European Plastics and Rubber Machinery

EUROMAP 82.4	OPC UA interfaces for plastics and rubber machinery - Peripheral devices - Part 4: Dosing Systems

Release Candidate 1.0.0, 2024-09-01

EUROMAP 82.4 (Release Candidate 1.0.0) is identical with

OPC 40082-4 (Release Candidate 1.0.0) and VDMA 40082-4:2024-09

www.euromap.org

EUROMAP Technical Commission
c/o VDMA • FV KuG

Lyoner Str. 18
DE 60528 Frankfurt am Main

Phone + 49 (69) 6603-1833
Fax + 49 (69) 6603-2833
E-Mail:euromap@vdma.org

Contents

Figures

Tables

OPC Foundation / EUROMAP

AGREEMENT OF USE

COPYRIGHT RESTRICTIONS

- This document is provided "as is" by the OPC Foundation and EUROMAP.
- Right of use for this specification is restricted to this specification and does not grant rights of use for referred documents.
- Right of use for this specification will be granted without cost.
- This document may be distributed through computer systems, printed or copied as long as the content remains unchanged and the document is not modified.
- OPC Foundation and EUROMAP do not guarantee usability for any purpose and shall not be made liable for any case using the content of this document.
- The user of the document agrees to indemnify OPC Foundation and EUROMAP and their officers, directors and agents harmless from all demands, claims, actions, losses, damages (including damages from personal injuries), costs and expenses (including attorneys' fees) which are in any way related to activities associated with its use of content from this specification.
- The document shall not be used in conjunction with company advertising, shall not be sold or licensed to any party.
- The intellectual property and copyright is solely owned by the OPC Foundation and EUROMAP.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OPC or EUROMAP specifications may require use of an invention covered by patent rights. OPC Foundation or EUROMAP shall not be responsible for identifying patents for which a license may be required by any OPC or EUROMAP specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OPC or EUROMAP specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

WARRANTY AND LIABILITY DISCLAIMERS

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OPC FOUDATION NOR EUROMAP MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OPC FOUNDATION NOR EUROMAP BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you.

RESTRICTED RIGHTS LEGEND

This Specification is provided with Restricted Rights. Use, duplication or disclosure by the U.S. government is subject to restrictions as set forth in (a) this Agreement pursuant to DFARs 227.7202-3(a); (b) subparagraph (c)(1)(i) of the Rights in Technical Data and Computer Software clause at DFARs 252.227-7013; or (c) the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 subdivision (c)(1) and (2), as applicable. Contractor / manufacturer are the OPC Foundation, 16101 N. 82nd Street, Suite 3B, Scottsdale, AZ, 85260-1830

COMPLIANCE

The combination of EUROMAP and OPC Foundation shall at all times be the sole entities that may authorize developers, suppliers and sellers of hardware and software to use certification marks, trademarks or other special designations to indicate compliance with these materials as specified within this document. Products developed using this specification may claim compliance or conformance with this specification if and only if the software satisfactorily meets the certification requirements set by EUROMAP or the OPC Foundation. Products that do not meet these requirements may claim only that the product was based on this specification and must not claim compliance or conformance with this specification.

TRADEMARKS

Most computer and software brand names have trademarks or registered trademarks. The individual trademarks have not been listed here.

GENERAL PROVISIONS

Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby.

This Agreement shall be governed by and construed under the laws of Germany.

This Agreement embodies the entire understanding between the parties with respect to, and supersedes any prior understanding or agreement (oral or written) relating to, this specification.

Forewords

OPC UA is a machine to machine communication technology to transmit characteristics of products (e.g. manufacturer name, device type or components) and process data (e.g. temperatures, pressures or feed rates). To enable vendor unspecific interoperability the description of product characteristics and process data has to be standardized utilizing technical specifications, the OPC UA companion specifications.

This specification was created by a joint working group of the OPC Foundation and EUROMAP. It is adopted identically as VDMA Specification.

EUROMAP

EUROMAP is the European umbrella association of the plastics and rubber machinery industry which accounts for annual sales of around 13.5 billion euro and a 40 per cent share of worldwide production. Almost 75 per cent of its European output is shipped to worldwide destinations. With global exports of 10.0 billion euro, EUROMAP's around 1,000 machinery manufacturers are market leaders with nearly half of all machines sold being supplied by EUROMAP members.

EUROMAP provides technical recommendations for plastics and rubber machines. In addition to standards for machine descriptions, dimensions and energy measurement, interfaces between machines feature prominently. The provision of manufacturer independent interfaces ensures high levels of machine compatibility.

OPC Foundation

OPC is the interoperability standard for the secure and reliable exchange of data and information in the industrial automation space and in other industries. It is platform independent and ensures the seamless flow of information among devices from multiple vendors. The OPC Foundation is responsible for the development and maintenance of this standard.

OPC UA is a platform independent service-oriented architecture that integrates all the functionality of the individual OPC Classic specifications into one extensible framework. This multi-layered approach accomplishes the original design specification goals of:

- Platform independence: from an embedded microcontroller to cloud-based infrastructure
- Secure: encryption, authentication, authorization and auditing
- Extensible: ability to add new features including transports without affecting existing applications
- Comprehensive information modelling capabilities: for defining any model from simple to complex

1 Scope

OPC 40082-4 describes the data model for dosing systems as part of a plastic & rubber production cell/line (e.g. extrusion/injection moulding) for data exchange with manufacturing execution systems (MES) and other components of the cell/line. Dosed materials may be granulate, powder or liquids. For liquid silicon and rubber dosing systems, please refer to OPC 40082-3: LSR Dosing Systems.

The target of OPC 40082-4 is to provide a unique interface for dosing systems from different manufacturers to ensure compatibility.

The following functionalities are covered:

– General information about the dosing system (manufacturer, model, serial number…), current configuration and status.

– Process information like throughput for monitoring and process optimization.

– Closed loop controls between the components of the production cell/line.

Following functions are not included:

– Safety related signals like emergency stop

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments and errata) applies

OPC 10000-1, *OPC Unified Architecture - Part 1: Overview and Concepts*

http://www.opcfoundation.org/UA/Part1/

OPC 10000-3, *OPC Unified Architecture - Part 3: Address Space Model*

http://www.opcfoundation.org/UA/Part3/

OPC 10000-5, *OPC Unified Architecture - Part 5: Information Model*

http://www.opcfoundation.org/UA/Part5/

OPC 10000-6, *OPC Unified Architecture - Part 6: Mappings*

http://www.opcfoundation.org/UA/Part6/

OPC 10000-7, *OPC Unified Architecture - Part 7: Profiles*

http://www.opcfoundation.org/UA/Part7/

OPC 10000-8, *OPC Unified Architecture - Part 8: Data Access*

http://www.opcfoundation.org/UA/Part8/

OPC 10000-9, *OPC Unified Architecture - Part 9: Alarms and Conditions*

http://www.opcfoundation.org/UA/Part9/

OPC 10000-100, *OPC Unified Architecture - Part 100: Devices*

http://www.opcfoundation.org/UA/Part100/

OPC 40001-1, *OPC UA for Machinery - Part 1: Basic Building Blocks*

http://www.opcfoundation.org/UA/Machinery/

OPC 40083: *OPC UA interfaces for plastics and rubber machinery – General Type definitions*

http://www.opcfoundation.org/UA/PlasticsRubber/GeneralTypes

3 Terms, definitions and conventions

3.1 Overview

It is assumed that basic concepts of OPC UA information modelling are understood in this specification. This specification will use these concepts to describe the OPC 40082-4 Information Model. For the purposes of this document, the terms and definitions given in the documents referenced in Clause 2 apply.

Note that OPC UA terms and terms defined in this specification are *italicized* in the specification.

3.2 Conventions used in this document

The conventions described in OPC 40083 apply.

3.3 Abbreviations

EXT Extruder

IMM Injection Moulding Machine

MES Manufacturing Execution System

4 General information to OPC UA interfaces for plastics and rubber machinery and OPC UA

For general information on OPC UA interfaces for plastics and rubber machinery and OPC UA see OPC 40083.

For liquid silicon and rubber dosing systems, please refer to OPC 40082-3: LSR Dosing Systems.

5 Use cases

The following functionalities are covered:

– General information about the dosing system (manufacturer, model, serial number…), current configuration and status.

– Optimizing process: Throughput of dosing has to be adjusted with other components in the production cell/line (e.g. extruder/haul-off speed).

– Process information (throughput, actual weight)

– Monitoring (Status, Errors)

– Recipe management: Dosing systems store their configurations in so-called recipes. These include information on nominal process parameters (temperatures, dosing volumes …). OPC 40082-4 allows transferring datasets between dosing systems and MES/line controllers for building a central repository of recipes.

6 OPC 40082-4 Information Model overview

The task of a dosing system is to bring material into the production process. This can be done in a continuous or discontinuous manner (batch) or a combination of both (see Figure 1 for an example). The material can be gravity fed or controlled and combined with an optional starve feeder (e.g. for extrusion).

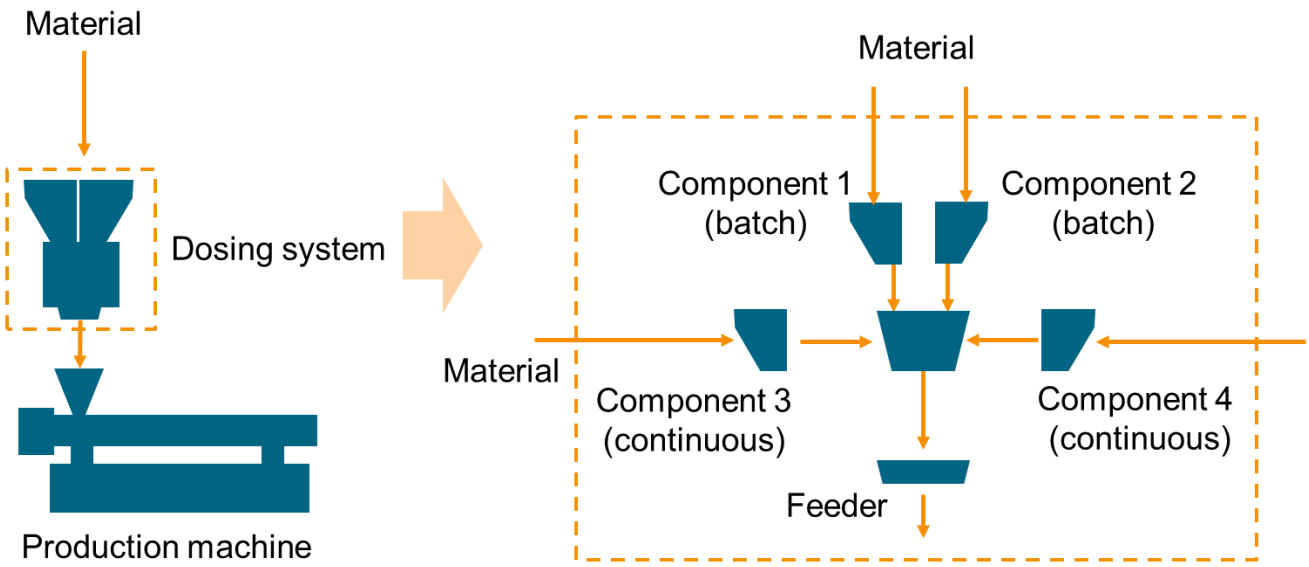

Figure 1: Dosing System example

The information model defined in this specification models the complex structure of a dosing system as follows:

The top level *ObjectType DosingSystemType* contains the objects for machine identification, machine state and configuration of the complete dosing system. The individual components are modelled in the *OperationType* with all properties required to monitor and control the operation of the system.

7 DosingSystemType

7.1 DosingSystemType definition

This OPC UA *ObjectType* is used for the root *Object* representing a dosing system as part of a production cell/line. It is formally defined in Table 1.

The instance(s) of *DosingSystemType* shall be located under the *Machines Object* of the Server (see OPC UA for Machinery).

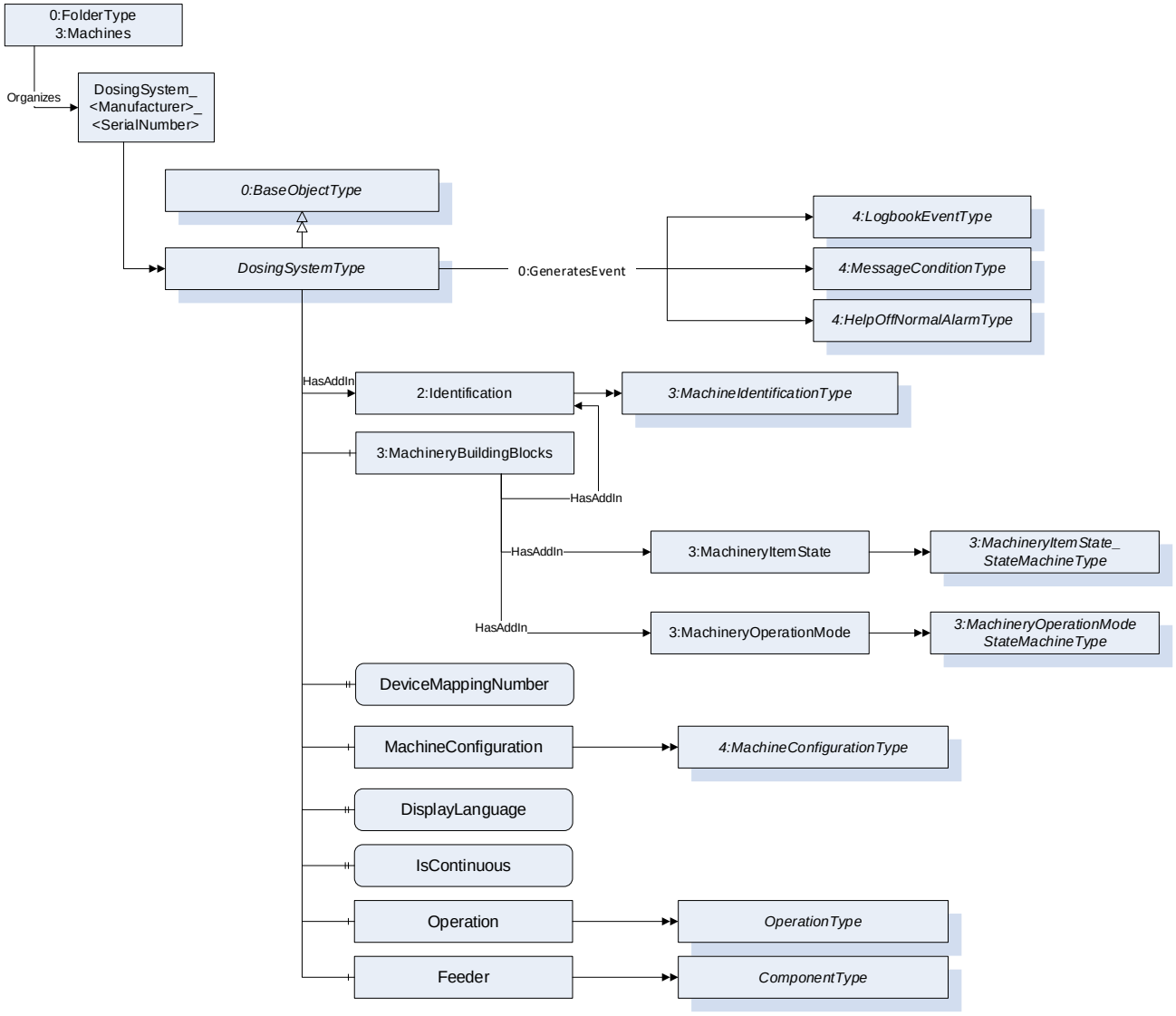

Figure 2: Overview DosingSystemType

NOTE: If the OPC UA server is implemented in the controller of the dosing system so only one instance of *DosingSystemType* will be created. But it is also possible that one OPC UA server is connected to several machine controls as one interface to the MES. In this case several instances of *DosingSystemType* will be created.

Table 1 – DosingSystemType Definition

Attribute	Value				
BrowseName	DosingSystemType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:BaseObjectType defined in OPC 10000-5					
0:HasAddIn	Object	2:Identification		3:MachineIdentification Type	M
0:HasComponent	Object	3:MachineryBuildingBlocks		0:FolderType	M
0:HasComponent	Object	MachineConfiguration		4:MachineConfigurationType	M
0:HasProperty	Variable	DisplayLanguage	0:LocaleId	0:PropertyType	O, RW
0:HasProperty	Variable	IsContinuous	0:Boolean	0:PropertyType	M, RO
0:HasComponent	Object	Operation		OperationType	M
0:HasProperty	Variable	DeviceMappingNumber	0:UInt32	0:PropertyType	M, RW
0:HasComponent	Object	Feeder		ComponentType	O
0:GeneratesEvent	ObjectType	4:LogbookEventType	Defined in OPC 40083		
0:GeneratesEvent	ObjectType	4:MessageConditionType	Defined in OPC 40083		
0:GeneratesEvent	ObjectType	4:HelpOffNormalAlarmType	Defined in OPC 40083		
Conformance Units					
OPC 40082-4 Basic					
OPC 40082-4 Alarms					

7.2 Identification and MachineryBuildingBlocks

The *MachineIdentificationType* is defined in OPC UA for Machinery (OPC 40001-1) and provides basic information on a machine/device.

For the *InstanceDeclaration* the *ModellingRules* of the *Properties Model* and *DeviceClass* are overridden to mandatory.

The *Object MachineryBuildingBlocks* contains building blocks from OPC UA for Machinery as defined in OPC 40001-1. For this version of OPC 40082-4, the *Object* uses the two *AddIns MachineryItemState* and *MachineryOperationMode*.

Table 2 – DosingSystemType Additional Subcomponents

BrowsePath	References	NodeClass	BrowseName	DataType	TypeDefinition	Other
2:Identification	0:HasProperty	Variable	2:Model	0:LocalizedText	0:PropertyType	M, RO
2:Identification	0:HasProperty	Variable	2:DeviceClass	0:String	0:PropertyType	M, RO
3:MachineryBuilding Blocks	0:HasAddIn	Object	2:Identification		3:MachineIdentificationType	M
3:MachineryBuilding Blocks	0:HasAddIn	Object	3:MachineryItem State		3:MachineryItemState_State MachineType	M
3:MachineryBuilding Blocks	0:HasAddIn	Object	3:MachineryOpe rationMode		3:MachineryOperationMode StateMachineType	M

The *DeviceClass Property* shall have the value “Dosing System”.

7.3 DeviceMappingNumber

Description: Unique identifier/address/number for devices of the same *DeviceType* within a local network. Several peripheral devices of the same *DeviceType* can be connected to a production machine (e.g. IMM or extruder). In most applications, the production machine must map the connected peripheral devices to internal logical devices and zones in a fixed configuration (e.g. hot runner systems according to the wiring or temperature control devices according to the tubing).

The mapping shall be stable after reconnecting the devices and is therefore not possible via IP addresses, which can be assigned dynamically via DHCP. *DeviceMappingNumber* sets the mapping order of peripheral devices of the same type on the local network and is therefore of type *UInt32*.

Example: 1

7.4 MachineConfiguration

The *MachineConfiguration Object* represents the current configuration of the Dosing System. The *MachineConfigurationType* is defined in OPC 40083.

7.5 DisplayLanguage

With the *DisplayLanguage Property* the client can set the desired language on the user interface at the Dosing System. If the peripheral device does not support the configured language, it can keep the previous setting or use English as the default.

7.6 IsContinuous

The dosing device can be installed on an extruder (EXT) or an injection moulding machine (IMM). EXT is a continuous process, IMM is a cyclic process. For inline dosing devices, directly mounted on the feed throat of an IMM the *IsContinuous* flag is false, when installed on an EXT this flag is true.

For batch blenders which are cyclic based by themselves, this flag is set true since the output flow of a batch blender does not need to be synchronized with for example an IMM.

7.7 Operation

This *ObjectType* contains parameters which are necessary to operate the Dosing System. The *OperationType* is defined in 8.

7.8 Feeder

A feeder can be used for starve feeding (under feeding) of a production machine and is defined by the *ComponentType*. E.g. a screw feeder where *DosingMode* is set to ONLY_CONVEYING. The *ComponentType* is defined in 12.

8 OperationType

8.1 OperationType definition

The *OperationType* contains objects which are necessary to operate the *DosingSystem*. It is formally defined in Table 3. The top level of the *OperationType* describes the whole dosing system while detailed information about the single components can be found in the *ComponentsType Object*.

Figure 3: OperationType Overview

Table 3 – OperationType Definition

Attribute	Value				
BrowseName	OperationType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	IsActive	0:Boolean	0:PropertyType	O, RO
0:HasComponent	Variable	SetActive	0:Boolean	0:BaseDataVariableType	O, RW
0:HasComponent	Method	IdentifyDevice			O
0:HasComponent	Method	EnableDevice			M
0:HasProperty	Variable	OperationMode	4:MachineModeEnumeration	0:PropertyType	M, RO
0:HasComponent	Method	SetOperationMode			M
0:HasComponent	Variable	ActivateRemoteControl	0:UInt16	0:MultiStateValueDiscreteType	M, RW
0:HasComponent	Variable	RemoteControlActivated	0:UInt16	0:MultiStateValueDiscreteType	M, RO
0:HasComponent	Method	StartDosing			O
0:HasComponent	Method	StopDosing			O
0:HasComponent	Method	StopDosingAfterCycle			O
0:HasComponent	Object	Components		ComponentsType	M
0:HasComponent	Object	Throughput		4:MonitoredParameterType	O, RW
0:HasComponent	Object	ShotWeight		4:MonitoredParameterType	O, RW
0:HasProperty	Variable	DosingDuration	0:Duration	0:PropertyType	O, RW
0:HasComponent	Object	Mixing		MixingType	O
0:HasComponent	Object	BatchWeight		4:MonitoredParameterType	O, RW
0:HasComponent	Object	MaterialConsumption		MaterialConsumptionType	O
0:HasComponent	Object	Maintenance		DosingMaintenanceType	O
0:HasComponent	Object	ProductionDataSetManagement		4:ProductionDatasetManagementType	O
0:HasComponent	Object	ElectricalEnergy		4:EnergyType	O
0:HasProperty	Variable	HighestActiveAlarmSeverity	0:UInt16	0:PropertyType	M, RO
0:HasComponent	Variable	ActiveErrors	4:ActiveErrorDataType[]	0:BaseDataVariableType	M, RO
0:HasComponent	Method	ResetAllErrors			O
0:HasComponent	Method	ResetErrorById			O
0:HasComponent	Method	FirstFilling			O
Conformance Units					
OPC 40082-4 Basic					

8.2 IsActive

The *IsActive* Property provides information if the dosing unit is active in the current production.

8.3 SetActive

The *SetActive* property activates the dosing unit for the current production.

NOTE: Malfunction or activation by mistake should be covered on machine side (not part of this specification).

8.4 IdentifyDevice

The *DosingSystem* on which this method is called shows itself by e.g. activation of a LED.

Signature: `IdentifyDevice();`

Table 4 – IdentifyDevice Method AddressSpace Definition

Attribute	Value				
BrowseName	IdentifyDevice				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

NOTE: This Method is identical to the IdentifyDevice Method in OPC 40082-1.

8.5 EnableDevice

Enables the device, for example activate motor drives or set a dosing system in state “RUN”.

Signature: `EnableDevice();`

Table 5 – EnableDevice Method AddressSpace Definition

Attribute	Value				
BrowseName	EnableDevice				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

8.6 OperationMode

The value *OperationMode* represents the current machine operation mode.

The *MachineModeEnumeration* is defined in OPC 40083.

8.7 SetOperationMode

Method to set the OperationMode of the dosing device. The argument of this method is the *OperationMode*, which is defined in 8.6.

Note: Not all modes have to be supported by the dosing device. In this case, the server shall respond with the status code “Bad_InvalidArgument”.

Signature

```
SetOperationMode (
   [in] 4:MachineModeEnumeration    OperationMode);
```

Table 6 – SetOperationMode Method Arguments

Argument	Description
OperationMode	Desired machine operation mode

Table 7 – SetOperationMode Method AddressSpace Definition

Attribute	Value				
BrowseName	SetOperationMode				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
0:HasProperty	Variable	0:InputArguments	0:Argument[]	0:PropertyType	M

8.8 ActivateRemoteControl

It is necessary to synchronize the dosing between production machine and dosing systems. This can be done via a separate interface e.g. via hardwired signals or also via OPC UA (if the process is robust against small time delays that can be caused by the client/server-connection)
With *ActivateRemoteControl* the client selects the method of remote control. If the server provides only one method for remote control, the other one is not listed in the possible values of the *MultiStateValueDiscreteType*.

Table 8: Values for ActivateRemoteControl and RemoteControlActivated

EnumValue	ValueAsText	Description
0	OFF	Remote control switched off (local/HMI control)
1	SEPARATE_INTERFACE	Using a separate interface from the production machine for remote controlling this dosing device
2	OPC_UA	Using this OPC UA connection for remote controlling this dosing device

On the dosing device, the *OperationMode* can be set regardless of the RemoteControl state.

8.9 RemoteControlActivated

With this feedback signal, the dosing device signalizes if it is ready to be controlled via this or a separate interface . See Table 8 for possible values.

8.10 StartDosing

If *RemoteControlActivated* = 2 (OPC UA), this method can be used to start dosing, no arguments.

If *RemoteControlActivated* is in any other state than 2 (OPC UA), the return statement shall be “Bad_InvalidState”

In case of inline dosing directly mounted on the feed throat of the IMM, this method needs to be called every cycle at start of plastification.

Signature

```
StartDosing();
```

Table 9: StartDosing Method AddressSpace Definition

Attribute	Value				
BrowseName	StartDosing				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

8.11 StopDosing

Method to stop dosing immediately. No arguments.

Signature

```
StopDosing();
```

Table 10 – StopDosing Method AddressSpace Definition

Attribute	Value				
BrowseName	StopDosing				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

8.12 StopDosingAfterCycle

Method to stop dosing after the current cycle (e.g. for batch blending). No arguments.

Signature

```
StopDosingAfterCycle();
```

Table 11 – StopDosingAfterCycle Method AddressSpace Definition

Attribute	Value				
BrowseName	StopDosingAfterCycle				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

8.13 Throughput

Current throughput of the *DosingSystem* in mass or volume per time (e.g. kg/h). The modelling rule for this node is optional to cover pure feeders, but is mandatory for all other dosing units.

8.14 ShotWeight

Current shotweight of the parts produced by the production machine in mass (e.g. kg).

8.15 DosingDuration

Used for inline dosing where the dosing device is mounted on the feed throat of the production machine. This parameter defines the duration the dosing shall be active after receiving the StartDosing signal.

If DosingDuration is used, the dosing device shall stop dosing after the duration. *StopDosing* or *StopDosingAfterCycle* are not necessary in this case but shall have a higher priority if used anyway.

8.16 Mixing

In case the dosing station is equipped with an active mixing device like a batch blender, this object provides information about the mixing process. The *MixingType* is defined in chapter 9.

8.17 BatchWeight

Target batch weight for batch blenders in absolute values.

8.18 Components

This *ObjectType* is a container for all *Components* in the *DosingSystem*. The *ComponentsType* is described in Chapter 11.

8.19 MaterialConsumption

Total material consumption for this dosing device. The *MaterialConsumptionType* is defined in chapter 10.

8.20 HighestActiveAlarmSeverity

Description: Indication of the severity of the highest active alarm (0 = no active alarm – 1000 = possible error). Together with *ActiveErrors*, it provides a minimal error handling for devices without alarm support. However, the variable shall be filled even if alarms are supported.

Example: 400

8.21 ActiveErrors

Description: List of the active errors of the device. It provides a minimal error handling for devices without alarm support. However, the variable shall be filled even if alarms are supported. The *ActiveErrorDataType* is defined in OPC 40083. If there is no active error, the array is empty.

8.22 ResetAllErrors

Description: Method to reset all errors of the device.

Signature

```
ResetAllErrors();
```

8.23 ResetErrorById

Description: Method to reset one error of the device.

Signature

```
ResetErrorById(
   [in]    String          Id);
```

Table 12 –ResetErrorById Method Arguments

Argument	Description
Id	Id of the error, listed in *ActiveErrors*, that shall be reset.

Table 13 – ResetErrorById Method AddressSpace Definition

Attribute	Value				
BrowseName	ResetErrorById				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

8.24 FirstFilling

Method to fill the *DosingSystem* for the first time with the mixture of materials from the current recipe.

No Arguments.

Signature: `FirstFilling();`

Table 14: FirstFilling Method AddressSpace Definition

Attribute	Value				
BrowseName	FirstFilling				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

8.25 ElectricalEnergy

Information about the electrical power and energy consumption of the *DosingSystem*. The *EnergyType* is defined in OPC 40083.

8.26 Maintenance

Information on the maintenance status of the dosing system. The *DosingMaintenanceType* is defined in chapter 15.

8.27 ProductionDatasetManagement

The *ProductionDatasetManagementType* is defined in OPC 40083 and provides functionalities for the management of recipes/machine settings.

9 MixingType

9.1 MixingType definition

In case the dosing station is equipped with an active mixing device like on a batch blender, the mixing process can be controlled with the type *MixingType,* which is defined in Table 15.

Table 15: MixingType definition

Attribute	**Value**				
BrowseName	MixingType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of *0:BaseObjectType* defined in OPC UA Part 5					
0:HasComponent	Variable	MixingMode	0:UInt16	0:MultiStateValueDiscreteType	M, RW
0:HasProperty	Variable	MixingInterval	0:UInt16	0:PropertyType	O, RW
0:HasProperty	Variable	MixingDuration	0:Duration	0:PropertyType	O, RW
0:HasComponent	Variable	IdleMixing	0:Boolean	0:BaseDataVariableType	O, RW

9.2 MixingMode

The mixing mode parameter can be used to set the required mixing mode. The MultiStateValueDiscreteType should provide the available mixing modes.

Table 16: Values for ActivateRemoteControl and RemoteControlActivated

EnumValue	**ValueAsText**	**Description**
0	NO_MIXING	No mixer available or mixing OFF
1	CONTINUOUS_MIXING	Mixer is activated after the first batch(blender) and is mixing continuous until the dosing device is stopped
2	TIMED_MIXING	The mixer is started after each batch (blender) and mixes for a defined duration (MixingDuration 9.4)
3	INTERVAL_MIXING	The mixer device is started after a defined number of shots, IMM (MixingInterval 9.3). Duration of mixing is set by *MixingDuration*

9.3 MixingInterval

Interval between two mixing processes, in number of shots

Note: Mandatory if *MixingMode* is set to *INTERVAL_MIXING*

9.4 MixingDuration

Duration of a mixing process

Note: Mandatory, if *MixingMode* is set to *TIMED_MIXING* or *INTERVAL_MIXING*

9.5 IdleMixing

This flag can be used for “idle mixing”. With idle mixing the mixer of the batch blender rotates once in a while to keep the material in the mixer flowing and material does not get blocked by the mixer blades.

10 MaterialConsumptionType

10.1 MaterialConsumptionType definition

Provides information about the material consumption. Used by *Operation* and *Components*.

Table 17: MaterialConsumptionType definition

Attribute	**Value**				
BrowseName	MaterialConsumptionType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of *0:BaseObjectType* defined in OPC UA Part 5					
0:HasComponent	Variable	TotalConsumption	0:Double	0:AnalogUnitType	M, RO
0:HasComponent	Variable	Scrap	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Method	Reset			M

10.2 TotalConsumption

Total mass/volume of fed material (good + scrap) since last reset.

10.3 Scrap

Total mass/volume of fed scrap since last reset.

10.4 Reset

The *Method Reset* resets the two counters *TotalConsumption* and *Scrap*. The signature of this *Method* is specified below. Table 18 specifies the *AdressSpace* representation.

Signature

```
Reset()
```

Table 18 – Reset Method AddressSpace Definition

Attribute	**Value**				
BrowseName	Reset				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**

11 ComponentsType

11.1 ComponentsType definition

This *ObjectType* is a container for all *Component*s in the *DosingSystem*. It is formally defined in Table 19.

Table 19 – ComponentsType Definition

Attribute	**Value**				
BrowseName	ComponentsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	<Component_Nr>		ComponentType	MP
Conformance Units					
OPC 40082-4 Basic					

When instances for components are created, the *BrowseNames* shall be “<Component_Nr>” where Nr is a three-digit number with leading zeros, starting with “001”. The *ComponentType* is defined in chapter 12.

12 ComponentType

12.1 ComponentType definition

The *ComponentType* represents a device which brings material into the dosing system.

The *ComponentType* is abstract. A derived subtype, either for continuous dosing (*ContinuousComponentType)* or discontinuous dosing (*BatchComponentType)* shall be used.

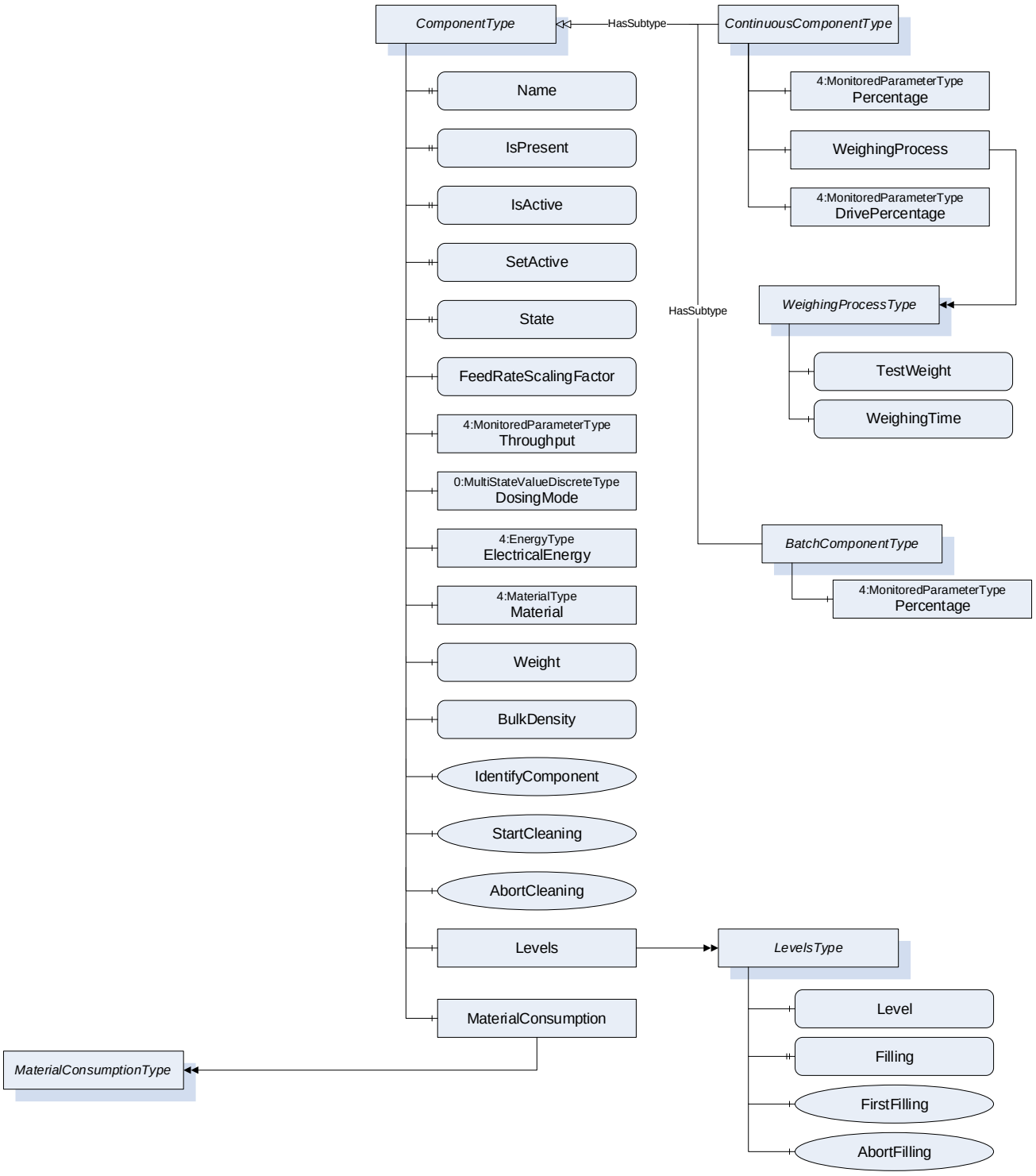

Figure 4: ComponentType Overview

Table 20 – ComponentType Definition

Attribute	Value				
BrowseName	ComponentType				
IsAbstract	True				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	Name	0:LocalizedText	0:PropertyType	M, RO
0:HasProperty	Variable	IsPresent	0:Boolean	0:PropertyType	M, RO
0:HasProperty	Variable	IsActive	0:Boolean	0:PropertyType	O, RO
0:HasComponent	Variable	SetActive	0:Boolean	0:BaseDataVariableType	O, RW
0:HasComponent	Variable	State	ComponentStateEnumeration	0:BaseDataVariableType	M, RO
0:HasProperty	Variable	FeedRateScalingFactor	0:Double	0:PropertyType	O, RW
0:HasComponent	Object	Throughput		4:MonitoredParameterType	O, RW
0:HasComponent	Variable	DosingMode	0:UInt16	0:MultiStateValueDiscreteType	M, RW
0:HasComponent	Object	ElectricalEnergy		4:EnergyType	O
0:HasComponent	Object	Material		4:MaterialType	O
0:HasComponent	Object	MaterialConsumption		MaterialConsumptionType	O
0:HasComponent	Variable	Weight	0:Double	0:AnalogUnitType	O, RO
0:HasComponent	Variable	BulkDensity	0:Double	0:AnalogUnitType	O, RW
0:HasComponent	Method	IdentifyComponent			O
0:HasComponent	Method	StartCleaning			O
0:HasComponent	Method	AbortCleaning			O
0:HasComponent	Object	Levels		LevelsType	O
0:HasSubtype	ObjectType	ContinuousComponentType	Defined in 12.17		
0:HasSubtype	ObjectType	BatchComponentType	Defined in 0		
Conformance Units					
OPC 40082-4 Basic					

12.2 Name

Component identification name available/configured

12.3 IsPresent

The *IsPresent* flag indicates if the hardware of this component is installed. This *IsPresent* system is used for those Dosing equipment manufacturers using a fixed OPC UA model for different hardware models

12.4 SetActive

The *SetActive* property activates the dosing component for the current production. *SetActive* = *False* deactivates the component, even if the setpoint is bigger than 0% in the recipe.

NOTE: In some cases where the dosing *Percentage* setpoint of this component is bigger than 0% and *SetActive* = False, the real total material output of the dosing station is not 100% but 100%-x% of this component.

Example: a dosing station with 3 components, respectively 80%, 15%, 5% and an Extruder *Throughput* demand of 100kg/h. The component of 5% SetActive = False. The dosing system will actually supply 95% (95kg/h).

12.5 IsActive

The *IsActive* Property provides information if the dosing component is active in the current production.

12.6 State

The *State* property of this component to read the actual working state of the component.

Table 21 – ComponentStateEnumeration Definition

Name	Value	Description
Off	0	The component has been switched off
Standby	1	The component is waiting for a start dosing signal
Dosing	2	The component is actively feeding material
Calibration	3	Component is in calibration mode and can typically not be used for dosing or filling
Error	4	The component has an active error
Cleaning	5	The component is in cleaning mode e.g. hopper draining

12.7 FeedrateScalingFactor

Output of the *Component* per *DrivePercentage*.

Example:

A dosing screw with a maximum rotational speed of 300 rpm, the material output flow is 900 grams per second. 300 rpm means 100% *DrivePercentage*. The Feederate would then be 900/100 = 9 gram per % per second (g/(%*s))

Running at 50% *DrivePercentage*, the output will be 50*9=450 g/s.

Example:

A vibrational feeder running at 50Hz, the material output flow is 10 gram per second. The maximum frequency of the feeder is 250 Hz. 250Hz equals 100% DrivePercentage, 50Hz equals 20% DrivePercentage. The federate is 10/20 = 0.5 gram per % per second (g/%/s).

12.8 Throughput

Material Output per time. For example 100kg/h.

12.9 DosingMode

The *DosingMode* Property provides information, how the dosing of the component is controlled.

Table 22 – DosingModeEnumeration Definition

Name	Value	Description
ONLY_CONVEYING	0	The throughput is not controlled. The feeder only transports the material (e.g. by screw, conveyor belt) or the material is only falling through a feed opening
VOLUMETRIC	1	The throughput is controlled by a volumetric dosing system.
GRAVIMETRIC	2	The throughput is controlled by a gravimetric dosing system.
OTHER	3	Throughput is controlled, but in another mode than these above

12.10 ElectricalEnergy

Information about the electrical power and energy consumption of the component. The *EnergyType* is defined in OPC 40083.

12.11 Material

Information about the material used in the component. The *MaterialType* is defined in OPC 40083.

12.12 Weight

Weight of the Material (e.g. inside the hopper)

12.13 BulkDensity

Bulk density of the material which is dosed. Weight per litre material, kg/l. (not to be confused with the density of the material itself defined in MaterialType)

12.14 IdentifyComponent

The dosing component on which this method is called shows itself by e.g. activation of a LED.

Signature: `IdentifyComponent();`

NOTE: This Method is identical to the IdentifyDevice Method in OPC 40082-1.

Table 23 – IdentifyComponent Method AddressSpace Definition

Attribute	Value				
BrowseName	IdentifyComponent				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

12.15 StartCleaning

Start the automated cleaning sequence for this component. Once cleaning the *State* will be set to “Cleaning” and can be used to monitor if cleaning sequence has finished.

Method without arguments.

Signature: `StartCleaning();`

Table 24 – StartCleaning Method AddressSpace Definition

Attribute	Value				
BrowseName	StartCleaning				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

12.16 AbortCleaning

Method without arguments to abort the current cleaning sequence.

Signature: `AbortCleaning();`

Table 25 – AbortCleaning Method AddressSpace Definition

Attribute	Value				
BrowseName	AbortCleaning				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

12.17 Levels

Information about the filling of the dosing component. The *LevelsType* is defined in chapter 13.

12.18 MaterialConsumption

Information about the material consumption of the component. The *MaterialConsumptionType* is defined in chapter 10.

12.19 ContinuousComponentType

12.19.1 ContinuousComponentType definition

The *ContinuousComponentType* is a subtype of the abstract *ComponentType* and describes *Components*, where the materials are fed to the production machine in a continuous flow instead of batches. Injection moulding can be considered as continuous flow during plastification.

The *ContinuousComponentType* is formally defined in Table 26.

Table 26: ContinuousComponentType Definition

Attribute	**Value**				
BrowseName	ContinuousComponentType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of ComponentType defined in 12					
0:HasComponent	Object	Percentage		4:MonitoredParameterType	O
0:HasComponent	Object	WeighingProcess		WeighingProcessType	O
0:HasComponent	Object	DrivePercentage		4:MonitoredParameterType	O

12.19.2 Percentage

Part (%) of the recipe dosed by this component (recipe).

NOTE: The sum of the Percentages of all components inside a Dosing System must add up to 100%, regrind excluded.

12.19.3 WeighingProcess

The weighing process can be used to perform a material pre-calibration or to determine the feedrate of the dosing system used by this component.

The *WeighingProcessType* is defined in chapter 14.

12.19.4 DrivePercentage

Relative value (0-100%) of the maximum drive power of the component drive.

Example:

A dosing screw with a maximum rotational speed of 300 rpm. 300 rpm equals 100% *DrivePercentage*. At 50% *DrivePercentage* the screw rotates 150 rpm.

12.20 BatchComponentType

12.20.1 BatchComponentType definition

The *BatchComponentType* is a subtype of the abstract *ComponentType* and describes *Components*, where the materials are fed to the production machine in batches e.g. prepared by a batch blender.

Table 27 – BatchComponentType Definition

Attribute	**Value**				
BrowseName	BatchComponentType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of ComponentType defined in 12					
0:HasComponent	Object	Percentage		4:MonitoredParameterType	M

12.20.2 Percentage

Part (%) of the recipe dosed by this component (recipe).

NOTE: The sum of the Percentages of all components inside a Dosing System must add up to 100%, regrind excluded.

13 LevelsType

13.1 LevelsType definition

The *LevelsType* gives information about the filling of the dosing component and is described in Table 28.

Table 28: LevelsType Definition

Attribute	**Value**				
BrowseName	LevelsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of *0:BaseObjectType* defined in OPC UA Part 5					
0:HasComponent	Object	Level		4:MonitoredParameterType	M
0:HasComponent	Variable	Filling	0:Boolean	0:BaseDataVariableType	O
0:HasComponent	Method	StartFilling			O
0:HasComponent	Method	AbortFilling			O

13.2 Level

The actual level of the material inside the component (e.g. actual hopper level).

Values are given as relative values between 0 – 100 %.

MinValue of *MonitoredParameterType* is used to trigger an alarm.

13.3 Filling

Flag to indicate if the hopper is being filled right now.

13.4 StartFilling

Method to fill the *Component* up to the *MaxValue* of the *MonitoredParameterType* in the *Level Variable.*

No Arguments.

Signature: `StartFilling();`

Table 29 – StartFilling Method AddressSpace Definition

Attribute	**Value**				
BrowseName	StartFilling				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

13.5 AbortFilling

Method to abort filling the *Component*.

No Arguments.

Signature: `AbortFilling();`

Table 30 – AbortFilling Method AddressSpace Definition

Attribute	**Value**				
BrowseName	AbortFilling				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**

14 WeighingProcessType

14.1 WeighingProcessType definition

The weighing process can be used to perform a material pre-calibration or to determine the feedrate of the dosing system used by this component.

Table 31: WeighingProcessType Definition

Attribute	**Value**				
BrowseName	WeighingProcessType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 0:*BaseObjectType* defined in OPC 10000-5					
0:HasComponent	Variable	TestWeight	0:Double	0:AnalogUnitRangeType	M, RW
0:HasProperty	Variable	WeighingTime	0:Duration	0:PropertyType	M, RW

14.2 TestWeight

Weight of the material of this component

14.3 WeighingTime

Duration of the weighing process.

15 DosingMaintenanceType

15.1 DosingMaintenanceType definition

The *DosingMaintenanceType* is a *Subtype* of the *MaintenanceType*, which is defined in OPC 40083. The *Inverval Variable* is mandatory and read/writeable in this subtype.

Table 32: DosingMaintenanceType Definition

Attribute	**Value**				
BrowseName	DosingMaintenanceType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of 4:MaintenanceType defined in 40083					
0:HasComponent	Variable	4:Interval	0:Double	0:AnalogItemType	M, RW

16 Alarmmanagement

All alarms shall be from the *HelpOffNormalAlarmType* defined in OPC 40083.

As defined in OPC 40083, the root node of the specific interface, e.g. an instance of *DosingSystemType*, sets the *SubscribeToEvents* flag in the *EventNotifier* attribute.

The client subscribes to events at this root node and receives the events already defined in this specification, such as temperature limit alarms or diagnostic events.

In the case of *Component* -related messages, the *SourceNode* (included in *BaseEventType*) shall contain the *NodeId* of the related component. In case of medium or high severity, the production machine can sort out bad parts or stop production.

17 Profiles and Conformance Units

This chapter defines the corresponding profiles and conformance units for the OPC UA Information Model for OPC 40082-4. *Profiles* are named groupings of conformance units. Facets are profiles that will be combined with other *Profiles* to define the complete functionality of an OPC UA *Server* or *Client.*

17.1 Conformance Units

This chapter defines the corresponding *Conformance Unit* for OPC 40082-4.

Table 33 – Conformance Units for OPC 40082-4

Category	Title	Description
Server	OPC 40082-4 Basic	Support of *DosingSystemType* and all mandatory child elements giving information on the dosing system and its status. There is at least one instance of the *DosingSystemType* in the *Machines Object.*
Server	OPC 40082-4 Alarms	Support of *HelpOffNormalAlarmType,* providing error information. If this facet is supported and a client subscribes to the events, the server shall provide all errors via alarms in addition to the error variables included in the *OperationType*

17.2 Profiles

17.2.1 Profile list

The following tables specify the facets available for *Servers* that implement the OPC 40082-4Information Model companion specification.

NOTE: The names of the supported profiles are available in the *Server Object* under *ServerCapabilities.ServerProfileArray*

Table 34 lists all Profiles defined in this document and defines their URIs.

Table 34 – Profile URIs for OPC 40082-4

Profile	URI
OPC 40082-4 Basic Server Profile	http://opcfoundation.org/UA-Profile/PlasticsRubber/Dosing/Server/Basic
OPC 40082-4 Alarms Server Facet	http://opcfoundation.org/UA-Profile/PlasticsRubber/Dosing/Server/Alarms

17.2.2 Server Facets

17.2.2.1 Overview

The following sections specify the *Facets* available for *Servers* that implement the OPC 40082-4 companion specification. Each section defines and describes a *Facet* or *Profile*.

17.2.2.2 OPC 40082-4 Basic Server Profile

Table 35 - OPC 40082-4 Basic Server Profile

Group	Conformance Unit / Profile Title	Mandatory / Optional
OPC 40082-4	OPC 40082-4 Basic	M
Profile	0:ComplexType Server Facet	M
Profile	0:Method Server Facet	M
Profile	2:BaseDevice Server Facet	M

17.2.2.3 OPC 40082-3 Alarms Server Facet

Table 36 - OPC 40082-4 Alarms Server Facet

Group	Conformance Unit / Profile Title	Mandatory / Optional
OPC 40082-4	OPC 40082-4 Alarms	M
Profile	0:A & C Alarm Server Facet	M

18 Namespaces

18.1 Namespace Metadata

Table 37 defines the namespace metadata for this document. The *Object* is used to provide version information for the namespace and an indication about static *Nodes*. Static *Nodes* are identical for all *Attributes* in all *Servers*, including the *Value Attribute*. See OPC 10000-5 for more details.

The information is provided as *Object* of type *NamespaceMetadataType*. This *Object* is a component of the *Namespaces Object* that is part of the *Server Object*. The *NamespaceMetadataType ObjectType* and its *Properties* are defined in OPC 10000-5.

The version information is also provided as part of the ModelTableEntry in the UANodeSet XML file. The UANodeSet XML schema is defined in OPC 10000-6.

Table 37 – NamespaceMetadata Object for this Document

Attribute	**Value**	
BrowseName	http://opcfoundation.org/UA/PlasticsRubber/Dosing/	
Property	**DataType**	**Value**
NamespaceUri	String	http://opcfoundation.org/UA/PlasticsRubber/Dosing/
NamespaceVersion	String	RC 1.0.0
NamespacePublicationDate	DateTime	2024-09-01
IsNamespaceSubset	Boolean	False
StaticNodeIdTypes	IdType []	0
StaticNumericNodeIdRange	NumericRange []	
StaticStringNodeIdPattern	String	

Note: The *IsNamespaceSubset Property* is set to False as the UaNodeSet XML file contains the complete Namespace. *Servers* only exposing a subset of the Namespace need to change the value to True.

18.2 Handling of OPC UA Namespaces

Namespaces are used by OPC UA to create unique identifiers across different naming authorities. The *Attributes NodeId* and *BrowseName* are identifiers. A *Node* in the UA *AddressSpace* is unambiguously identified using a *NodeId*. Unlike *NodeIds*, the *BrowseName* cannot be used to unambiguously identify a *Node*. Different *Nodes* may have the same *BrowseName*. They are used to build a browse path between two *Nodes* or to define a standard *Property*.

Servers may often choose to use the same namespace for the *NodeId* and the *BrowseName*. However, if they want to provide a standard *Property*, its *BrowseName* shall have the namespace of the standards body although the namespace of the *NodeId* reflects something else, for example the *EngineeringUnits Property*. All *NodeIds* of *Nodes* not defined in this document shall not use the standard namespaces.

Table 38 provides a list of mandatory and optional namespaces used in an OPC 40082-4 OPC UA *Server*.

Table 38 – Namespaces used in a OPC 40082-4 Server

NamespaceURI	Description	Use
http://opcfoundation.org/UA/	Namespace for *NodeIds* and *BrowseNames* defined in the OPC UA specification. This namespace shall have namespace index 0.	Mandatory
Local Server URI	Namespace for nodes defined in the local server. This namespace shall have namespace index 1.	Mandatory
http://opcfoundation.org/UA/DI/	Namespace for *NodeIds* and *BrowseNames* defined in OPC 10000-100. The namespace index is *Server* specific.	Mandatory
http://opcfoundation.org/UA/Machinery/	Namespace for *NodeIds* and *BrowseNames* defined in OPC UA for Machinery – Part 1: Basic Building Blocks (OPC 40001-1). The namespace index is *Server* specific.	Mandatory
http://opcfoundation.org/UA/PlasticsRubber/ GeneralTypes/	Namespace for *NodeIds* and *BrowseNames* defined in OPC 40083. The namespace index is server specific.	Mandatory
http://opcfoundation.org/UA/PlasticsRubber/Dosing/	Namespace for *NodeIds* and *BrowseNames* defined in this document. The namespace index is *Server* specific.	Mandatory
Vendor specific types	A *Server* may provide vendor-specific types like types derived from *ObjectTypes* defined in this document in a vendor-specific namespace.	Optional
Vendor specific instances	A *Server* provides vendor-specific instances of the standard types or vendor-specific instances of vendor-specific types in a vendor-specific namespace. It is recommended to separate vendor specific types and vendor specific instances into two or more namespaces.	Mandatory

Table 39 provides a list of namespaces and their indices used for *BrowseNames* in this document. The default namespace of this document is not listed since all *BrowseNames* without prefix use this default namespace.

Table 39 – Namespaces used in this document

NamespaceURI	Namespace Index	Example
http://opcfoundation.org/UA/	0	0:EngineeringUnits
http://opcfoundation.org/UA/DI/	2	2:DeviceClass
http://opcfoundation.org/UA/Machinery/	3	3:MachineIdentificaionType
http://opcfoundation.org/UA/PlasticsRubber/GeneralTypes/	4	4:MachineInformationType

Annex A
(normative)

OPC 40082-4 Namespace and mappings

A.1 NodeSet and supplementary files for OPC 40082-4 Information Model

The OPC 40082-4 *Information Model* is identified by the following URI:

http://opcfoundation.org/UA/PlasticsRubber/Dosing/

Documentation for the NamespaceUri can be found here.

The *NodeSet* associated with this version of specification can be found here:

https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/PlasticsRubber/Dosing/&v=1.0.0&i=1

The *NodeSet* associated with the latest version of the specification can be found here:

https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/PlasticsRubber/Dosing/&i=1

Supplementary files for the OPC 40082-4 *Information Model* can be found here:

https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/PlasticsRubber/Dosing/&v=1.0.0&i=2

The files associated with the latest version of the specification can be found here:

https://reference.opcfoundation.org/nodesets/?u=http://opcfoundation.org/UA/PlasticsRubber/Dosing/&i=2

_____