

OPC DAY FINLAND 2019

6.-7.11.2019 @ EXPO AND CONVENTION CENTRE MESSUKESKUS HELSINKI

#OPCUA #OPCDAY #OPCDAYFINLAND #AUTOMAATIO

OPC UA Information Modeling Overview



Dr. Wolfgang Mahnke
ascolab / Unified Automation
wolfgang.mahnke@ascolab.com

SPONSORS:



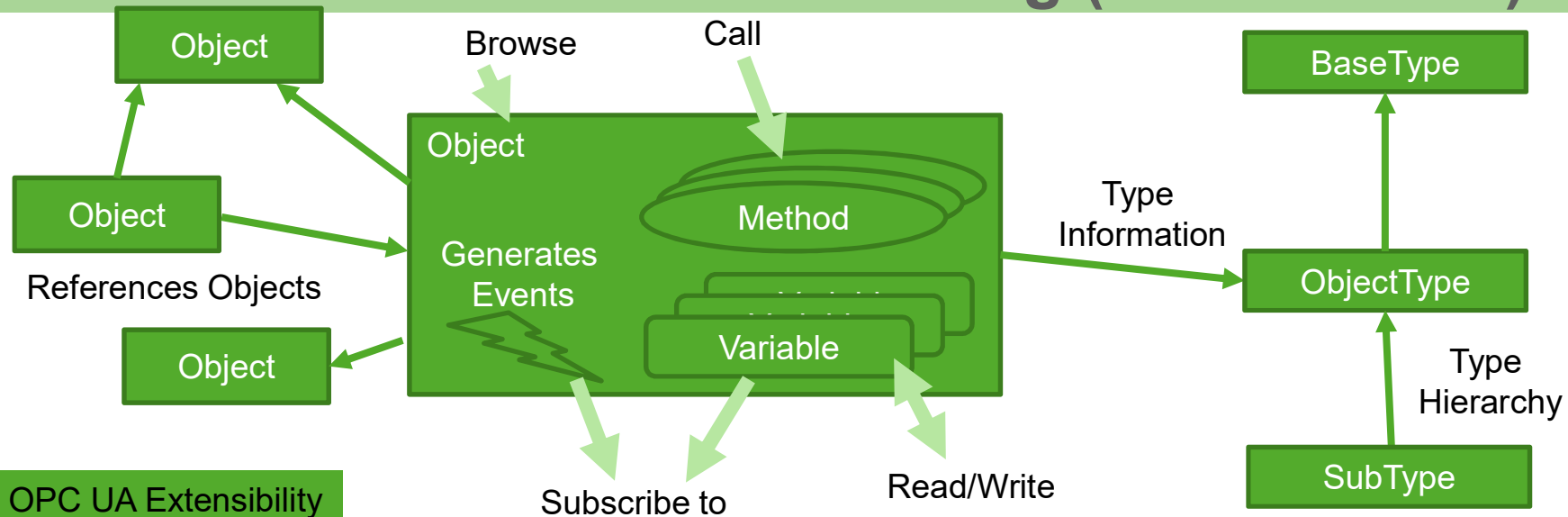
FINNISH SOCIETY OF AUTOMATION
SUOMEN AUTOMAATIOSEURA RY



BECKHOFF



OPC UA Information Modelling (in a Nutshell)



OPC UA Extensibility

ObjectType	Structure (Variables and Methods) and Semantic of Objects
VariableType	Structure (Sub-Variables and DataType) and Semantic of Variables
DataType	Structure and Semantic of DataTypes
ReferenceType	Semantic of References, ReferenceType Hierarchy
EventTypes	Structure (Event Fields) and Semantic of Events

Examples:

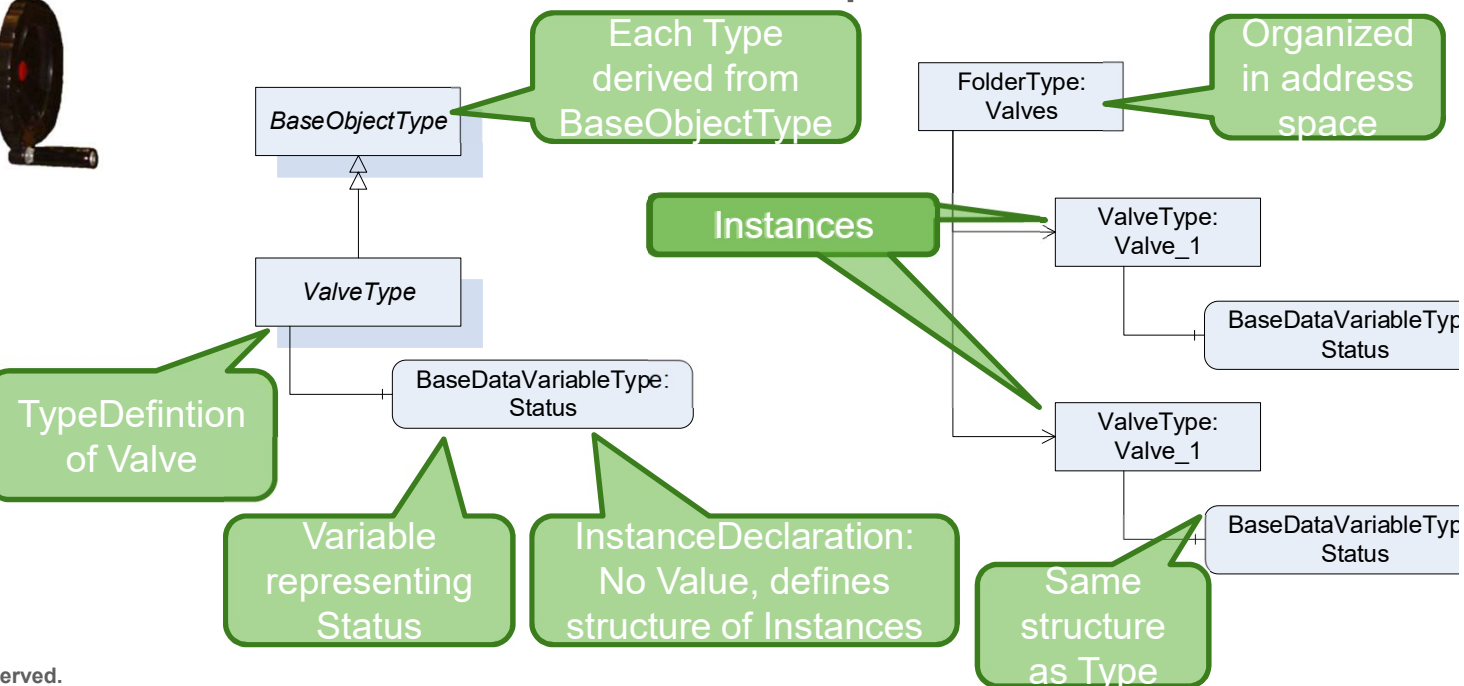
- File Transfer
- State Machines

Let's model a Device – Valve



> Base Functionality

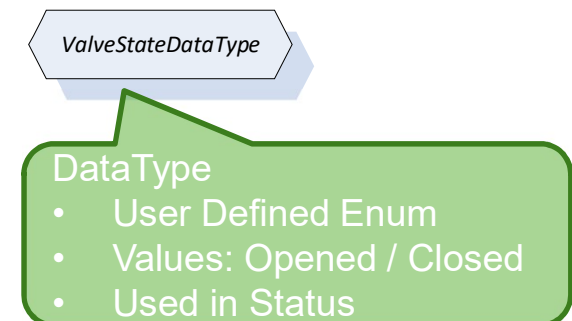
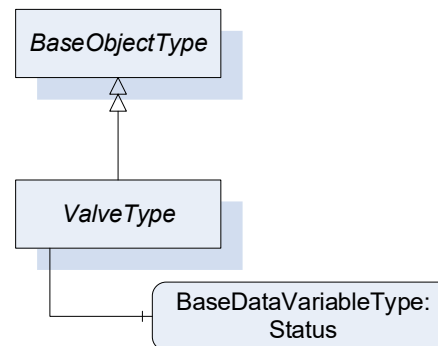
> Value can be open or closed



Let's model a Device – Valve



- > **DataType of Status?**
 - > Opened / Closed -> Boolean?
 - > Semantic Matters -> User Defined Enumeration



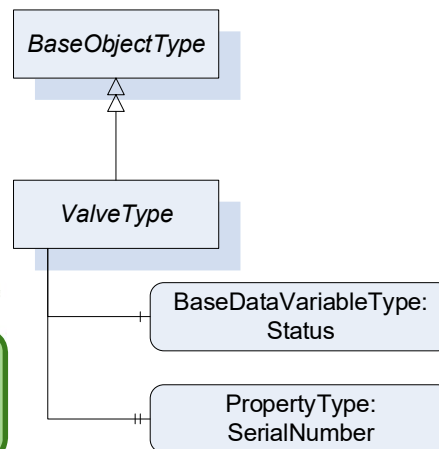
Let's model a Device – Valve



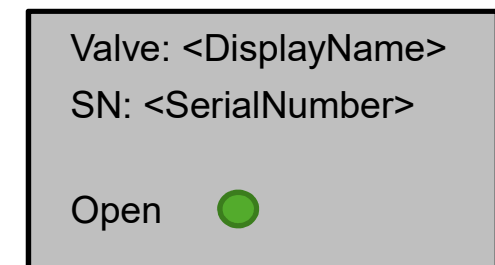
TypeDefintion
of Valve

> Identification

> Each Device has a SerialNumber



Property containing
Serial Number (String)

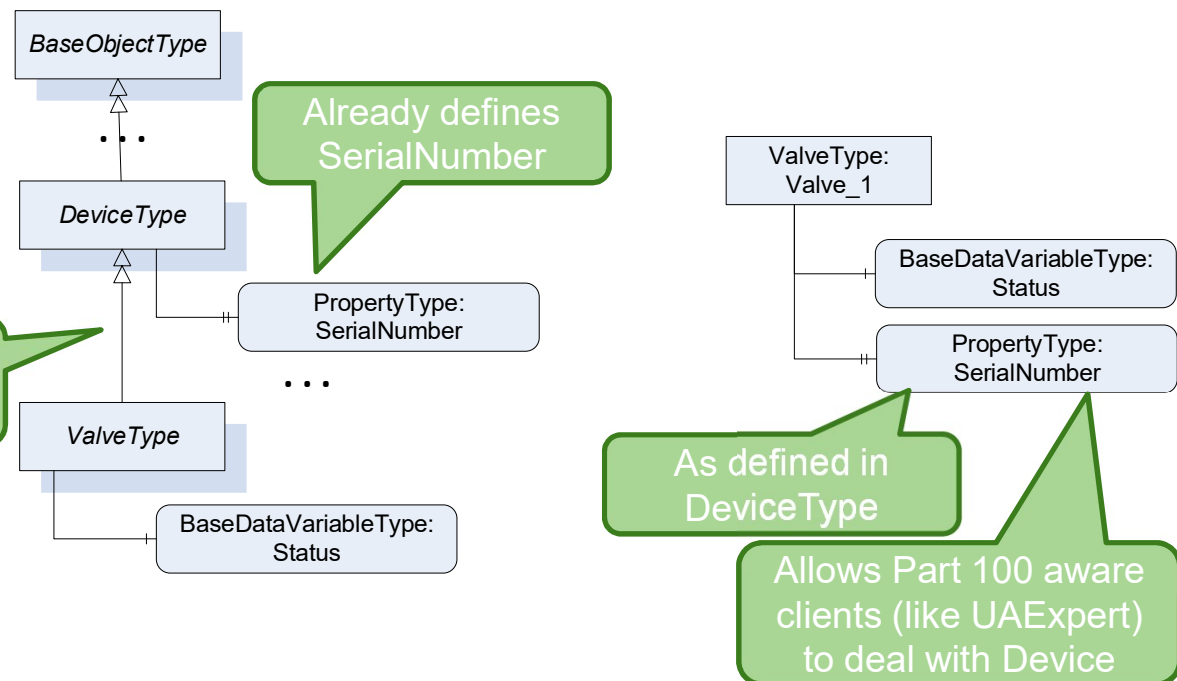


Simple UI Element
based on TypeDefintion

Let's model a Device – Valve



- > Identification – Better Approach
 - > Use Part 100 OPC UA Devices



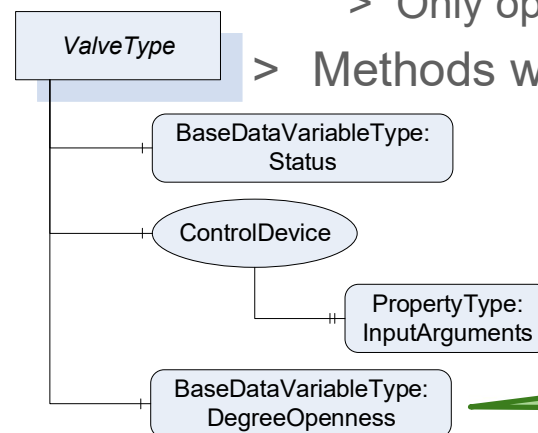
Let's model a Device – Valve



> Controlling the Device

- > Pure Opened / Closed could be realized by writing the Status Variable
- > Let's assume you want to
 - > Open the valve for a specific duration
 - > Only open it to a certain degree

> Methods with Input Arguments



```
ControlDevice (  
    [in] Byte DegreeOpenness  
    [in] Duration TimeOpened  
) ;
```

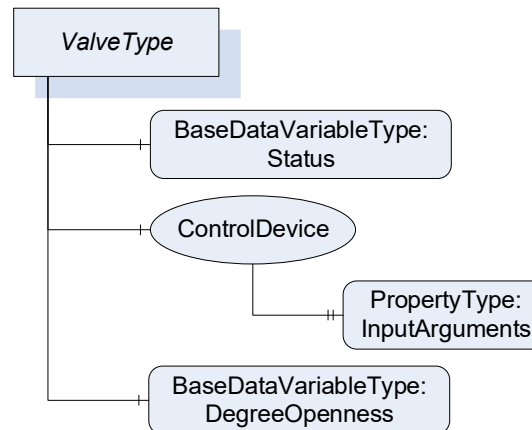
Additional
Variable

Let's model a Device – Valve



> Controlling the Device

> Updated UI



Valve: <DisplayName>
SN: <SerialNumber>

Open ☒

Degree Openness

Control

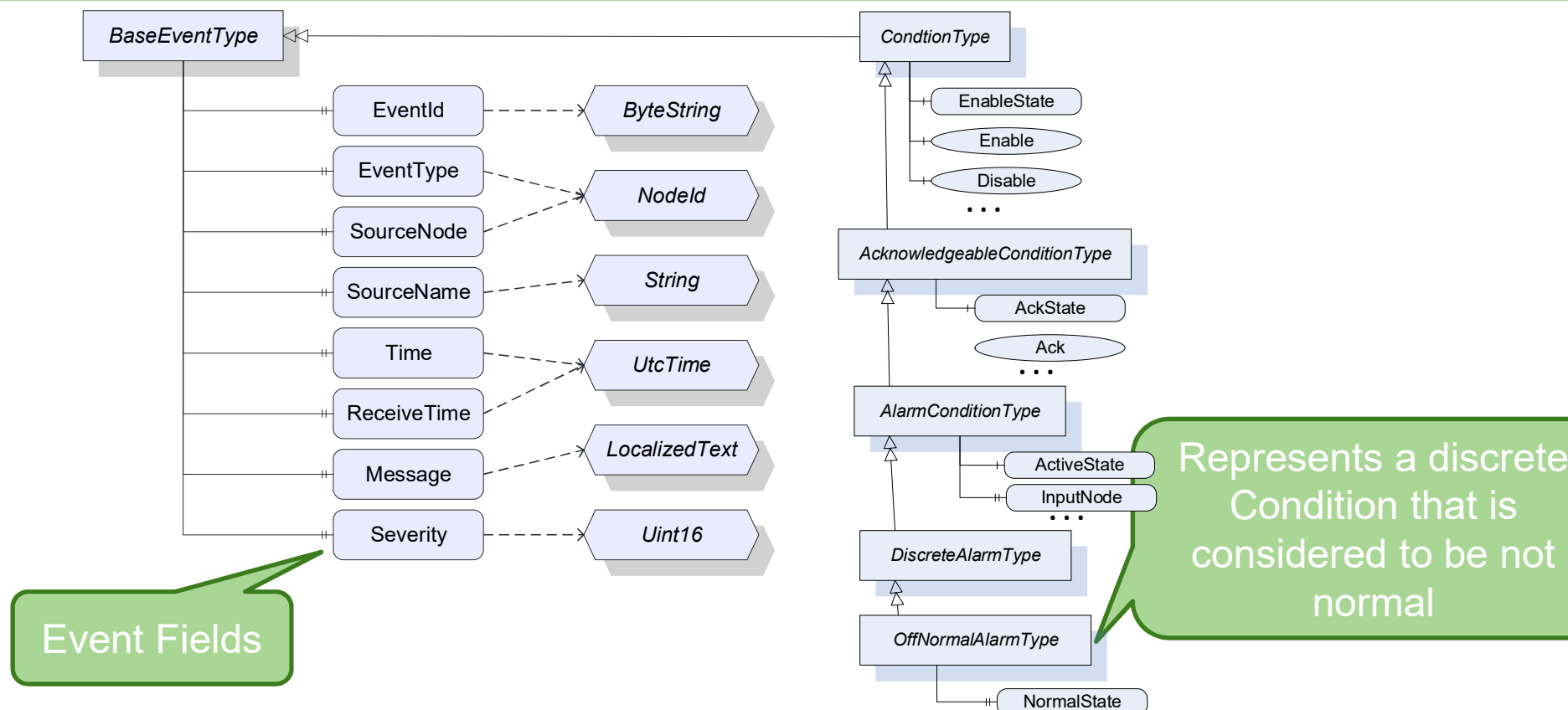
Control

Degree Openness

Time Opened

Cancel **OK**

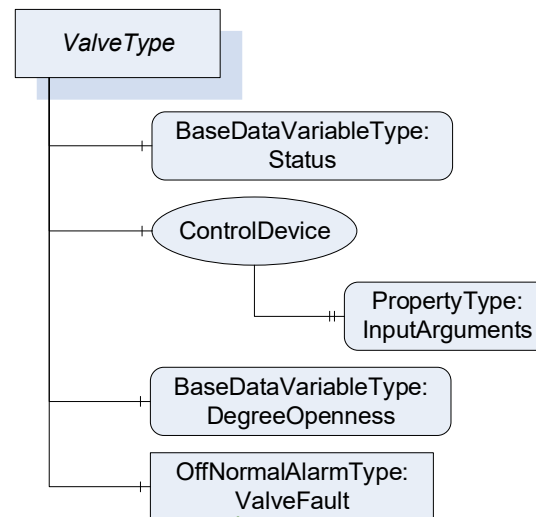
Events, Alarms and Conditions



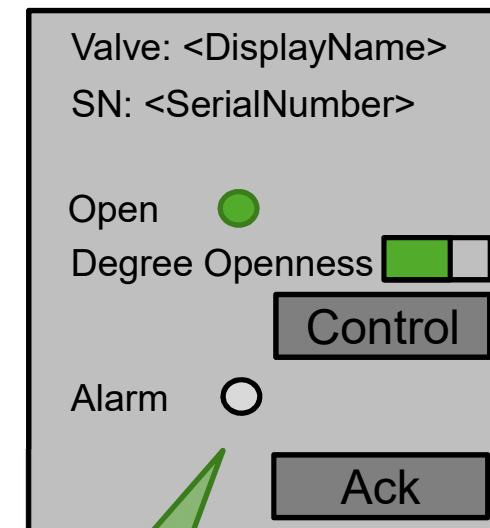
Let's model a Device – Valve



> Add an Alarm



Alarm can be represented as Object



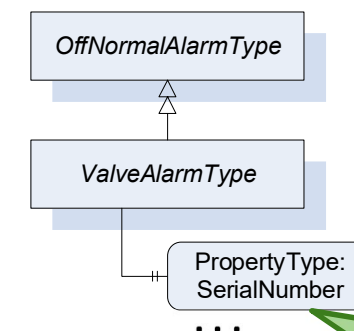
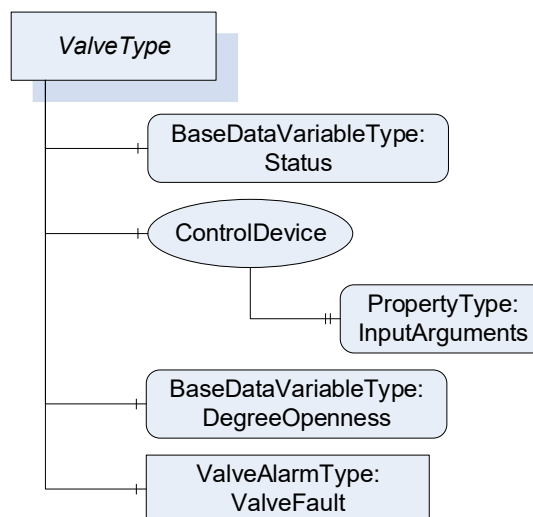
Updated UI

Let's model a Device – Valve

> Add an Alarm - Advanced



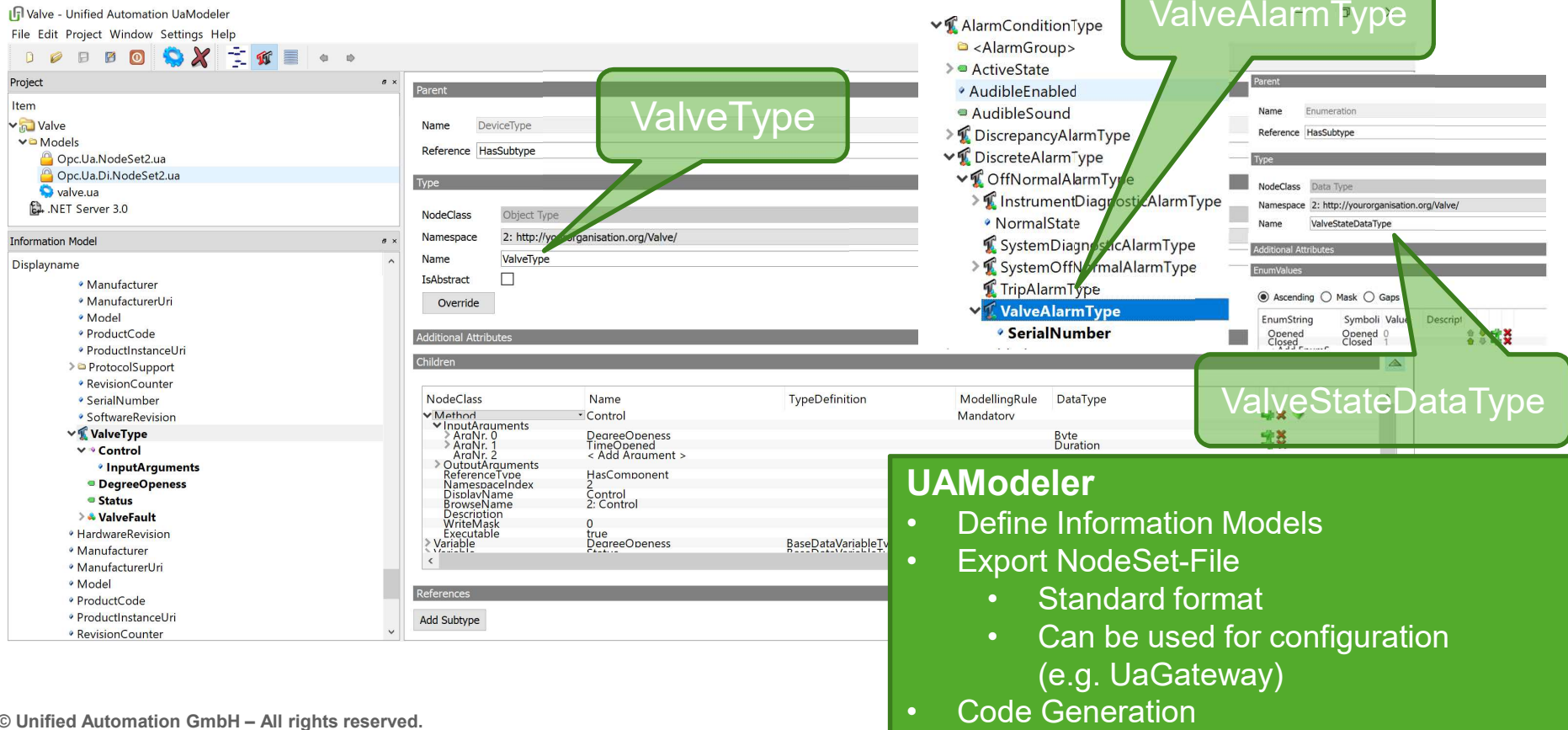
Event List



Subtype with additional Event Fields

EventId	Message	Severity	Time	Ack	Active	SerialNumber
123	Valve broken	90		NO	YES	#1234567890
125	Valve OK	5		NO	NO	#0987654321
129	Valve broken	80		YES	YES	#1234567890

Modelling is supported by tooling



The screenshot displays the UAModeler software interface. The left pane shows the project structure with 'Valve' as the root, containing 'Models' and 'valve.ua'. The main pane shows the 'ValveType' being edited, with its 'Children' table listing various properties and methods. A callout box labeled 'ValveType' points to the 'ValveType' entry in the 'Children' table. Another callout box labeled 'ValveAlarmType' points to the 'ValveAlarmType' entry in the 'AlarmConditionType' list. A third callout box labeled 'ValveStateDataType' points to the 'ValveStateDataType' entry in the 'EnumValues' list.

ValveType

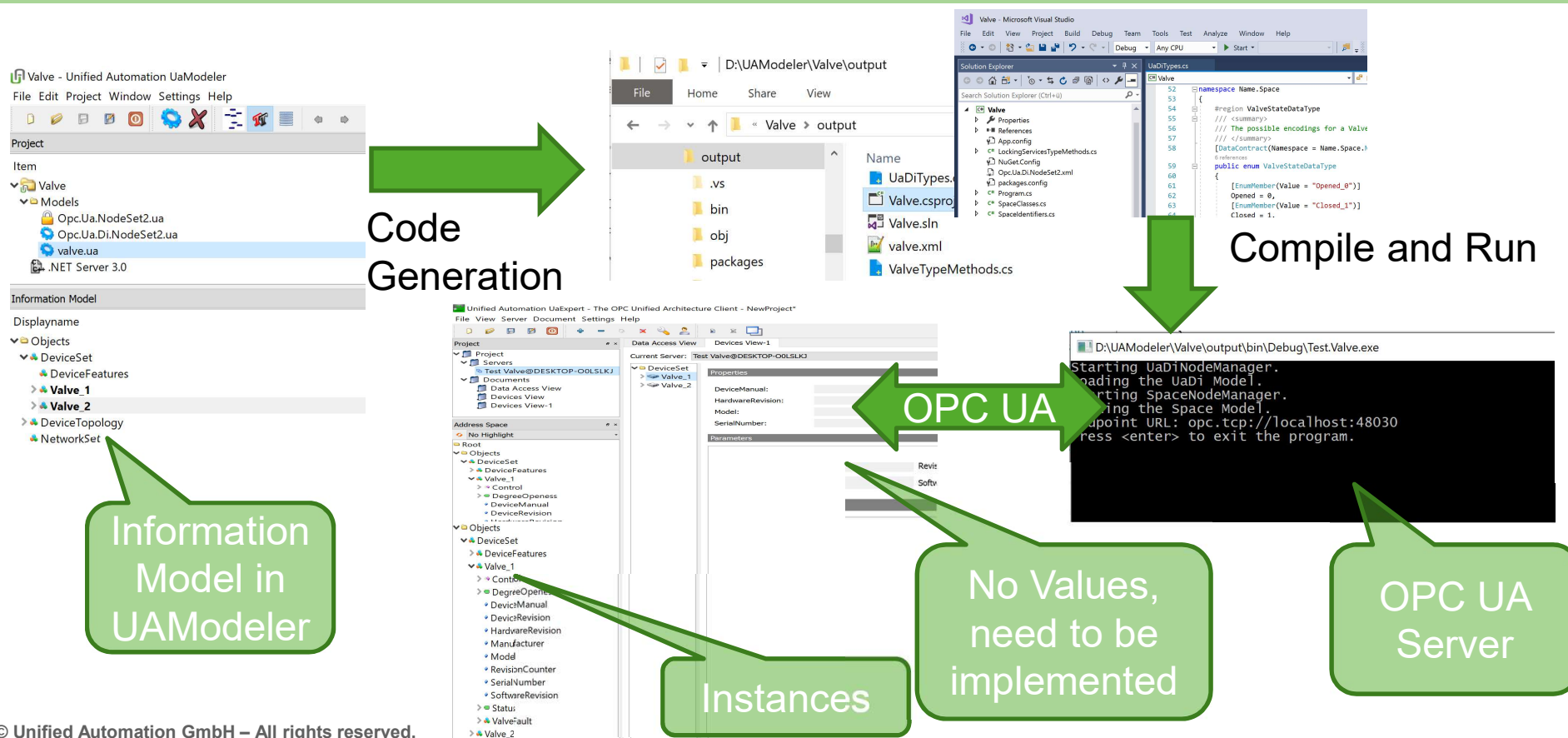
ValveAlarmType

ValveStateDataType

UAModeler

- Define Information Models
- Export NodeSet-File
 - Standard format
 - Can be used for configuration (e.g. UaGateway)
- Code Generation

Demo



Questions?



Dr. Wolfgang Mahnke

ascolab GmbH

wolfgang.mahnke@ascolab.com

OPC DAY FINLAND 2019

NOVEMBER 6.-7.11.2019 #OPCUA #OPCDAY #OPCDAYFINLAND #AUTOMAATIO



OPC Day Seminar: 13:00-17:30 Meeting room 203, 2nd

