spôsobom ako sa zoznámiť s OPC UA serverom a klientom z programátorského hľadiska je pomocou softvérovej vývojovej sady od OPC Foundation UA-.NETStandard.

### **Základny OPC UA server klient**

Postup vytvorenia OPC UA servera a klienta od OPC Foundation je možné nájsť tu [http://opcfoundation.github.io/UA-.NETStandard/help/server\\_development.htm](http://opcfoundation.github.io/UA-.NETStandard/help/server_development.htm), prípadne tu [OPC Unified Architecture a jazyk C#](#) (Hók, 2017).

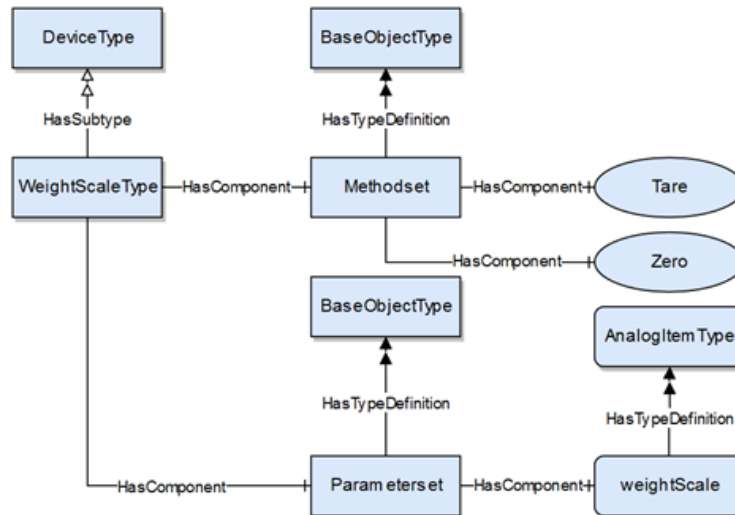
### **OPC UA Server s vlastným informačným modelom**

Pokročilejší príklad OPC UA servera s vytvorením vlastného informačného modelu pre váhu je možné nájsť tu [STUBA-rupr / UA-.NETStandard WeightScaleServer](#) (Pribiš, 2019b).



Obrázok 14 – Objekty OPC UA servera zobrazené v UA Expert (Pribiš, 2019b)

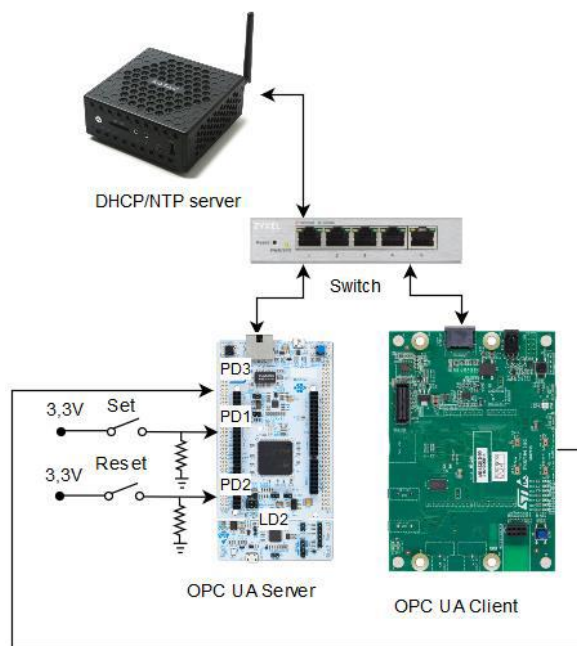
Podrobný popis vytvorenia OPC UA informačného modelu sa nachádza tu [STUBA-rupr /DI-InformationModel](#) (Pribiš 2019c).



Obrázok 15 –Grafická anotácia OPC UA informačného modelu pre váhu (Pribiš, 2019b)

### OPC UA server klient na vloženom (z a.j. embedded) zariadení

V úvode textu sú spomínané IoT a IIoT zariadenia ktoré dokážu poskytovať dáta do siete internetu alebo intranetu. Takéto zariadenie je možné vytvoriť aj pomocou OPC UA. V tomto konkrétnom prípade sa jedná o typ komunikácie server – klient, nie publisher – subscriber ako je spomínané vyššie. Pre túto implementáciu bol zvolená voľná otvorená softvérová vývojová sada open62541 (Pribiš et al, 2019d).



Obrázok 16 –Príklad použitia OPC UA servera a klienta na vloženom zariadení (Pribiš et al, 2019d)

## 6. Záver

Tento text ozrejmjuje potrebu sémantickej interoperability, ktorá má umožňovať otvorenú komunikáciu a najmä porozumenie vymenených dát medzi rôznorodými zariadeniami internetu a priemyselného internetu vecí. Predstavuje OPC Unified Architecture, ako *informačný model* i ako *komunikačný protokol*, ktorým je možné vytvárať informačné modely takmer pre všetky oblasti reálneho či digitálneho sveta a to v otvorenej a štandardizovanej podobe.

## Zoznam literatúry

- Manditereza K., 2019, *How To Collect and Analyse Industrial Data Using Node-Red & Aveva Insight*, získané 16. novembra 2020 z [https://www.youtube.com/watch?v=E4e4Nzlc\\_p4](https://www.youtube.com/watch?v=E4e4Nzlc_p4)
- CloudRail, 2020, *Connect IO-Link Sensors to Aveva Insight*, získané 16. novembra 2020 z <https://www.youtube.com/watch?v=mCZIDkWsxPo>
- AVEVA, 2020, *AVEVA Insight | BI Gateway*, získané 16. novembra 2020 z <https://www.youtube.com/watch?v=xY95cu0qsE0>
- IGI Global, 2020, *What is Interoperability*, získané 16. novembra 2020 z <https://www.igi-global.com/dictionary/interoperability-medical-devices-information-systems/15494>
- L. Beňo, 2019, *Protokol MQTT a OPC UA Pub/Sub*, získané 16. novembra 2020 z <https://elearning.mechatronika.cool/blog/protokol-mqtt-a-opc-ua-pub-sub/>
- A. Bielík, M. Grega a P. Halász, 2017, *Mosquitto Telemetry Transport protokol pre IoT*, získané 16. novembra 2020 z <https://elearning.mechatronika.cool/blog/message-queuing-telemetry-transport-mqtt-protokol-pre-internet-of-things/>
- S. Dague, 2018, *Using MQTT to send and receive data for your next project*, získané 16. novembra 2020 z <https://opensource.com/article/18/6/mqtt>
- J. K. Gerrikagoitia, G. Unamuno, E. Urkia and A. Serna, 2019, *Digital Manufacturing Platforms in the Industry 4.0 from Private and Public Perspectives*, Applied Sciences
- Lange, J., Ivanitz, F. and Burke, T.J., 2010, *OPC from Data Access to Unified Architecture* 4th ed, Berlin Offenbach
- opcfoundation.org, 2019, *OPC Unified Architecture Interoperability for Industrie 4.0 and the Internet of Things*, získané 16. novembra 2020 z <https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Interoperability-For-Industrie4-and-IoT-EN.pdf>
- OPC Foundation, 2017a *OPC Unified Architecture Part 3: Address Space Model*, získané 16. novembra 2020 z <https://opcfoundation.org/developer-tools/specifications-unified-architecture>
- OPC Foundation, 2019, *OPC Unified Architecture Part 100: Devices*, získané 16. novembra 2020 z <https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-100-device-information-model/>
- OPC Foundation, 2017b, *OPC Unified Architecture Part 5: Information Model*, získané 16. novembra 2020 z <https://opcfoundation.org/developer-tools/specifications-unified-architecture>
- OPC Foundation, 2017c, *OPC Unified Architecture Part 4: Services*, získané 16. novembra 2020 z <https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-4-services/>
- OPC Foundation, 2017d, *OPC Unified Architecture Part 7: Profiles*, získané 16. novembra 2020 z <https://opcfoundation.org/developer-tools/specifications-unified-architecture/>
- OPC Foundation, 2017e, *OPC UA Profiles*, získané 16. novembra 2020 z <https://apps.opcfoundation.org/profilereporting/>
- Pribiš R., 2019a, *Návrh a tvorba OPC UA adresného priestoru*, získané 16. novembra 2020 z <https://elearning.mechatronika.cool/blog/navrh-a-tvorba-opc-ua-adresneho-priestoru/>
- Pribiš R., 2018, *OPC UA – história a špecifikácia – dočasný učebný text*, získané 16. novembra 2020 z <https://elearning.mechatronika.cool/blog/navrh-a-tvorba-opc-ua-adresneho-priestoru/>

KEGA 038STU-4/2018, 2018, *OPC UA – Interoperabilita pre priemysel 4.0 a internet vecí (IoT)*, získané 16. novembra 2020 z <https://elearning.mechatronika.cool/blog/opc-ua-interoperabilita-pre-priemysel-4-0-a-internet-veci-iot/>

Pribiš R., Beňo L. a Drahoš P., 2020, *An Industrial Communication Platform for Industry 4.0 - case study*, získané 16. novembra 2020 z <https://ieeexplore.ieee.org/document/9039873>

Hók M., 2017, *OPC UA Príklady v C#*, získané 16. novembra 2020 z <https://elearning.mechatronika.cool/blog/opc-unified-architecture-a-jazyk-c/>

Pribiš R., 2019b, *STUBA-rupr / UA-.NETStandard WeightScaleServer*, získané 16. novembra 2020 z <https://github.com/STUBA-rupr/UA-.NETStandard/blob/master/SampleApplications/Workshop/Boiler/WeightScaleServer/readme-SK.md#vytvorenie-obslu%C5%BE%C3%A9ho-k%C3%B3du-pre-met%C3%B3dy-tare-a-zero-adresn%C3%A9ho-priestoru-weightscale>

Pribiš R., 2019c, *STUBA-rupr / DI-InformationModel*, získané 16. novembra 2020 z <https://github.com/STUBA-rupr/DI-InformationModel>

Pribiš R., Beňo L. a Drahoš P., 2019, *Implementation of Micro embedded OPC Unified Architecture server-client*, získané 16. novembra 2020 z <https://www.sciencedirect.com/science/article/pii/S2405896319326916>