VDMA 40086-1

ICS 35.240.50; 83.200

Comments by 2023-03-01

OPC UA for Plastics and Rubber Machinery – Material Supply Systems – Part 1: Ordermanagement

OPC UA für Kunststoff- und Gummimaschinen – Materialversorgungssysteme – Teil 1: Auftragsverwaltung

VDMA 40086-1:2023-01 is identical with OPC 40086-1 (Release Candidate RC 1.00.01) and EUROMAP 86.1 (Release Candidate RC 1.00.01)

Application Warning Notice

This draft with date of issue 2022-11-25 is being submitted to the public for review and comment.

Because the final VDMA Specification may differ from this version, the application of this draft is subject to special agreement.

Comments are requested

– preferably as a file by e-mail to marc.schmitt@vdma.org

– or in paper form to VDMA e.V. Plastics and Rubber Machinery,
Lyoner Straße 18, 60528 Frankfurt.

Document comprises 47 pages

VDMA

Contents

Figures

Tables

OPC Foundation / VDMA

AGREEMENT OF USE

COPYRIGHT RESTRICTIONS

- This document is provided "as is" by the OPC Foundation and VDMA.
- 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 VDMA 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 VDMA 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 VDMA.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OPC or VDMA specifications may require use of an invention covered by patent rights. OPC Foundation or VDMA shall not be responsible for identifying patents for which a license may be required by any OPC or VDMA specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OPC or VDMA 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 VDMA 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 VDMA 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 VDMA 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 VDMA 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.

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

EUROMAP

EUROMAP represents around 1,000 companies manufacturing equipment for the plastics and rubber industry in the field of core machinery (pre-processing, converting, post- processing).

Member associations are located in Austria, France, Germany, Italy, Luxembourg, Spain, Switzerland, Turkey and the United Kingdom.

1 Scope

This document deals with the handling of granules, powder or liquids in a material supply system. Simple systems are considered, like bagged goods to loaders as well as highly complex systems with silos, pipes, dryers, mixers and dosing units.

It should be possible to order material mixtures (with specific properties) from an MES or downstream system to a specific transfer point (red marked). Furthermore, all process data generated during the process in the material supply system should be recorded and made available to other systems.

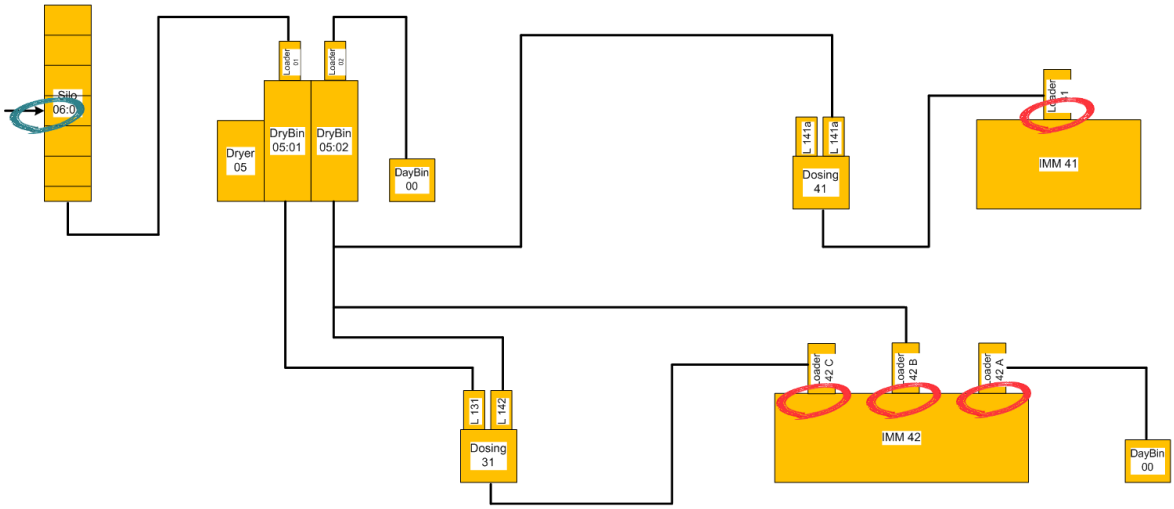

Figure 1 – Material Supply System Overview

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 10000-4, *OPC Unified Architecture - Part 4: Services*

http://www.opcfoundation.org/UA/Part4/

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/

3 Terms, definitions and conventions

3.1 Overview

It is assumed that basic concepts of OPC UA information modelling and are understood in this specification. This specification will use these concepts to describe the OPC 40086-1 Information Model. For the purposes of this document, the terms and definitions given in the documents references in Clause 2 apply.

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

3.2 OPC UA for OPC 40086-1 terms

3.2.1 MSS

Material Supply System - Complete system from the silo via mixer/dryer/intermediate silos/distributor/conveyor to the transfer point. The MSS has its own control system and coordinates/optimizes the internal processes.

3.2.2 MSSOutput

Describes the point from the MSS to the downstream system (injection moulding machine, extruder, ...).

3.2.3 MSSInput

Describes the point where material enters the MSS (filling pipe of the silo, ...).

3.2.4 Material

Material stands for a (raw) material that is managed in the MaterialList and can be used as an input to a device.

3.2.5 Recipe

Description of a material mixture.

3.2.6 ResultCode

The ResultCode gives information about each in-parameter of a method and is given by the MSS. It´s formatted in a binary code:

– 0: OK

– 1: NOK

3.2.7 ResultText

The ResultText is a additional information to the ResultCode. It is manufacturer dependent and optional given by the MSS.

3.2.8 LotId

Describes the unique ID of the produced material in the MSS when process relevant conditions are changed or modified. Examples for a new creation of a LotId are:

– Drying Temperature

– Drying Time

– Material

– New Supplier

– New Job

3.2.9 Jobs

The delivery of material by the MSS is organized by jobs. With a job a target machine (e.g. injection moulding machine) orders material according to a defined recipe which shall be provided at a specified transfer point. The order can also come from a MES which organizes the jobs for the different machines.

3.2.10 Transfer point

At the Transfer point the material is handed over to the downstream processing machine. There is a problem with traceability at the transfer point: On the one hand, the material flow at the transfer point can be discontinuous (batch part) or continuous. For this purpose, batch formation makes it possible to react to changes with volume, throughput and time and to create a new batch. This also reduces the amount of data to be transferred.

3.3 Conventions used in this document

The conventions described in OPC 40001-1 apply.

4 Use cases

OPC 40086-1 covers the following functionalities:

4.1 General information about the Material Supply System and its materials

The condition and process of the individual devices in the MSS are to be recorded (e.g. filling level, temperature, ...) Each of the devices has a material input (incl. condition) and a material output. A general basic model seems possible here, from which the various devices are derived.

4.2 Job management

The delivery of material by the MSS is organized by jobs. With a job a target machine (e.g. injection moulding machine) orders material according to a defined recipe which shall be provided at a specified transfer point. The order can also come from a MES which organizes the jobs for the different machines.

4.3 Traceability

The job represents the status of the MSS. It is important that the actual status is also recorded. For this purpose, it must be recorded when which material was in which condition and where. For this purpose, an event with a unique LotId is to be triggered for each delivered "batch part" (=unit of material that was transported to the transfer point by a conveyor) (not only a time stamp, as otherwise a time-consuming synchronization of all devices would be necessary).

Based on the LotId, the complete chain with delivery time, material mixing, drying times and temperatures of the materials, from which silo, which way through the MSS should be retrievable.

The topic of data storage, especially with small systems, must also be defined.

All devices involved (silo, dryer, mixer,...) should be listed and the processing of the materials (e.g. drying time/duration) should be included.

The time component is also important here, as the material remains in the devices for a certain time, can be stored in intermediate silos and transport also takes a certain amount of time.

An event is generated for each batch part that leaves the MSS. This contains a unique LotId, which can later be used to determine the history. The retrieval of the history is also to become part of the interface.

5 MSS_Interface Information Model overview

The MSS_Interface deals with the handling of granules, powder and liquids in a material supply system. Simple systems are considered, like bagged goods to loaders as well as highly complex systems with silos, pipes, dryers, mixers and dosing units.

It should be possible to order material mixtures (with specific properties) from an MES or downstream system to a specific transfer point. Furthermore, all process data generated during the process in the material supply system should be recorded and made available to other systems.

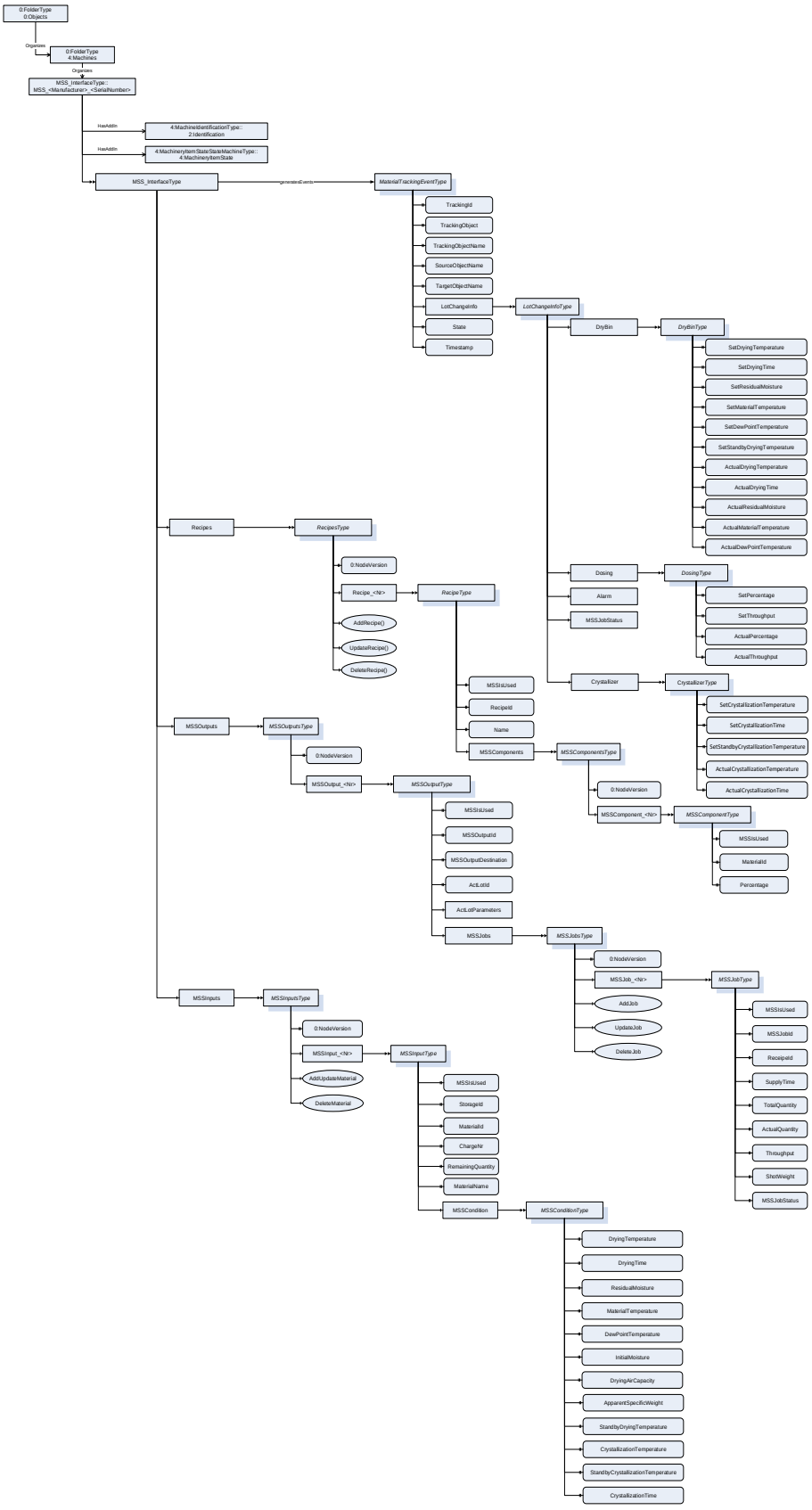

Figure 2 – MSS_InterfaceType Overview

5.1 MSS_InterfaceType ObjectType Definition

5.1.1 Overview

The *MSS_InterfaceType* provides the root Object representing a material supply system with all its subcomponents relevant for the interaction with downstream systems. The instance(s) of *MSS_InterfaceType* shall be located under the Object, which represents the MSS and which is directly located under the *Machines Object* of the Server and has an *Object Identification*, *MachieryItemState* and *MachineryOperationState* as defined in OPC 40001-1, and is formally defined in Table 1.

Table 1 – MSS_InterfaceType Definiton

Attribute	**Value**				
BrowseName	MSS_InterfaceType				
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:MachineIdentificationType	M
0:HasComponent	Object	3:MachineryBuildingBlocks		0:FolderType	M
0:HasComponent	Object	Recipes		RecipesType	M
0:HasComponent	Object	MSSOutputs		MSSOutputsType	M
0:HasComponent	Object	MSSInputs		MSSInputsType	O
0:GeneratesEvent	ObjectType	MaterialTrackingEventType			
Conformance Units					
OPC 40086-1 Entry Point					

The BrowseName of the instance shall be “MSS_<Manufacturer>_<SerialNumber>” → e.g. “MSS_motan_123”.

5.1.2 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 and the *Property ControllerName* is added.

The *Object MachineryBuildingBlocks* contains building blocks from OPC UA for Machinery as defined in OPC 40001-1. For this version of OPC 40086-1, the *Object* uses the two *AddIns MachineryItemState* and *MachineryOperationMode*.

Table 2 – MSS_InterfaceType 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
2:Identification	0:HasProperty	Variable	ControllerName	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		MachineryItemState_StateMa chineType	M
3:MachineryBuilding Blocks	0:HasAddIn	Object	3:MachineryOpe rationMode		3:MachineryOperationMode StateMachineType	M

The *ControllerName Property* represents the name of the machine controller (e.g. “CP22xx”).

5.1.3 Recipes

The *Recipes* provides all material mixtures of the MSS. The *RecipesType* is definded in 5.2.

5.1.4 MSSOutputs

The *MSSOutputs* provides the assignment of a downstream system to transfer points. The *MSSOutputsType* is definded in 5.6.

5.1.5 MSSInputs

The *MSSInputs* provides the assignment of incoming raw material to the MSS. The *MSSInputsType* is definded in 5.10.

5.1.6 MaterialTrackingEventType

The *MaterialTrackingEventType* provides the informations about the produced materials. The *MaterialTrackingEventType* is definded in 5.13.

5.2 RecipesType ObjectType Definition

5.2.1 Overview

The *RecipesType* provides the list of *RecipeType* Objects.

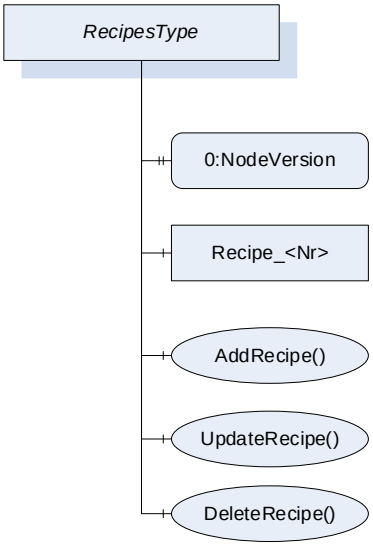

Figure 3 – RecipesType Overview

Table 3 – RecipesType Definiton

Attribute	**Value**				
BrowseName	RecipesType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	Recipe_<Nr>		RecipeType	MP
0:HasComponent	Method	AddRecipe			M
0:HasComponent	Method	UpdateRecipe			M
0:HasComponent	Method	DeleteRecipe			M
0:GeneratesEvent	ObjectType	0:GeneralModelChangeEventType			
Conformance Units					
OPC 40086-1 Recipes					

5.2.2 Recipe_<Nr>

The *Recipe_<Nr>* provides a placeholder for the recipes. When instances are created the *BrowseNames* shall be “Recipe_<Nr>” where <Nr> is a three-digit number with leading zeros, starting with “001”. The *RecipeType* is definded in 5.3.

5.2.3 AddRecipe

The *Method AddRecipe* adds a new recipe with the needed properties. The signature of this *Method* is specified below. Table 4 and Table 5 specify the *Arguments* and *AdressSpace* representation, respectively.

Signature

```
AddRecipe (
   [in]    0:String         RecipeName,
   [in]    0:String[]       MSSComponentsId,
   [in]    0:String[]       MSSComponentsPercentage,
   [out]   0:String         RecipeId,
   [out]   0:Int32          ResultCode,
   [out]   0:String         ResultText);
```

Table 4 – AddRecipe Method Arguments

Argument	Description
RecipeName	Is defined in 5.3.4
MSSComponentsId	An array of MaterialId of MSSComponents which is defined in 5.5.2
MSSComponentsPercentage	An array of Percentage of MSSComponents which is defined in 5.5.4
RecipeId	Is defined in 5.3.2
ResultCode	Is defined in 3.2.6
ResultText	Is defined in 3.2.7

Table 5 – AddRecipe Method AddressSpace Definiton

Attribute	Value				
BrowseName	AddRecipe				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
0:HasProperty	Variable	0:InputArguments	0:Argument[]	0:PropertyType	0:Mandatory
0:HasProperty	Variable	0:OutputArguments	0:Argument[]	0:PropertyType	0:Mandatory

Example for a ResultCode with 2 MaterialIds:

MSB → 001010 ← LSB:

LSB 0 → RecipeName is OK

1 → MSSComponentsIds[0] = 7; is NOK (e.g. MaterialId is not listed)

0 → MSSComponentsIds [1] = 2; is OK

1 → MSSComponentsPercentage [0] = 110; is NOK (e.g. Percentage > 100%)

0 → MSSComponentsPercentage [1] = 25; is OK

MSB 0 → RecipeId is OK

Note: The sum of percentages must be verified to 100%.

5.2.4 UpdateRecipe

The *Method UpdateRecipe* updates a recipe. The signature of this *Method* is specified below. Table 6 and Table 7 specify the *Arguments* and *AdressSpace* representation, respectively.

Signature

```
UpdateRecipe (
   [in]    0:String         RecipeId,
   [in]    0:String         RecipeName,
   [in]    0:String[]       MSSComponentsId,
   [in]    0:String[]       MSSComponentsPercentage,
   [out]   0:Int32          ResultCode,
   [out]   0:String         ResultText);
```

Table 6 – UpdateRecipe Method Arguments

Argument	Description
RecipeId	Is defined in 5.3.2
RecipeName	Is defined in 5.3.4
MSSComponentsId	An array of MaterialId of MSSComponents which is defined in 5.5.2
MSSComponentsPercentage	An array of Percentage of MSSComponents which is defined in 5.5.4
ResultCode	Is defined in 3.2.6
ResultText	Is defined in 3.2.7

Table 7 – UpdateRecipe Method AddressSpace Definiton

Attribute	Value				
BrowseName	UpdateRecipe				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
0:HasProperty	Variable	0:InputArguments	0:Argument[]	0:PropertyType	0:Mandatory
0:HasProperty	Variable	0:OutputArguments	0:Argument[]	0:PropertyType	0:Mandatory

Example for a ResultCode with 2 MaterialIds:

MSB → 011000 ← LSB: (Value = 24)

LSB 0 → RecipeId is OK

0 → RecipeName is OK

1 → MSSComponentsIds[0] = 7; is NOK (e.g. MaterialId is not listed)

0 → MSSComponentsIds[1] = 2; is OK

1 → MSSComponentsPercentage [0] = 110; is NOK (e.g. Percentage > 100%)

MSB 0 → MSSComponentsPercentage [1] = 25; is OK

Note: The sum of percentages must be verified to 100%.

5.2.5 DeleteRecipe

The *Method DeleteRecipe* deletes an existing recipe. The signature of this *Method* is specified below. Table 8 and Table 9 specify the *Arguments* and *AdressSpace* representation, respectively.

Signature

```
DeleteRecipe (
   [in]    0:String         RecipeId,
   [out]   0:Int32          ResultCode,
   [out]   0:String         ResultText);
```

Table 8 – DeleteRecipe Method Arguments

Argument	Description
RecipeId	Is defined in 5.3.2
ResultCode	Is defined in 3.2.6
ResultText	Is defined in 3.2.7

Table 9 – DeleteRecipe Method AddressSpace Definiton

Attribute	Value				
BrowseName	DeleteRecipe				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
0:HasProperty	Variable	0:InputArguments	0:Argument[]	0:PropertyType	0:Mandatory
0:HasProperty	Variable	0:OutputArguments	0:Argument[]	0:PropertyType	0:Mandatory

Example for a *ResultCode*:

MSB → 0 ← LSB:

0 → RecipeId is OK

5.3 RecipeType ObjectType Definition

5.3.1 Overview

The *RecipeType* provides the description of a material mixture specifying

– which materials (components)

– in which proportions

– in each case in which condition (degree of drying, residual moisture, ...)

are included.

Due to the desired condition, the recipe may contain requirements for different devices (e.g. drying temperature). However, it does not contain the exact dryer in which the material was treated. The assignment is made by the internal logic of the MSS and is formally defined in Table 10.

(Analogy: Cake recipe in which the ingredients and processing steps are described.)

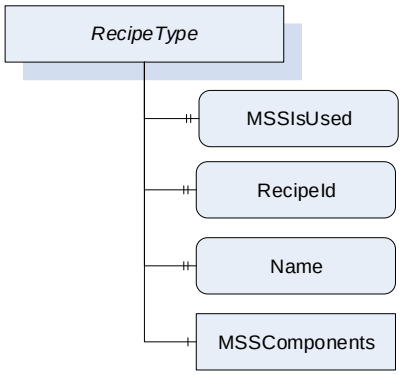

Figure 4 – RecipeType Overview

Table 10 – RecipeType Definiton

Attribute	**Value**				
BrowseName	RecipeType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	MSSIsUsed	0:Boolean	0:PropertyType	M, RO
0:HasProperty	Variable	RecipeId	0:String	0:PropertyType	M, RW
0:HasProperty	Variable	RecipeName	0:String	0:PropertyType	M, RW
0:HasComponent	Object	MSSComponents		MSSComponentsType	M
Conformance Units					
OPC 40086-1 Recipes					

5.3.2 MSSIsUsed

Is defined in 5.7.2

5.3.3 RecipeId

The *RecipeId* provides the unique Id of the recipe and is created by MSS during the method “AddRecipe”.

5.3.4 RecipeName

The Recipe*Name* provides the internal Name of the Recipe which is given by superordinate systems (e.g. ERP,MES).

5.3.5 MSSComponents

The *MSSComponents* provides the information of the components. The *MSSComponents*Type is definded in 5.4.

5.4 MSSComponentsType ObjectType Definition

Overview:

The *MSSComponentsType* provides a list for the material components and is formally defined in Table 11.

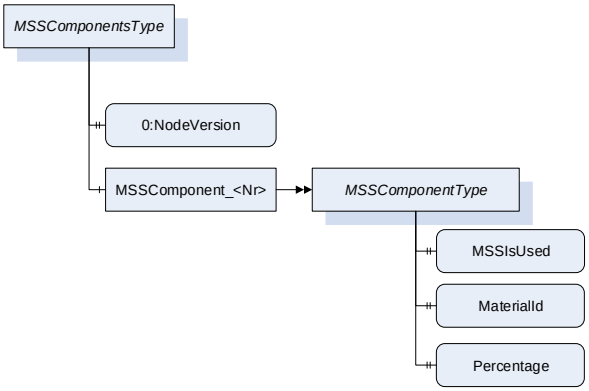

Figure 5 – MSSComponentsType Overview

Table 11 – MSSComponentsType Definiton

Attribute	**Value**				
BrowseName	MSSComponentsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	MSSComponent_<Nr>		MSSComponentType	MP
0:GeneratesEvent	ObjectType	0:GeneralModelChangeEventType			
Conformance Units					
OPC 40086-1 Components					

5.4.1 MSSComponent_<Nr>

The *MSSComponent_<Nr>* provides a placeholder for the components. When instances are created the BrowseNames shall be “MSSComponent_<Nr>” where <Nr> is a three-digit number with leading zeros, starting with “001”. The *MSSComponent*Type is definded in 5.5.

5.5 MSSComponentType ObjectType Definition

5.5.1 Overview

The *MSSComponentType* provides an additional information of the raw material and is formally defined in Table 12.

Table 12 – MSSComponentType Definiton

Attribute	Value				
BrowseName	MSSComponentType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	MSSIsUsed	0:Boolean	0:PropertyType	M, RO
0:HasProperty	Variable	MaterialId	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	Percentage	0:String	0:PropertyType	M, RO
Conformance Units					
OPC 40086-1 Components					

5.5.2 MSSIsUsed

Is defined in 5.7.2

5.5.3 MaterialId

The *MaterialId* provides the unique Id of the material and is given by superordinate systems (e.g. ERP,MES).

5.5.4 Percentage

The *Percentage* provides the proportions of the MSS Components and is given by the superordinate systems (e.g. ERP,MES).

5.6 MSSOutputsType ObjectType Definition

5.6.1 Overview

The *MSSOutputsType* provides information for all physical connections (Transfer Points) from the MSS to a downstream system and is formally defined in Table 13.

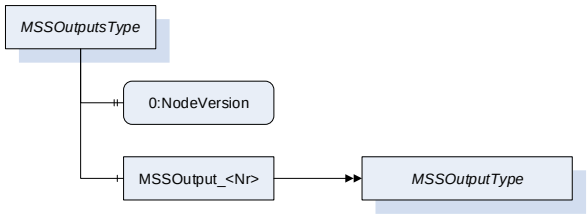

Figure 6 – MSSOutputsType Overview

Table 13 – MSSOutputsType Definiton

Attribute	Value				
BrowseName	MSSOutputsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	MSSOutput_<Nr>		MSSOutputType	MP
0:GeneratesEvent	ObjectType	0:GeneralModelChangeEventType			
Conformance Units					
OPC 40086-1 Outputs					

5.6.2 MSSOutput_<Nr>

The *MSSOutput_<Nr>* provides a placeholder for the physical connection (Transfer Point) from the MSS to a downstream system. When instances are created the BrowseNames shall be “MSSOutput _<Nr>” where <Nr> is a three-digit number with leading zeros, starting with “001”. The *MSSOutput*Type is definded in 5.7.

5.7 MSSOutputType ObjectType Definition

5.7.1 Overview

The *MSSOutputType* provides informations for the physical connection (Transfer Point) from the MSS to all downstream systems and is formally defined in Table 14.

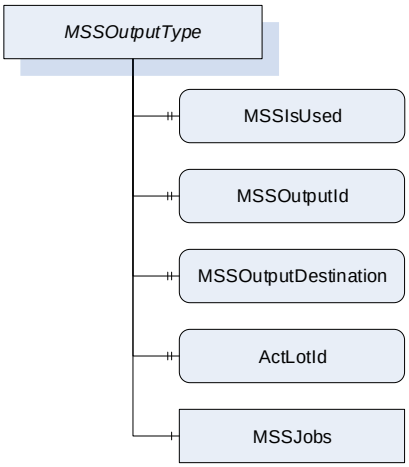

Figure 7 – MSSOutputType Overview

Table 14 – MSSOutputType Definiton

Attribute	**Value**				
BrowseName	MSSOutputType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	MSSIsUsed	0:Boolean	0:PropertyType	M, RO
0:HasProperty	Variable	MSSOutputId	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	MSSOutputDestination	0:String	0:PropertyType	M, RW
0:HasProperty	Variable	ActLotId	0:String	0:PropertyType	M, RO
0:HasComponent	Object	MSSJobs		MSSJobsType	M
Conformance Units					
OPC 40086-1 Outputs					

5.7.2 MSSIsUsed

The MSS*IsUsed* provides the information if the specific objecttype available and is given by the MSS.

5.7.3 MSSOutputId

The *MSSOutputId* provides the unique Id of the output transfer point of the MSS and is given by the MSS.

5.7.4 MSSOutputDestination

The *MSSOutputDestination* provides the name of the destination point where the material goes to and is given by the MES (e.g. component hopper of the machine).

5.7.5 ActLotId

The *ActLotId* provides the actual Id of the lot of the output transfer point and is given by the MSS.

5.7.6 MSSJobs

The *MSSJobs* provides job information for the MSS. The *MSSJobsType* is definded in 5.8.

5.8 MSSJobsType ObjectType Definition

5.8.1 Overview

The *MSSJobsType* provides job information for the MSS and is formally defined in Table 15. From the point of view of an downstream system (e.g.injection moulding machine), a job is an order placed with a machine. For example, the production order 1000 "mobile phone housings" can be divided among two injection moulding machines with 500 pieces each.

From the point of view of the MSS, a job is an "order" with the following information

– which material mixture (= which recipe is used)

– where (which transfer point)

– when

– in which quantity/throughput

should be made available.

The job can come from the downstream system or from an MES. It is readable from all connected devices.

An existing job can also be changed.

Jobs are grouped below an MSS output. This means that each connected downstream system can recognize the jobs that affect it.

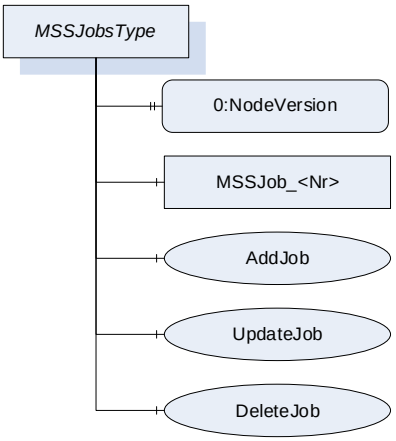

Figure 8 – MSSJobsType Overview

Table 15 – MSSJobsType Definiton

Attribute	Value				
BrowseName	MSSJobsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	MSSJob_<Nr>		MSSJobType	OP
0:HasComponent	Method	AddJob			M
0:HasComponent	Method	UpdateJob			M
0:HasComponent	Method	DeleteJob			M
0:GeneratesEvent	ObjectType	0:GeneralModelChangeEventType			
Conformance Units					
OPC 40086-1 Jobs					

5.8.2 MSSJob_<Nr>

The *MSSJob_<Nr>* provides a placeholder for the Jobs. When instances are created the BrowseNames shall be “MSSJob _<Nr>” where <Nr> is a three-digit number with leading zeros, starting with “001”. The *MSSJobType* is definded in 5.9.

5.8.3 AddJob

The *Method AddJob* provides to add a job on the MSS. The signature of this *Method* is specified below. Table 16 and Table 17 specify the *Arguments* and *AdressSpace* representation, respectively. If a job is created, the MSS creates a job instance and returns its ID (also generated by the MSS). The status may contain error messages (e.g. recipe unknown). ActualQuantity displays the quantity already delivered.

Signature

```
AddJob (
   [in]    0:String          RecipeId,
   [in]    0:DateTime        SupplyTime,
   [in]    0:Double          TotalQuantity,
   [in]    0:Double          Throughput,
   [in]    0:Double          ShotWeight,
   [in]    0:String[]        MSSOutput,
   [out]   0:String          MSSJobId,
   [out]   0:Int32           ResultCode,
   [out]   0:String          ResultText);
```

Table 16 – AddJob Method Arguments

Argument	Description
RecipeId	Is defined in 5.3.2
SupplyTime	Is defined in 5.9.5, DateTime in UTC
TotalQuantity	Is defined in 5.9.6, unit [kg]
Throughput	Is defined in 5.9.8, unit [kg/h]
ShotWeight	Is defined in 5.9.9, unit [kg]
MSSOutput	Is defined in 5.6.2
MSSJobId	Is defined in 5.9.3
ResultCode	Is defined in 3.2.6
ResultText	Is defined in 3.2.7

Table 17 – AddJob Method AddressSpace Definiton

Attribute	Value				
BrowseName	AddJob				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
0:HasProperty	Variable	0:InputArguments	0:Argument[]	0:PropertyType	0:Mandatory
0:HasProperty	Variable	0:OutputArguments	0:Argument[]	0:PropertyType	0:Mandatory

Example for a ResultCode (For each input-parameter of the method, there is a corresponding bit in the ResultCode):

MSB → 001010 ← LSB: (Value = 10)

LSB 0 → RecipeId is OK

1 → SupplyTime is NOK (maybe in the past)

0 → TotalQuantity is OK

1 → Throughput is NOK (maybe to high for the MSS)

0 → ShotWeight is OK

MSB 0 → MSSOutput_<Nr> is OK

Note: Detailed errorinformation has to be returned by the OPC UA Statuscode.

5.8.4 UpdateJob

The *Method UpdateJob* can update or change the job information (e.g. shift of the start time, increase of the total quantity). The signature of this *Method* is specified below. Table 18 and Table 19 specify the *Arguments* and *AdressSpace* representation, respectively.

Signature

```
UpdateJob (
   [in]    0:String         MSSJobId,
   [in]    0:String         RecipeId,
   [in]    0:DateTime       SupplyTime,
   [in]    0:Double         TotalQuantity,
   [in]    0:Double         Throughput,
   [in]    0:Double         ShotWeight,
   [in]    0:String[]       MSSOutput,
   [out]   0:Int32          ResultCode,
   [out]   0:String         ResultText);
```

Table 18 – UpdateJob Method Arguments

Argument	Description
MSSJobId	Is defined in 5.9.3
RecipeId	Is defined in 5.3.2
SupplyTime	Is defined in 5.9.5, DateTime in UTC
TotalQuantity	Is defined in 5.9.6, unit [kg]
Throughput	Is defined in 5.9.8, unit [kg/h]
ShotWeight	Is defined in 5.9.9, unit [kg]
MSSOutput	Is defined in 5.6.2
ResultCode	Is defined in 3.2.6
ResultText	Is defined in 3.2.7

Table 19 – UpdateJob Method AddressSpace Definiton

Attribute	Value				
BrowseName	UpdateJob				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
0:HasProperty	Variable	0:InputArguments	0:Argument[]	0:PropertyType	0:Mandatory
0:HasProperty	Variable	0:OutputArguments	0:Argument[]	0:PropertyType	0:Mandatory

Example for a ResultCode:

MSB → 1000010 ← LSB: (Value = 66)

LSB 0 → MSSJobId is OK

1 → RecipeId is NOK (maybe this RecipeId is not in the MSS)

0 → SupplyTime is OK

0 → TotalQuantity is OK

0 → Throughput is OK

0 → ShotWeight is OK

MSB 1 → MSSOutput_<Nr> is NOK (maybe the TransferPoint is not defiened in the MSS)

5.8.5 DeleteJob

The *Method DeleteJob* can delete Jobs. The signature of this *Method* is specified below. Table 20 and Table 21 specify the *Arguments* and *AdressSpace* representation, respectively.

Signature

```
DeleteJob (
   [in]    0:String         MSSJobId,
   [out]   0:Int32          ResultCode,
   [out]   0:String         ResultText);
```

Table 20 – DeleteJob Method Arguments

Argument	Description
MSSJobId	Is defined in 5.9.3
ResultCode	Is defined in 3.2.6
ResultText	Is defined in 3.2.7

Table 21 – DeleteJob Method AddressSpace Definiton

Attribute	Value				
BrowseName	DeleteJob				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
0:HasProperty	Variable	0:InputArguments	0:Argument[]	0:PropertyType	0:Mandatory
0:HasProperty	Variable	0:OutputArguments	0:Argument[]	0:PropertyType	0:Mandatory

Example for a ResultCode:

MSB → 1 ← LSB: (Value = 1)

1 → MSSJobId is NOK (maybe the Job is not added on the MSS)

5.9 MSSJobType ObjectType Definition

5.9.1 Overview

The *MSSJobType* provides all job relevant informations and is formally defined in Table 22.

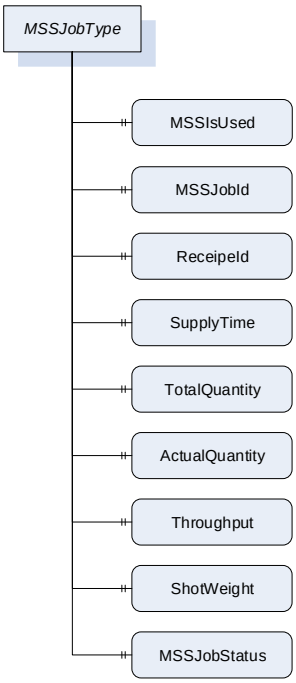

Figure 9 – MSSJobType Overview

Table 22 – MSSJobType Definiton

Attribute	**Value**				
BrowseName	MSSJobType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	MSSIsUsed	0:Boolean	0:PropertyType	M, RW
0:HasProperty	Variable	MSSJobId	0:String	0:PropertyType	M, RW
0:HasProperty	Variable	RecipeId	0:String	0:PropertyType	M, RW
0:HasProperty	Variable	SupplyTime	0:DateTime	0:PropertyType	M, RW
0:HasProperty	Variable	TotalQuantity	0:Double	0:PropertyType	M, RW
0:HasProperty	Variable	ActualQuantity	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	Throughput	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	ShotWeight	0:Double	0:PropertyType	O, RW
0:HasComponent	Variable	MSSJobStatus	0:UInt16	0:MultiStateValueDiscreteType	M, RW
Conformance Units					
OPC 40086-1 Jobs					

The child *Node MSSJobStatus* of the *MSSJobType* has additional *Attribute* values defined in Table 23.

Table 23 – *MSSJobType* Attribute values for child Nodes

BrowsePath	Value Attribute	Description Attribute
MSSJobStatus 0:EnumValues	{{0, OTHER, This state is used if none of the other states below apply. Set by operator. Status is INACTIVE}, {1, JOB_IN_PREPERATION, The MSS is preparing material for the job. Status is ACTIVE}, {2, JOB_IN_PRODUCTION, The MSS is processing material for the job. Status is ACTIVE}, {3, JOB_INTERRUPTED, The job is interrupted, but can continue after (paused). The nominal output is not reached. Status is ACTIVE}, {4, JOB_FINISHED, Nominal output reached. Status is INACTIVE}, {5, JOB_CANCELLED, The job is cancelled. Status is INACTIVE}}	-

5.9.2 MSSIsUsed

Is defined in 5.7.2

5.9.3 MSSJobId

The MSS*JobId* provides the unique Id of the Job and is created by MSS during the method “AddJob”.

5.9.4 RecipeId

Is defined in 5.3.2

5.9.5 SupplyTime

The *SupplyTime* provides the exact delivery time of the material at the MSSOutput (Transfer Point) and is created by superordinate systems (e.g. ERP,MES) during the method “AddJob” or “UpdateJob”.

5.9.6 TotalQuantity

The *TotalQuantity* provides the total required quantity of material for the job and is created by superordinate systems (e.g. ERP,MES) during the method “AddJob” or “UpdateJob”. The unit is [kg].

5.9.7 ActualQuantity

The *ActualQuantity* provides the actual produced quantity of material for the job and is given by the MSS.

5.9.8 Throughput

The *Throughput* provides the required throughput of the material for the job and is created by superordinate systems (e.g. ERP,MES) during the method “AddJob” or “UpdateJob”.

5.9.9 ShotWeight

The *ShotWeight* provides the total weight per moulding cycle and is created by superordinate systems (e.g. ERP,MES) during the method “AddJob” or “UpdateJob”.

5.9.10 MSSJobStatus

The *MSSJobStatus* provides the the current status of the job and is given by the MSS. The *TypeDefinition* is *MultiStateValueDiscreteType*, so the *Properties EnumValues* and *ValueAsText* must be filled with the supported values out of Table 24.

Table 24 – Values for MSSJobStatus

EnumValue	ValueAsText	Description
0	OTHER	This state is used if none of the other states below apply. Set by operator. Status is **INACTIVE**
1	JOB_IN_PREPERATION	The MSS is peparing material for the job. Status is **ACTIVE**
2	JOB_IN_PRODUCTION	The MSS is processing material for the job. Status is **ACTIVE**
3	JOB_INTERRUPTED	The job is interrupted, but can continue after (paused). The nominal output is not reached. Status is **ACTIVE**
4	JOB_FINISHED	Nominal output reached. Status is **INACTIVE**
5	JOB_CANCELLED	The job is cancelled. Status is **INACTIVE**

5.10 MSSInputsType ObjectType Definition

5.10.1 Overview

The *MSSInputsType* describes an input point of the MSS for material. With the method *AddMaterial()* you can add a Material with *MaterialId*, *ChargeNr* and *TotalQuantity* from your MES or ERP of the delivered product. You also get the information of the *RemainingQuantity* of the MSS back (e.g. reason of material loss during start-up, scrap, etc.). and is formally defined in Table 25.

Note: The *MSSInputsType* should only be used, in case of a connection to a superordinate system like a ERP or MES.

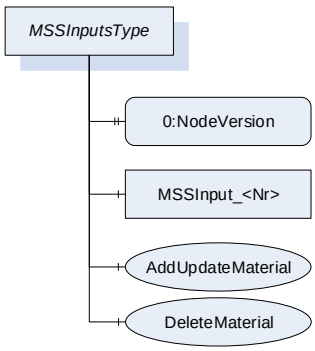

Figure 10 – MSSInputsType Overview

Table 25 – MSSInputsType Definiton

Attribute	Value				
BrowseName	MSSInputsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	0:NodeVersion	0:String	0:PropertyType	M, RO
0:HasComponent	Object	MSSInput_<Nr>		MSSInputType	OP
0:HasComponent	Method	AddUpdateMaterial			O
0:HasComponent	Method	DeleteMaterial			O
0:GeneratesEvent	ObjectType	0:GeneralModelChangeEventType			
Conformance Units					
OPC 40086-1 Inputs					

5.10.2 MSSInput _<Nr>

The *MSSInput_<Nr>* provides a placeholder for the physical connection from a truck, bag, etc. to the MSS. When instances are created the BrowseNames shall be “MSSInput _<Nr>” where <Nr> is a three-digit number with leading zeros, starting with “001”. The *MSSInputType* is definded in 5.11.

5.10.3 AddUpdateMaterial

The *Method AddUpdateMaterial* provides the possibility to add a material in the stock of the MSS. The signature of this *Method* is specified below. Table 26 and Table 28 specify the *Arguments* and *AdressSpace* representation, respectively.

Signature

```
AddUpdateMaterial (
   [in]    0:String                 MaterialId,
   [in]    0:String                 StorageId,
   [in]    0:String                 ChargeNr,
   [in]    0:Double                 TotalQuantity,
   [in]    0:String                 MaterialName,
   [in]    MSSConditionStructType   MSSCondition,
   [out]   0:Int32                  ResultCode,
   [out]   0:String                 ResultText);
```

Table 26 – AddUpdateMaterial Method Arguments

Argument	Description
MaterialId	Is defined in 5.5.2
StorageId	Is defined in 5.11.2
ChargeNr	Is defined in 5.11.5
TotalQuantity	Is defined in 5.9.6, unit [kg]
MaterialName	Is defined in 5.11.7
MSSCondition	Is defined in Table 27
ResultCode	Is defined in 3.2.6
ResultText	Is defined in 3.2.7

Table 27 – MSSConditionStructType Definition

Name	Type	Description
MSSConditionStructType	structure	Subtype of 0:*Structure* as defined in OPC UA 10000-3
DryingTemperature	0:Double	Is defined in 5.12.2, unit [°C]
DryingTime	0:Duration	Is defined in 5.12.3, unit [ms]
ResidualMoisture	0:Double	Is defined in 5.12.4, unit [ppm]
MaterialTemperature	0:Double	Is defined in 5.12.5, unit [°C]
DewPointTemperature	0:Double	Is defined in 5.12.6, unit [°C]
InitialMoisture	0:Double	Is defined in 5.12.7, unit [ppm]
DryingAirCapacity	0:Double	Is defined in 5.12.8, unit [m³/h/kg]

Table 28 – AddUpdateMaterial Method AddressSpace Definiton

Attribute	Value				
BrowseName	AddUpdateMaterial				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
0:HasProperty	Variable	0:InputArguments	0:Argument[]	0:PropertyType	0:Mandatory
0:HasProperty	Variable	0:OutputArguments	0:Argument[]	0:PropertyType	0:Mandatory

Example for a ResultCode:

MSB → 100000 ← LSB: (Value = 32)

LSB 0 → MaterialId is OK

0 → StorageId is OK

0 → ChargeNr is OK

0 → TotalQuantity is OK

0 → MaterialName is OK

MSB 1 → MSSCondition is NOK

Note: Variables that are used in methods but are marked optional in MSSInputType can be filled with NULL or -999999999 here if they are not to be used! To update the material you need MaterialId, StorageId and ChargeNr as primary keys.

5.10.4 DeleteMaterial

The *Method DeleteMaterial* provides the possibility to delete a material in the stock of the MSS. The signature of this *Method* is specified below. Table 26 and Table 28 specify the *Arguments* and *AdressSpace* representation, respectively.

Signature

```
DeleteMaterial (
   [in]    0:String        MaterialId,
   [in]    0:String        StorageId,
   [in]    0:String        ChargeNr,
   [out]   0:Int32         ResultCode,
   [out]   0:String        ResultText);
```

Table 29 – DeleteMaterial Method Arguments

Argument	Description
MaterialId	Is defined in 5.5.2
StorageId	Is defined in 5.11.2
ChargeNr	Is defined in 5.11.5
ResultCode	Is defined in 3.2.6
ResultText	Is defined in 3.2.7

Table 30 – DeleteMaterial Method AddressSpace Definiton

Attribute	Value				
BrowseName	DeleteMaterial				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
0:HasProperty	Variable	0:InputArguments	0:Argument[]	0:PropertyType	0:Mandatory
0:HasProperty	Variable	0:OutputArguments	0:Argument[]	0:PropertyType	0:Mandatory

Example for a ResultCode:

MSB → 100 ← LSB:

LSB 0 → MaterialId is OK

0 → StorageId is OK

MSB 1 → ChargeNr is NOK

Note: To delete the material you need MaterialId, StorageId and ChargeNr as primary keys.

5.11 MSSInputType ObjectType Definition

5.11.1 Overview

The *MSSInputType* provides the physical connection from a truck, bag, etc. to the MSS and is formally defined in Table 31.

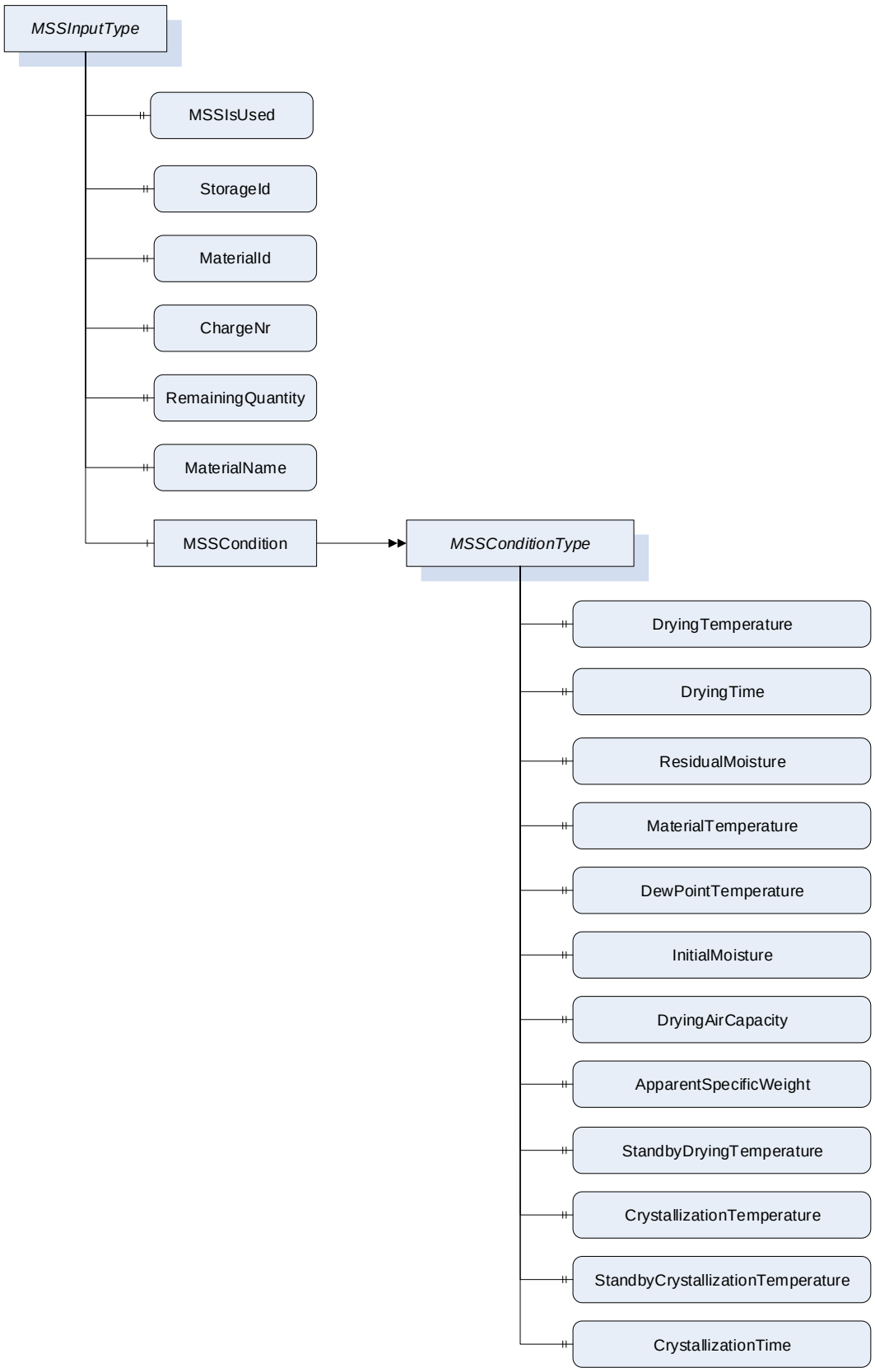

Figure 11 – MSSInputType Overview

Table 31 – MSSInputType Definiton

Attribute	Value				
BrowseName	MSSInputType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	MSSIsUsed	0:Boolean	0:PropertyType	M, RO
0:HasProperty	Variable	StorageId	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	MaterialId	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	ChargeNr	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	RemainingQuantity	0:Double	0:PropertyType	M, RW
0:HasProperty	Variable	MaterialName	0:String	0:PropertyType	M, RO
0:HasComponent	Object	MSSCondition		MSSConditionType	M
Conformance Units					
OPC 40086-1 Inputs					

5.11.2 MSSIsUsed

Is defined in 5.7.2.

5.11.3 StorageId

The *StorageId* provides the unique Id of the internal material storage of the MSS and is given by the MSS.

5.11.4 MaterialId

Is defined in 5.5.2.

5.11.5 ChargeNr

The *ChargeNr* provides the unique Charge of a material which is given by superordinate systems (e.g. ERP,MES). It is a combination of the *ManufacturerId* and *ChargeNr* (e.g. 123456_100) of the superordinate system.

5.11.6 RemainingQuantity

The *RemainingQuantity* provides the remaining quantity of the internal material storage of the MSS and is given by the MSS.

5.11.7 MaterialName

The *MaterialName* provides the internal Name of the Material and is given by the superordinate System.

5.11.8 MSSCondition

The *MSSCondition* provides an additional information of the raw material. The *MSSConditionType* is definded in 5.12.

5.12 MSSConditionType ObjectType Definition

5.12.1 Overview

The *MSSConditionType* provides additional information of the raw material and is formally defined in Table 32.

Table 32 – MSSConditionType Definiton

Attribute	Value				
BrowseName	MSSConditionType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	DryingTemperature	0:Double	0:PropertyType	M, RW
0:HasProperty	Variable	DryingTime	0:Duration	0:PropertyType	M, RW
0:HasProperty	Variable	ResidualMoisture	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	MaterialTemperature	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	DewPointTemperature	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	InitialMoisture	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	DryingAirCapacity	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	ApparentSpecificWeight	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	StandbyDryingTemperature	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	CrystallizationTemperature	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	StandbyCrystallizationTemperature	0:Double	0:PropertyType	O, RW
0:HasProperty	Variable	CrystallizationTime	0:Duration	0:PropertyType	O, RW
Conformance Units					
OPC 40086-1 Material Conditions					

5.12.2 DryingTemperature

The *DryingTemperature* provides the drying temperature of the raw material and is given by the superordinate system (e.g. MES, ERP, …).

5.12.3 DryingTime

The *DryingTime* provides the drying time of the raw material and is given by the superordinate system (e.g. MES, ERP, …).

5.12.4 ResidualMoisture

The *ResidualMoisture* provides the residual moisture of the material material and is given by the superordinate system (e.g. MES, ERP, …). (currently not available)

5.12.5 MaterialTemperature

The *MaterialTemperature* provides the temperature of the material and is given by the superordinate system (e.g. MES, ERP, …). (currently not available).

5.12.6 DewPointTemperature

The *DewPointTemperature* provides the measured dewpoint temperature of the air before the heating system and is given by MSS.

5.12.7 InitialMoisture

The *InitialMoisture* provides the moisture of the material which is to be fed to the MSS and is given by the superordinate system (e.g. MES, ERP, …).

5.12.8 DryingAirCapacity

The *DryingAirCapacity* provides the airflow to dry the material and is given by the superordinate system (e.g. MES, ERP, …).

5.12.9 ApparentSpecificWeight

The *ApparentSpecificWeight* is the bulk density of the material and is given by the superordinate system (e.g. MES, ERP, …). [kg/m³]

5.12.10 StandbyDryingTemperature

In case the dried material will stay in the drying bin after the drying process, the *StandbyDryingTemperature* has the task to retain the material quality and is given by the superordinate system (e.g. MES, ERP, …). [°C]

5.12.11 CrystallizationTemperature

The *CrystallizationTemperature* provides the crystallisation temperature of the material and is given by the superordinate system (e.g. MES, ERP, …). [°C]

5.12.12 StandbyCrystallizationTemperature

In case the crystalized material will stay in the crystallisation bin after the crystallisation process, the *StandbyCrystallizationTemperature* has the task to retain the material quality and is given by the superordinate system (e.g. MES, ERP, …). [°C]

5.12.13 CrystallizationTime

The *CrystallizationTime* provides the crystallisation time of the material and is given by the superordinate system (e.g. MES, ERP, …). [ms]

5.13 MaterialTrackingEventType ObjectType Definition

5.13.1 Overview

The *MaterialTrackingEventType* provides all process relevant information of the material for a job. The data can be stored in a database in a superordinate system to make it available later and to detect any material problems in the downstream process. It is formally defined in Table 33.

Rules to fire the event:

– Process relevant changes

– MSSJobStatus changed

– LotId changed

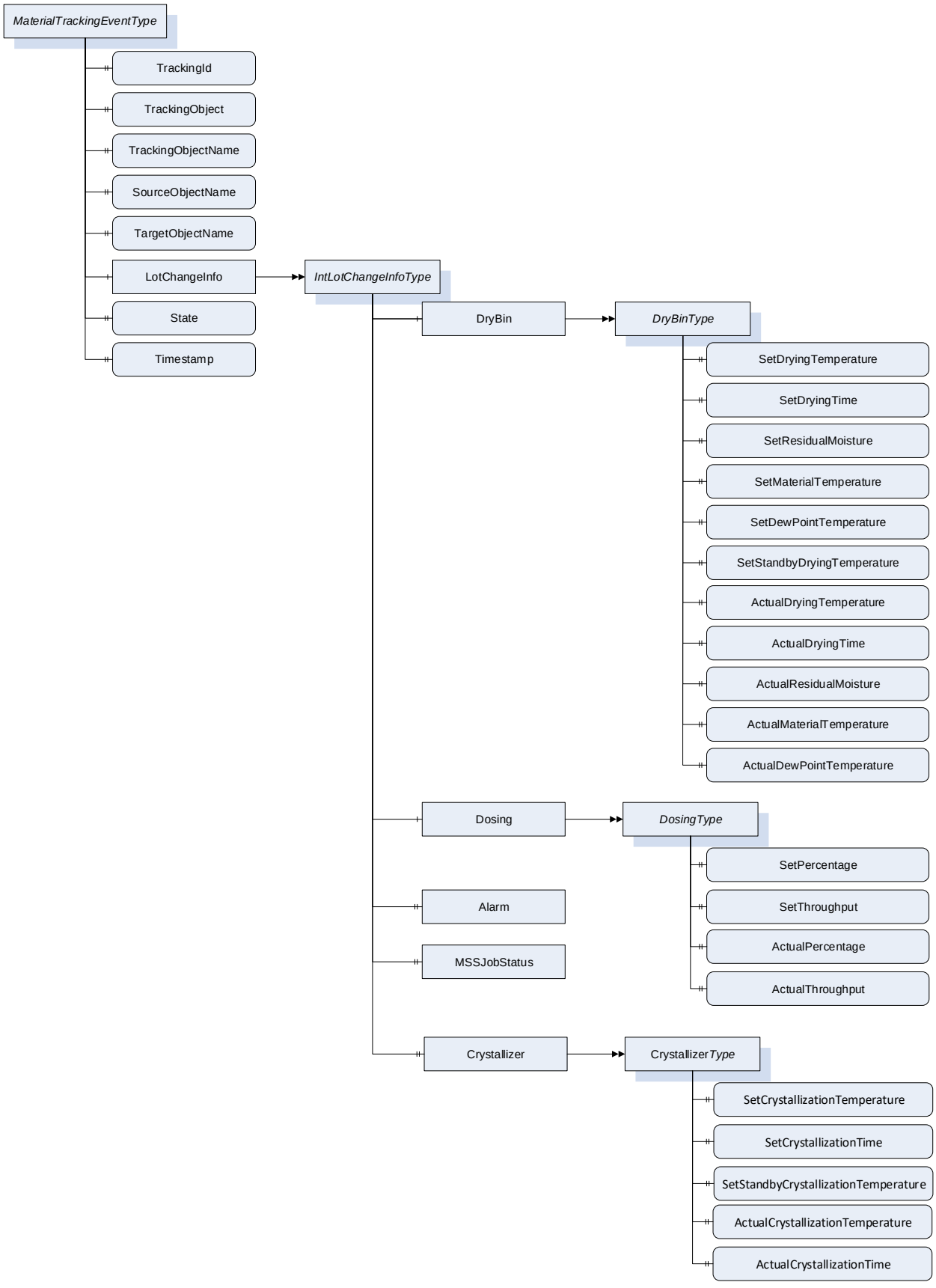

Figure 12 – MaterialTrackingEventType Overview

Table 33 – MaterialTrackingEventType Definiton

Attribute	Value				
BrowseName	MaterialTrackingEventType				
IsAbstract	True				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseEventType* defined in OPC 10000-5					
0:HasProperty	Variable	TrackingId	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	TrackingObject	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	TrackingObjectName	0:String	0:PropertyType	M, RO
0:HasProperty	Variable	SourceObjectName	0:String[]	0:PropertyType	M, RO
0:HasProperty	Variable	TargetObjectName	0:String[]	0:PropertyType	M, RO
0:HasComponent	Object	LotChangeInfo		LotChangeInfoType	M
0:HasProperty	Variable	State	0:Boolean	0:PropertyType	M, RO
0:HasProperty	Variable	Timestamp	0:DateTime	0:PropertyType	M, RO
Conformance Units					
OPC 40086-1 Material Tracking					

5.13.2 TrackingId

The *TrackingId* provides the unique Id of the object which fires an event and is given by the MSS.

5.13.3 TrackingObject

The *TrackingObject* provides a unique Id of the device and is given by the MSS

5.13.4 TrackingObjectName

The *TrackingObjectName* provides a unique Name of the device and is given by the MSS

5.13.5 SourceObjectName

The *SourceObjectName* provides a direct predecessor list and is given by the MSS

5.13.6 TargetObjectName

The *TargetObjectName* provides a direct successor list and is given by the MSS

5.13.7 LotChangeInfo

The *LotChangeInfo* provides the event reason and is given by the MSS. The *LotChangeInfoType* is definded in 5.14.

5.13.8 State

The *State* provides the material quality state (ok=true; nok=false) of the tracking object and is given by the MSS

5.13.9 Timestamp

The *Timestamp* provides actual time and date of the tracking event (in UTC) and is given by the MSS

5.14 LotChangeInfoType ObjectType Definition

5.14.1 Overview

The *LotChangeInfoType* provides optional information of the tracking object and is given by the MSS and is formally defined in Table 34.

Table 34 – LotChangeInfoType

Attribute	Value				
BrowseName	LotChangeInfoType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasComponent	Object	DryBin		DryBinType	O
0:HasComponent	Object	Dosing		DosingType	O
0:HasProperty	Variable	Alarm	0:String	0:PropertyType	O, RO
0:HasComponent	Variable	MSSJobStatus	0:UInt16	0:MultiStateValueDiscreteType	M, RO
0:HasComponent	Object	Crystallizer		CrystallizerType	O
Conformance Units					
OPC 40086-1 Material Tracking					

The child *Node MSSJobStatus* of the *LotChangeInfoType* has additional *Attribute* values defined in Table 35.

Table 35 – *MSSJobType* Attribute values for child Nodes

BrowsePath	Value Attribute	Description Attribute
MSSJobStatus 0:EnumValues	{{0, OTHER, This state is used if none of the other states below apply. Set by operator. Status is INACTIVE}, {1, JOB_IN_PREPERATION, The MSS is preparing material for the job. Status is ACTIVE}, {2, JOB_IN_PRODUCTION, The MSS is processing material for the job. Status is ACTIVE}, {3, JOB_INTERRUPTED, The job is interrupted, but can continue after (paused). The nominal output is not reached. Status is ACTIVE}, {4, JOB_FINISHED, Nominal output reached. Status is INACTIVE}, {5, JOB_CANCELLED, The job is cancelled. Status is INACTIVE}}	-

5.14.2 DryBin

The *DryBin* provides optional information of the tracking object and is given by the MSS. The *DryBinType* is definded in 5.15.

5.14.3 Dosing

The *Dosing* provides optional information of the tracking object and is given by the MSS. The *DosingType* is definded in 5.16.

5.14.4 Alarm

The *Alarm* provides optional information of the tracking object and is given by the MSS. The variable *Alarm* is a vendor specific information of a process relevant change.

5.14.5 MSSJobStatus

The *MSSJobStatus* is defined in 5.9.10

5.14.6 Crystallizer

The *Crystallizer* provides optional information of the tracking object and is given by the MSS. The *CrystallizerType* is definded in 5.17.

5.15 DryBinType ObjectType Definition

5.15.1 Overview

The *DryBinType* provides optional information of the tracking object and is given by the MSS and is formally defined in Table 36.

Table 36 – DryBinType

Attribute	**Value**				
BrowseName	DryBinType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	SetDryingTemperature	0:Double	0:PropertyType	M, RO
0:HasProperty	Variable	SetDryingTime	0:Duration	0:PropertyType	M, RO
0:HasProperty	Variable	SetResidualMoisture	0:Double	0:PropertyType	O, RO
0:HasProperty	Variable	SetMaterialTemperature	0:Double	0:PropertyType	O, RO
0:HasProperty	Variable	SetDewPointTemperature	0:Double	0:PropertyType	O, RO
0:HasProperty	Variable	SetStandbyDryingTemperature	0:Double	0:PropertyType	O, RO
0:HasProperty	Variable	ActualDryingTime	0:Duration	0:PropertyType	M, RO
0:HasProperty	Variable	ActualDryingTemperature	0:Double	0:PropertyType	M, RO
0:HasProperty	Variable	ActualResidualMoisture	0:Double	0:PropertyType	O, RO
0:HasProperty	Variable	ActualMaterialTemperature	0:Double	0:PropertyType	O, RO
0:HasProperty	Variable	ActualDewPointTemperature	0:Double	0:PropertyType	O, RO
Conformance Units					
OPC 40086-1 Material Tracking					

5.15.2 SetDryingTemperature

The *SetDryingTemperature* provides the dying temperature and is given by the *MSSCondition* Type, see 5.12.2.

5.15.3 SetDryingTime

The *SetDryingTime* provides the desired drying time and is given by the *MSSCondition* Type, see 5.12.3.

5.15.4 SetResidualMoisture

The *SetResidualMoisture* provides the desired residual moisture and is given by the *MSSCondition* Type, see 5.12.4.

5.15.5 SetMaterialTemperature

The *SetMaterialTemperature* provides the desired material temperature and is given by the *MSSCondition* Type, see 5.12.5.

5.15.6 SetDewPointTemperature

The *SetDewPointTemperature* provides the desired dew point temperature and is given by the *MSSCondition* Type, see 5.12.6.

5.15.7 SetStandbyDryingTemperature

The *SetStandbyDryingTemperature* provides the desired standby temperature and is given by the *MSSConditionType*, see 5.12.10

5.15.8 ActualDryingTemperature

The *ActualDryingTemperature* provides the actual drying temperature and is given by the MSS.

5.15.9 ActualDryingTime

The *ActualDryingTime* provides the actual drying time and is given by the MSS.

5.15.10 ActualResidualMoisture

The *ActualResidualMoisture* provides the actual residual moisture and is given by the MSS.

5.15.11 ActualMaterialTemperature

The *ActualMaterialTemperature* provides the actual material temperature and is given by the MSS.

5.15.12 ActualDewPointTemperature

The *ActualDewPointTemperature* provides the actual dew point temperature and is given by the MSS.

5.16 DosingType ObjectType Definition

5.16.1 Overview

The *DosingType* provides optional information of the tracking object and is given by the MSS and is formally defined in Table 37.

Table 37 – DosingType

Attribute	**Value**				
BrowseName	DosingType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	SetPercentage	0:String	0:PropertyType	M,RO
0:HasProperty	Variable	SetThroughput	0:Double	0:PropertyType	M,RO
0:HasProperty	Variable	ActualPercentage	0:String	0:PropertyType	M,RO
0:HasProperty	Variable	ActualThroughput	0:Double	0:PropertyType	M,RO
Conformance Units					
OPC 40086-1 Material Tracking					

5.16.2 SetPercentage

The *SetPercentage* provides the desired settings of percentage of the material of the mixture and is given by the *MSSComponent* Type, see 5.5.4.

5.16.3 SetThroughput

The *SetThroughput* provides the desired throughput of the material and is given by the *MSSJob* Type, see 5.9.8.

5.16.4 ActualPercentage

The *ActualPercentage* provides the actual settings of percentage of the material of the mixture and is given by the MSS.

5.16.5 ActualThroughput

The *ActualThroughput* provides the actual throughput of the material and is given by the MSS.

5.17 CrystallizerType ObjectType Definition

5.17.1 Overview

The *CrystallizerType* provides optional information of the tracking object and is given by the MSS and is formally defined in Table 38.

Table 38 – CrystallizerType

Attribute	**Value**				
BrowseName	CrystallizerType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the 0:*BaseObjectType* defined in OPC 10000-5					
0:HasProperty	Variable	SetCrystallizationTemperature	0:Double	0:PropertyType	M,RO
0:HasProperty	Variable	SetCrystallizationTime	0:Duration	0:PropertyType	M,RO
0:HasProperty	Variable	SetStandbyCrystallizationTemperature	0:Double	0:PropertyType	M,RO
0:HasProperty	Variable	ActualCrystallizationTemperature	0:Double	0:PropertyType	M,RO
0:HasProperty	Variable	ActualCrystallizationTime	0:Duration	0:PropertyType	M,RO
Conformance Units					
OPC 40086-1 Material Tracking					

5.17.2 SetCrystallizationTemperature

The *SetCrystallizationTemperature* provides the crystallization temperature and is given by the *MSSConditionType*, see 5.12.11.

5.17.3 SetCrystallizationTime

The *SetCrystallizationTime* provides the crystallization time and is given by the *MSSConditionType*, see 5.12.13.

5.17.4 SetStandbyCrystallizationTemperature

The *SetStandbyCrystallizationTemperature* provides the standby crystallization temperature and is given by the *MSSConditionType*, see 5.12.12.

5.17.5 ActualCrystallizationTemperature

The *ActualCrystallizationTemperature* provides the crystallization temperature and is given by the MSS.

5.17.6 ActualCrystallizationTime

The *ActualCrystallizationTime* provides the crystallization time and is given by the MSS.

6 Profiles and ConformanceUnits

6.1 Conformance Units

This chapter defines the corresponding *Conformance Units* for the OPC UA Information Model for OPC 40086-1.

Table 39 – Conformance Units for OPC 40086-1

Category	Title	Description
Server	OPC 40086-1 Entry Point	Support of *MSS_InterfaceType* with all mandatory children as entry point. There is at least one instance of the *MSS_InterfaceType* in the *Machines Object.*
Server	OPC 40086-1 Recipes	Supports the *RecipesType* for the management of recipes
Server	OPC 40086-1 Components	Supports the *MSSComponentsType* for the management of the components/materials
Server	OPC 40086-1 Outputs	Supports the *MSSOutputsType* for the description of the outputs (transfer points) of the material supply system
Server	OPC 40086-1 Jobs	Supports the *MSSJobsType* for the description of the jobs (orders) of the material supply system
Server	OPC 40086-1 Material Conditions	Supports the *MSSConditionType* for the description of the raw material conditions of the material supply system
Server	OPC 40086-1 Inputs	Supports the *MSSInputsType* for the description of the material inputs of the material supply system
Server	OPC 40086-1 Material Tracking	Supports the *MaterialTrackingEventType* for the tracking of the materials

6.2 Profiles

6.2.1 Profile list

Table 40 lists all Profiles defined in this document and defines their URIs.

Table 40 – Profile URIs for OPC 40086-1

Profile	URI
OPC 40086-1 Server Profile	http://opcfoundation.org/UA-Profile/PlasticsRubber/MSS/Server

6.2.2 Server Facets

6.2.2.1 Overview

The following sections specify the *Facets* available for *Servers* that implement the OPC 40086-1 companion specification. Each section defines and describes a *Facet* or *Profile*.

6.2.2.2 OPC 40086-1 Server Profile

Table 41 defines the basic *Profile* for a MSS

Table 41 – OPC 40086-1 Server Profile

Group	Conformance Unit / Profile Title	Mandatory / Optional
Server	0:Embedded Server 2017 (defined in OPC 10000-7)	M
Server	0:ComplexType Server Facet (defined in OPC 10000-7)	M
Server	0:Standard Event Subscription Server Facet (defined in OPC 10000-7)	M
Server	0:Method Server Facet (defined in OPC 10000-7)	M
Server	2:BaseDevice Server Facet (defined in OPC 10000-100)	M
MSS	OPC 40086-1 Entry Point	M
MSS	OPC 40086-1 Recipes	M
MSS	OPC 40086-1 Components	M
MSS	OPC 40086-1 Outputs	M
MSS	OPC 40086-1 Jobs	M
MSS	OPC 40086-1 Material Conditions	M
MSS	OPC 40086-1 Inputs	O
MSS	OPC 40086-1 Material Tracking	O

7 Namespaces

7.1 Namespace Metadata

Table 42 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 42 – NamespaceMetadata Object for this Document

Attribute	**Value**	
BrowseName	http://opcfoundation.org/UA/PlasticsRubber/MSS/	
Property	**DataType**	**Value**
NamespaceUri	String	http://opcfoundation.org/UA/PlasticsRubber/MSS/
NamespaceVersion	String	RC 1.00.01
NamespacePublicationDate	DateTime	2022-10-19
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.

7.2 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 43 provides a list of mandatory and optional namespaces used in an OPC 40086-1 OPC UA *Server*.

Table 43 – Namespaces used in a OPC 40086-1 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 (OPC 40001-1). The namespace index is *Server* specific.	Mandatory
http://opcfoundation.org/UA/PlasticsRubber/MSS/	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 44 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 44 – Namespaces used in this document

NamespaceURI	Namespace Index	Example
http://opcfoundation.org/UA/	0	0:EngineeringUnits
http://opcfoundation.org/UA/DI/	2	2:DeviceRevision
http://opcfoundation.org/UA/Machinery/	3	3:MachineIdentificaionType

Annex A
(normative)

OPC 40086-1 Namespace and mappings

A.1 Namespace and identifiers for OPC 40086-1 Information Model

This appendix defines the numeric identifiers for all of the numeric *NodeIds* defined in this specification. The identifiers are specified in a CSV file with the following syntax:

`<SymbolName>, <Identifier>, <NodeClass>`

Where the *SymbolName* is either the *BrowseName* of a *Type Node* or the *BrowsePath* for an *Instance Node* that appears in the specification and the *Identifier* is the numeric value for the *NodeId*.

The BrowsePath for an Instance Node is constructed by appending the BrowseName of the instance Node to the BrowseName for the containing instance or type. An underscore character is used to separate each BrowseName in the path. Let's take for example, the MachineInformationType ObjectType Node which has the ControllerName Property. The Name for the ControllerName InstanceDeclaration within the MachineInformationType declaration is: MachineInformationType_ControllerName.

The *NamespaceUri* for all *NodeIds* defined here is http://opcfoundation.org/UA/PlasticsRubber/MSS/

The CSV released with this version of the specification can be found here:

– http://www.opcfoundation.org/UA/schemas/PlasticsRubber/MSS/1.00/NodeIds.csv

NOTE The latest CSV that is compatible with this version of the specification can be found here:

– http://www.opcfoundation.org/UA/schemas/PlasticsRubber/MSS/NodeIds.csv

A computer processible version of the complete Information Model defined in this specification is also provided. It follows the XML Information Model schema syntax defined in OPC 10000-6.

The Information Model Schema for this version of the document (including any revisions, amendments or errata) can be found here:

– http://www.opcfoundation.org/UA/schemas/PlasticsRubber/MSS/1.00/Opc.Ua.PlasticsRubber.MSS.NodeSet2.xml

NOTE The latest Information Model schema that is compatible with this version of the specification can be found here:

– http://www.opcfoundation.org/UA/schemas/PlasticsRubber/MSS/Opc.Ua.PlasticsRubber.MSS.NodeSet2.xml