Aalto University
School of Electrical
Engineering

OPC Unified Architecture Client-Server

Information systems in industry ELEC-E8113

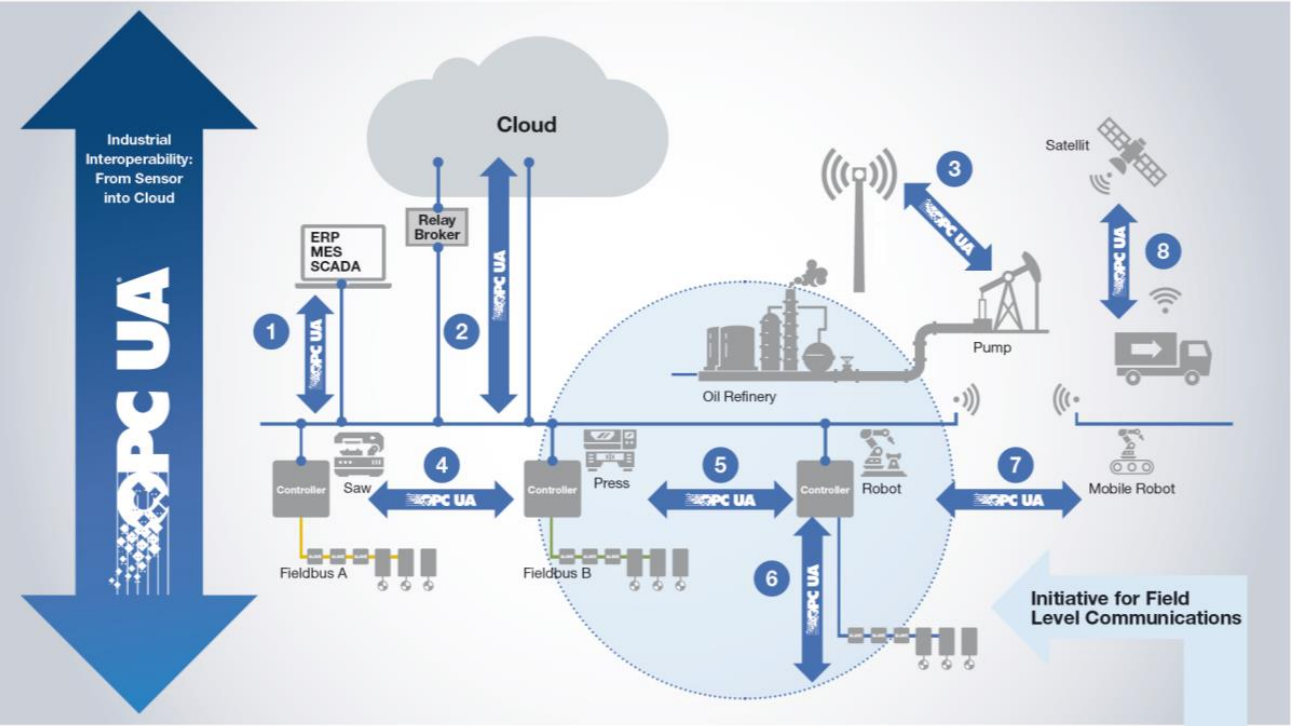
Start at 12.15!

Contents

- Architecture
- Address space model
- Built-in information models
- Companion information models

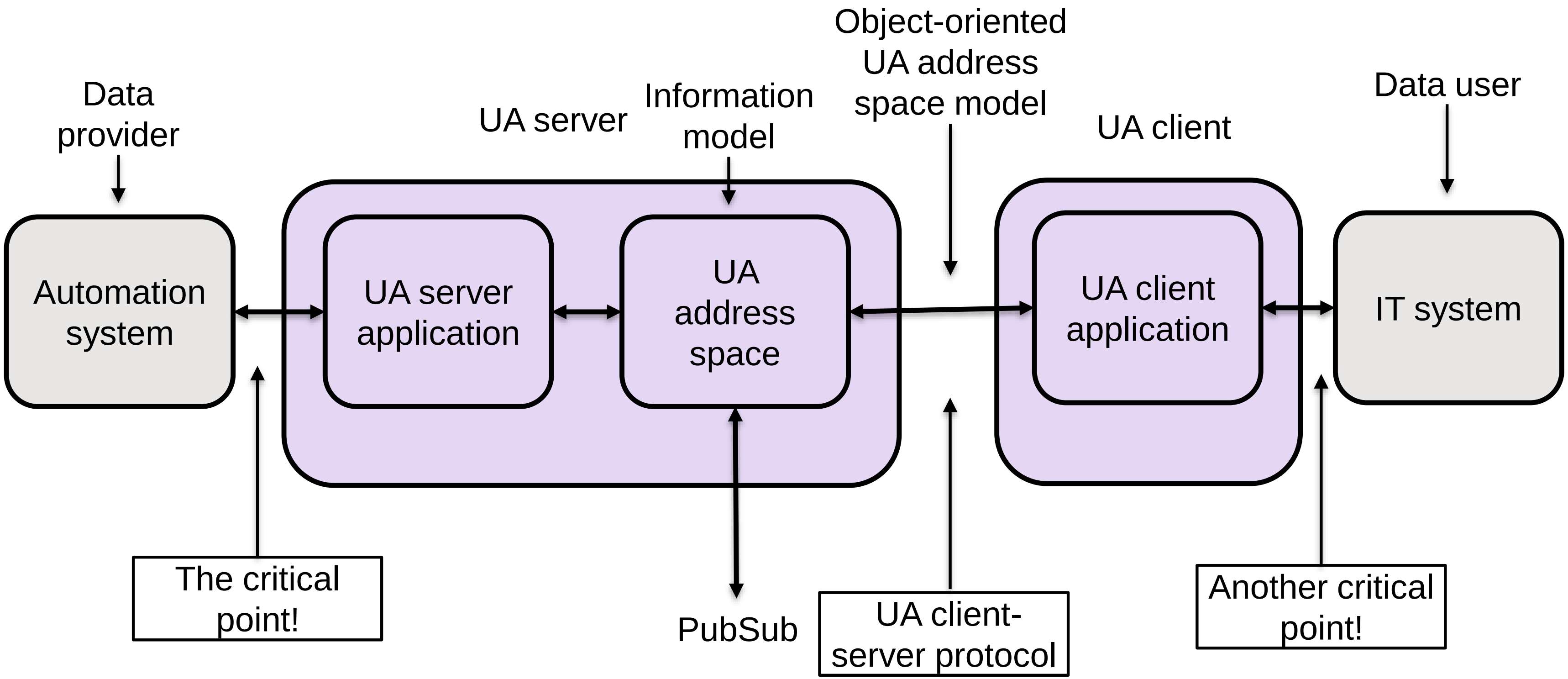
Rationale of the lecture: OPC UA is a good example of a communication technology particularly designed for data transfer between automation systems and information systems. OPC UA conforms to SOA.

Situation



1. IT / OT Communication
2. Cloud Integration
3. Secure Remote Access
4. Local OT Communication
5. Controller to Controller
6. Controller to Field Device
7. Wireless Integration (5G)
8. Future Ready

Focused view

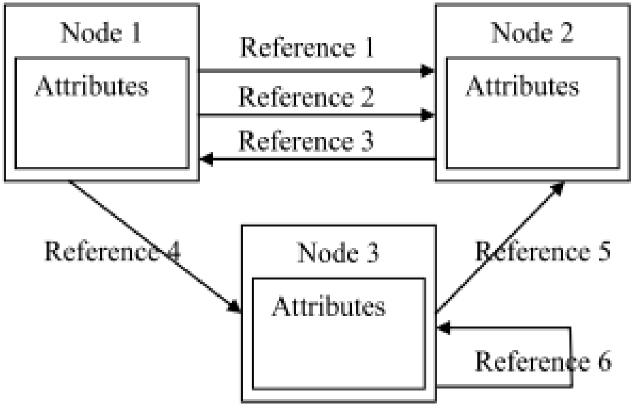


Basic concepts of OPC UA

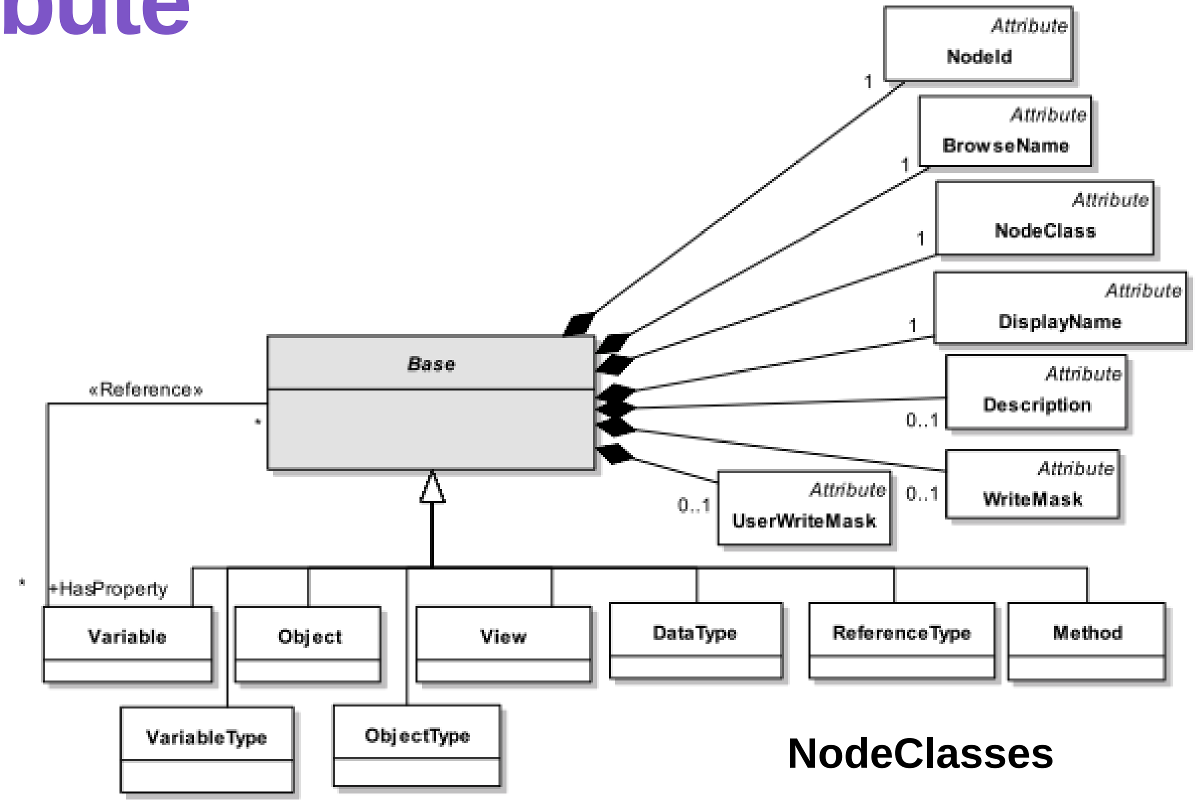
- **Specification and technology of communication developed by OPC Foundation and standardized by IEC**
- **Originally follows client-server model of communication. PubSub communication has been added afterwards.**
- **Conforms to SOA, but is not dependent on SOAP or REST**
- **Has a particular address space model, which allows representation of different information models**
- **Contains necessary security features**
- **Provides a migration path from currently widely used OPC Classic**

Address space model: Nodes and References

- The address space consists of Nodes and References for representing the object-oriented data model
- Nodes have Attributes
- Nodes have different NodeClass attributes for different types and instances of Objects and Variables (and a few other things)
- References have ReferenceType (and a few other things)
- The model is a network, not only hierarchy



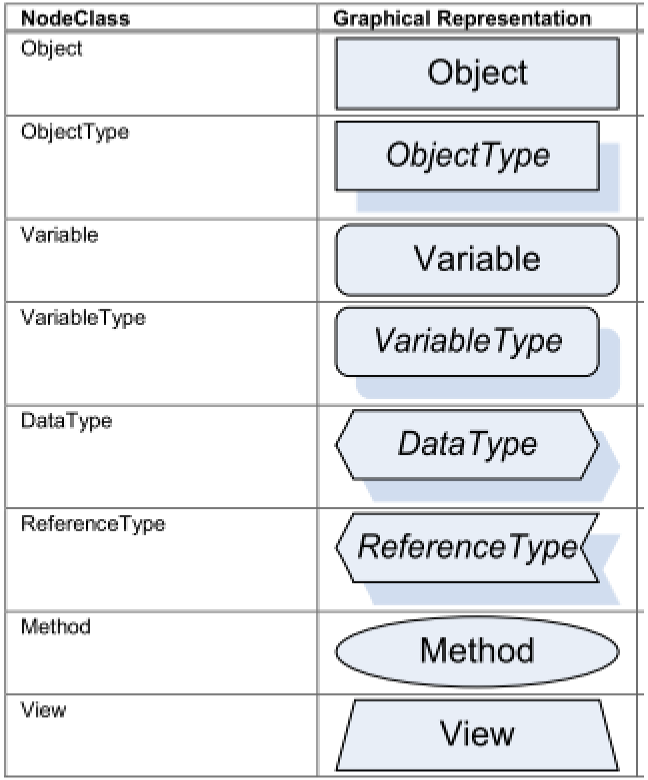
Address space model: NodeClass and Attribute



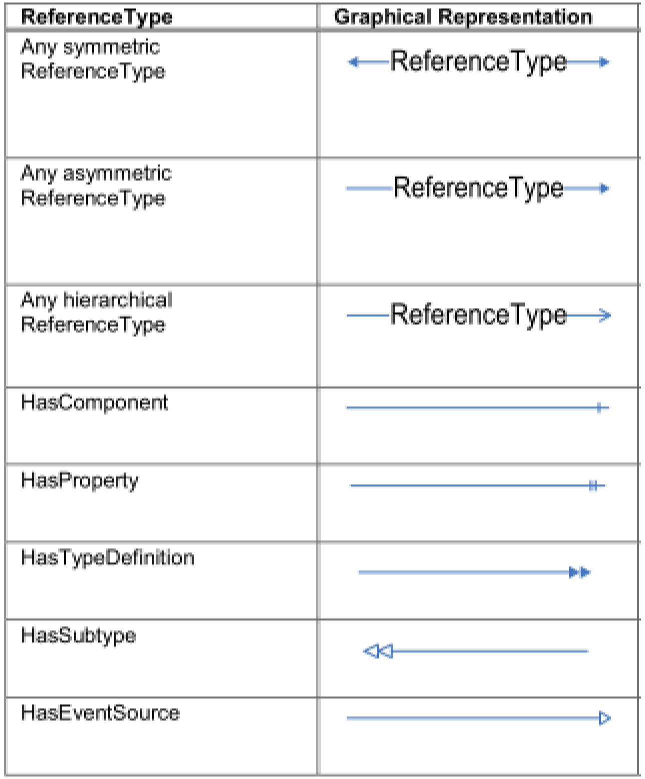
Attributes

NodeClasses

Address space model: graphical notation

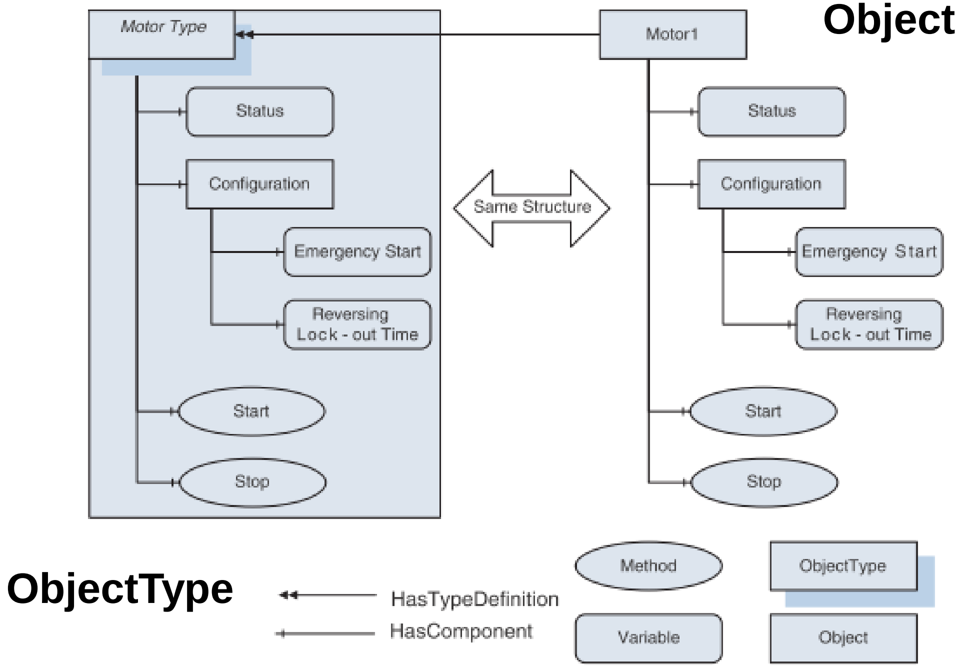


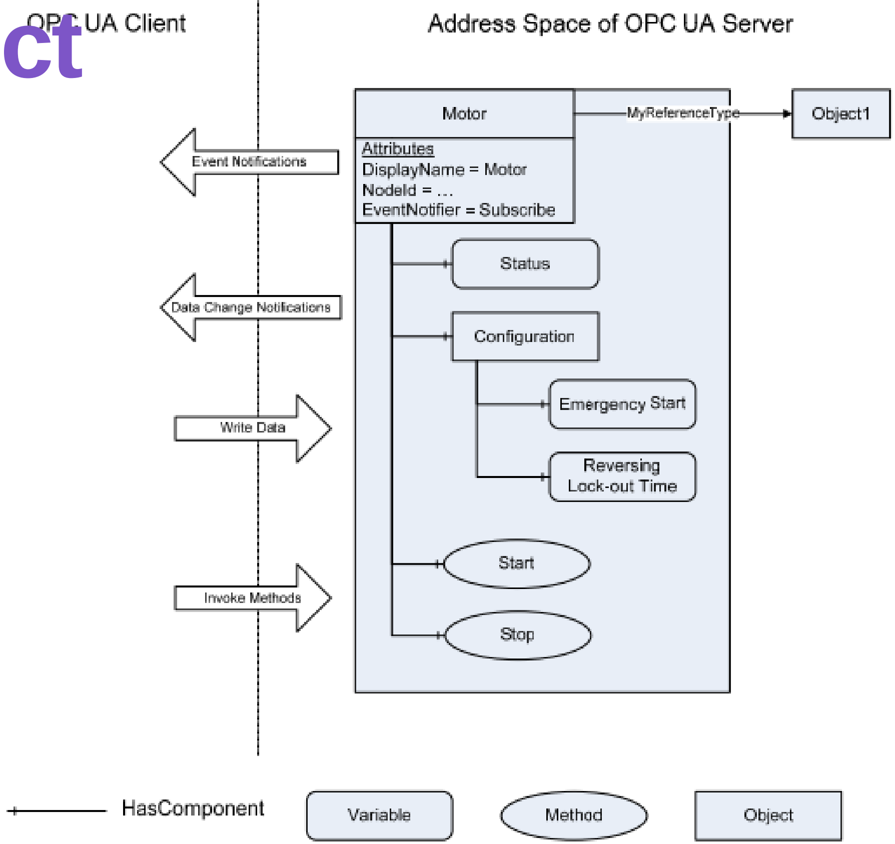
NodeClass	Graphical Representation
Object	Object
ObjectType	*ObjectType*
Variable	Variable
VariableType	*VariableType*
DataType	*DataType*
ReferenceType	*ReferenceType*
Method	Method
View	View



ReferenceType	Graphical Representation
Any symmetric ReferenceType	←ReferenceType→
Any asymmetric ReferenceType	—ReferenceType→
Any hierarchical ReferenceType	—ReferenceType—>
HasComponent	——+
HasProperty	——‖
HasTypeDefinition	——▸▸
HasSubtype	◁◁——
HasEventSource	——▷

Address space model: complex ObjectType and Object

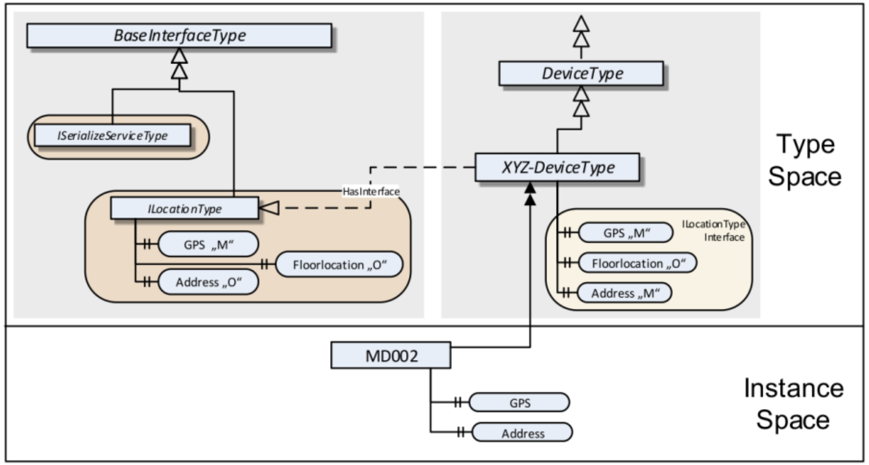




Aalto University
School of Electrical Engineering

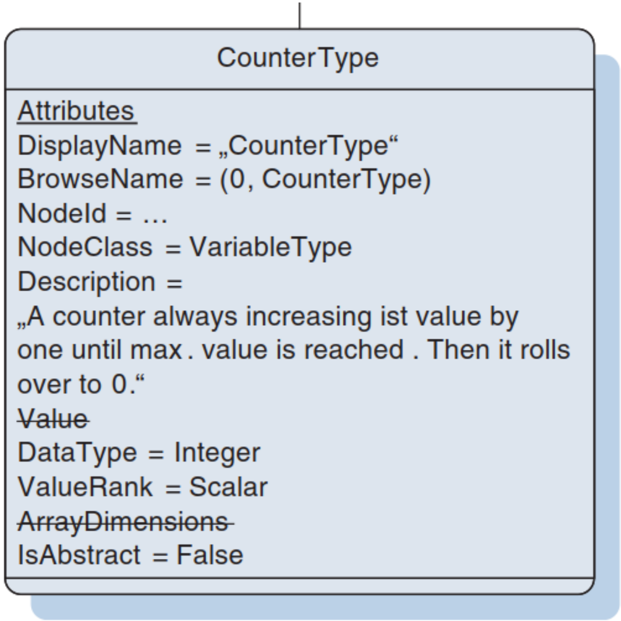
Modelling Rules and Interfaces

- Modelling rules can define if components or properties are mandatory or optional
- They can control inhertance in a type hierarchy
- Interfaces are subtypes of BaseInterfaceType. They can define properties and components
- ObjectTypes can refer to Interfaces with HasInterface Reference
- Interfaces and AddIns are a relatively recent addition to the OPC UA Address Space Model



Variables and VariableTypes

- Variables and VariableTypes have attributes e.g. Value, DataType, ValueRank and AccessLevel
- Value has server and source timestamps and status
- DataTypes can be predefined or application specific
- ValueRank differentiates beween scalars and arrays
- AccessLevel defines access rights
- Subtypes of BaseVariableType can define additional information

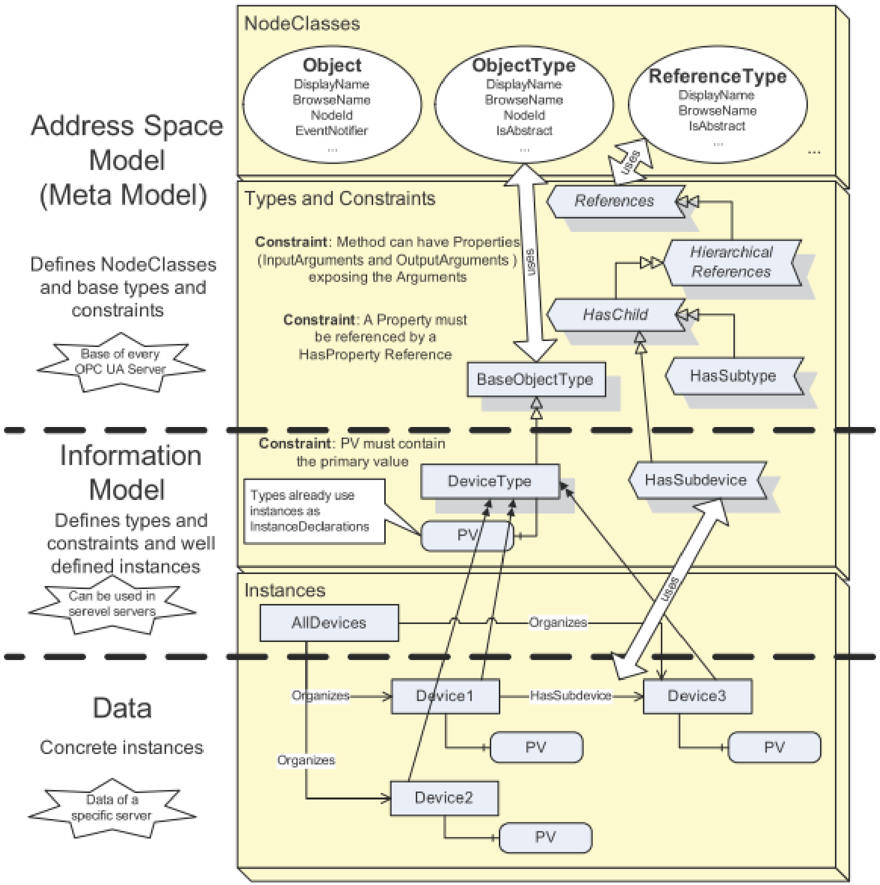


CounterType

Attributes
DisplayName = „CounterType“
BrowseName = (0, CounterType)
NodeId = …
NodeClass = VariableType
Description =
„A counter always increasing ist value by one until max . value is reached . Then it rolls over to 0.“
~~Value~~
DataType = Integer
ValueRank = Scalar
~~ArrayDimensions~~
IsAbstract = False

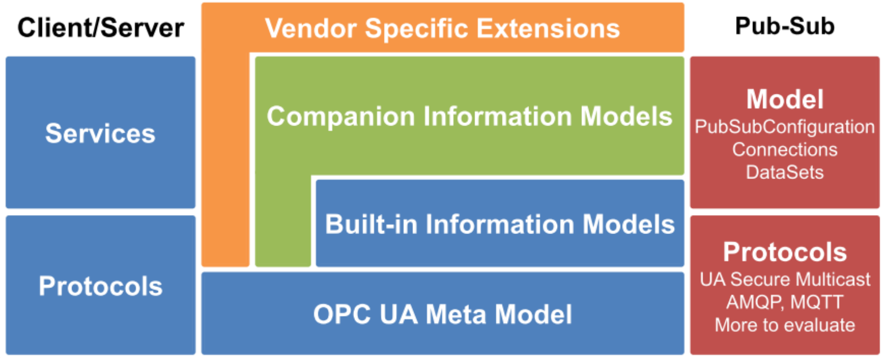
Information models

- **Information models contain types and some general instances intended to be useful for some applications**
- **The actual data will be available from the instances of the types of the information model**
- **Types of the information models are sub-types of the types in the built-in and other information models**


Address Space Model (Meta Model)
Defines NodeClasses and base types and constraints
Base of every OPC UA Server
Information Model
Defines types and constraints and well defined instances
Can be used in serevel servers
Data
Concrete instances
Data of a specific server
NodeClasses
Object
DisplayName
BrowseName
NodeId
EventNotifier
...
ObjectType
DisplayName
BrowseName
NodeId
IsAbstract
...
ReferenceType
DisplayName
BrowseName
IsAbstract
...
...
uses
Types and Constraints
References
Hierarchical References
HasChild
HasSubtype
uses
BaseObjectType
Constraint: Method can have Properties (InputArguments and OutputArguments) exposing the Arguments
Constraint: A Property must be referenced by a HasProperty Reference
Constraint: PV must contain the primary value
DeviceType
HasSubdevice
Types already use instances as InstanceDeclarations
PV
Instances
AllDevices
Organizes
uses
Organizes
Device1
HasSubdevice
Device3
PV
PV
Organizes
Device2
PV

Layers of information models

- The address spaces of OPC UA servers are intended to be extended with different information models for applications
- The standard contains a few built-in information models
- Companion information models are intended for particular application areas
- Vendors can specify their own extensions



Built-in information models

Name	Part	Content
Base information	5	Basic types and objects, e.g. Root object
Data access	8	Model for on-line values of variables
Alarms & conditions	9	Model for events and conditions
Programs	10	Model for programs and state machines
Historical access	11	Model for historical values of variables and events
Aggregates	13	Model for aggregated (calculated) values

- **More to come in parts 14-23 and maybe in 100-200?**

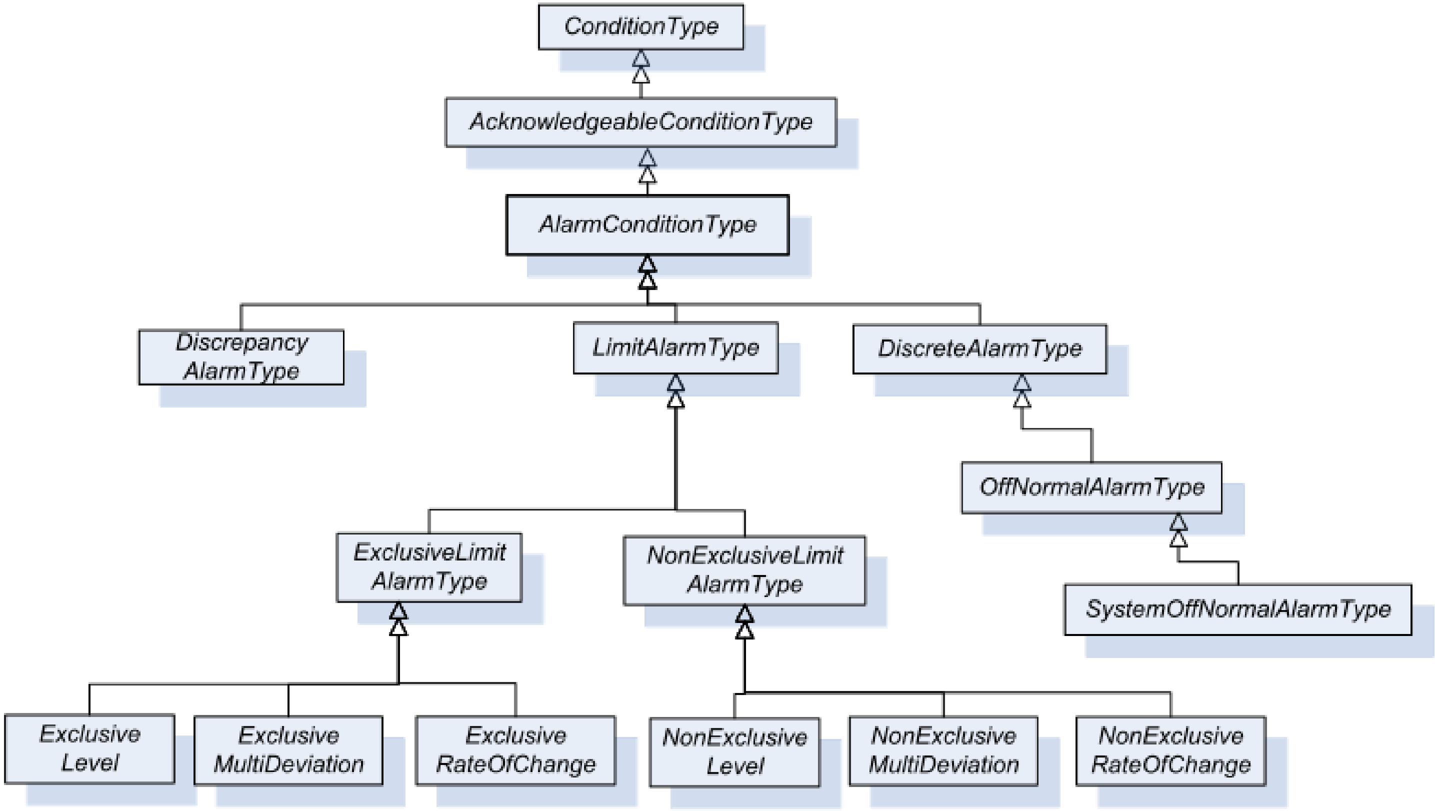
Base, data access, historical access, aggregates

- **Base**
  - Base types of objects and variables: BaseObjectType, BaseEventType, BaseVariableType, BaseInterfaceType, etc.
  - Standard hierarchy of objects: Root, Objects, Types, Views, Server, etc.
- **Data access (DA)**
  - Basic types of variables with more information: AnalogItemType, DiscreteItemType, ArrayItemType, etc.
- **Historical access (HA)**
  - Define how historical data is organized: HistoricalDataConfigurationType, etc.
- **Aggregates**
  - Functions for calculating aggregated data: AggregateFunctionType, etc.

Alarms & conditions, programs

- **Alarms & conditions**
  - Model of conditions which can be enabled/disabled and reacted to by a user: ConditionType (a subtype of BaseEventType), AcknowledgeableConditionType
  - Model of alarms: AlarmConditionType (a subtype of AcknowledgeableConditionType) and several subtypes
- **Programs**
  - State machine of long-running programs controlled through methods, state changes causing events: ProgramStateMachineType

Alarms & Conditions: type hierarchy

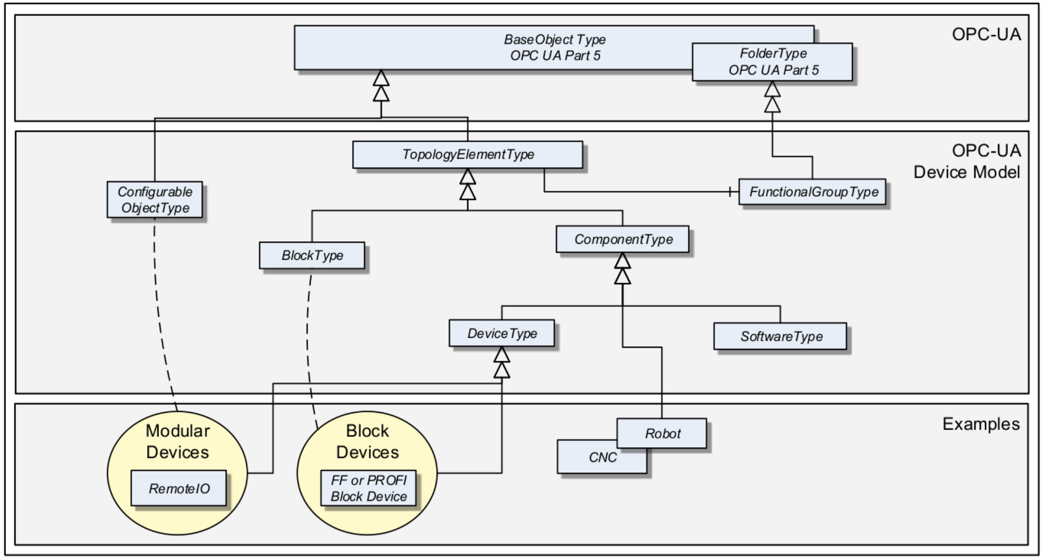


Companion specifications

- Currently work-in-process!
- About 30 companion specifications accepted by OPC Foundation
- Even more being developed while coordinated by VDMA
- OPC Unified Architecture, Part 100: Devices
  - Many sub-models, e.g. IEC61131-3
- Process automation
  - PA-DIM, FDI, FDT
- Batch process automation
  - PackML
- Discrete automation
  - Machinery, different kinds of machines, Robotics, Machine Vision, AAS, etc.
- Electric power systems
  - IEC61850
- And many more, and even more to come… how about harmonization?

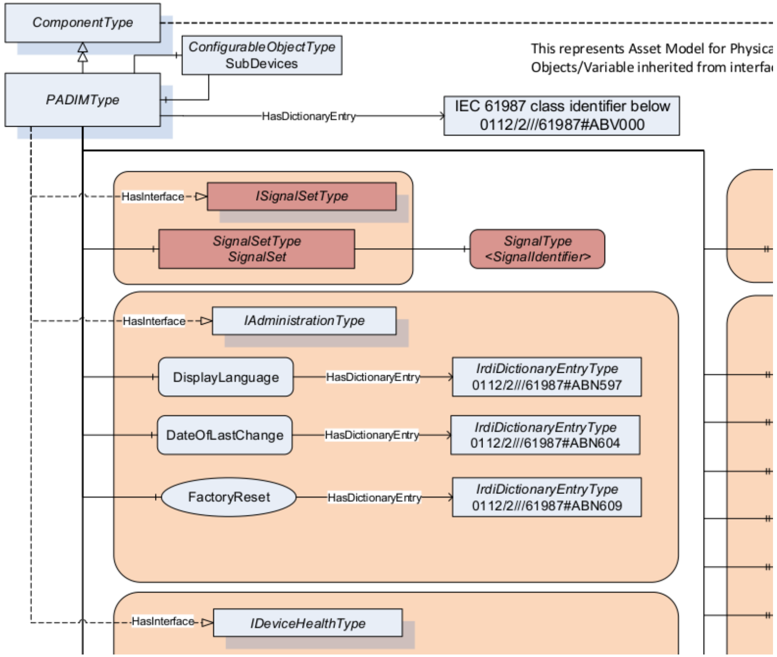
Devices

- **Abstract model of automation devices and networks**
- **Device model**
  - TopologyElementType, ComponentType, DeviceType, FunctionalGroupType, DeviceSet, etc.
- **Device communication model**
  - NetworkType, ConnectionPointType, NetworkSet, etc.
- **Device integration host model**
  - IsOnline, TransferServicesType, LockingServicesType, etc.



PA-DIM

- **PA-DIM can be used to describe data about devices in process industry**
- **PADIMType is the most important ObjectType in PA-DIM which describes a device and possible subdevices**
  - Several Interfaces for various viewpoints
  - Subtype of ComponentType (from OPC UA DI) and indirectly its supertypes
- **Several VariableTypes and some Methods**


ComponentType
ConfigurableObjectType
SubDevices
This represents Asset Model for Physica
Objects/Variable inherited from interfa
PADIMType
HasDictionaryEntry
IEC 61987 class identifier below
0112/2///61987#ABV000
HasInterface
ISignalSetType
SignalSetType
SignalSet
SignalType
<SignalIdentifier>
HasInterface
IAdministrationType
DisplayLanguage
HasDictionaryEntry
IrdiDictionaryEntryType
0112/2///61987#ABN597
DateOfLastChange
HasDictionaryEntry
IrdiDictionaryEntryType
0112/2///61987#ABN604
FactoryReset
HasDictionaryEntry
IrdiDictionaryEntryType
0112/2///61987#ABN609
HasInterface
IDeviceHealthType