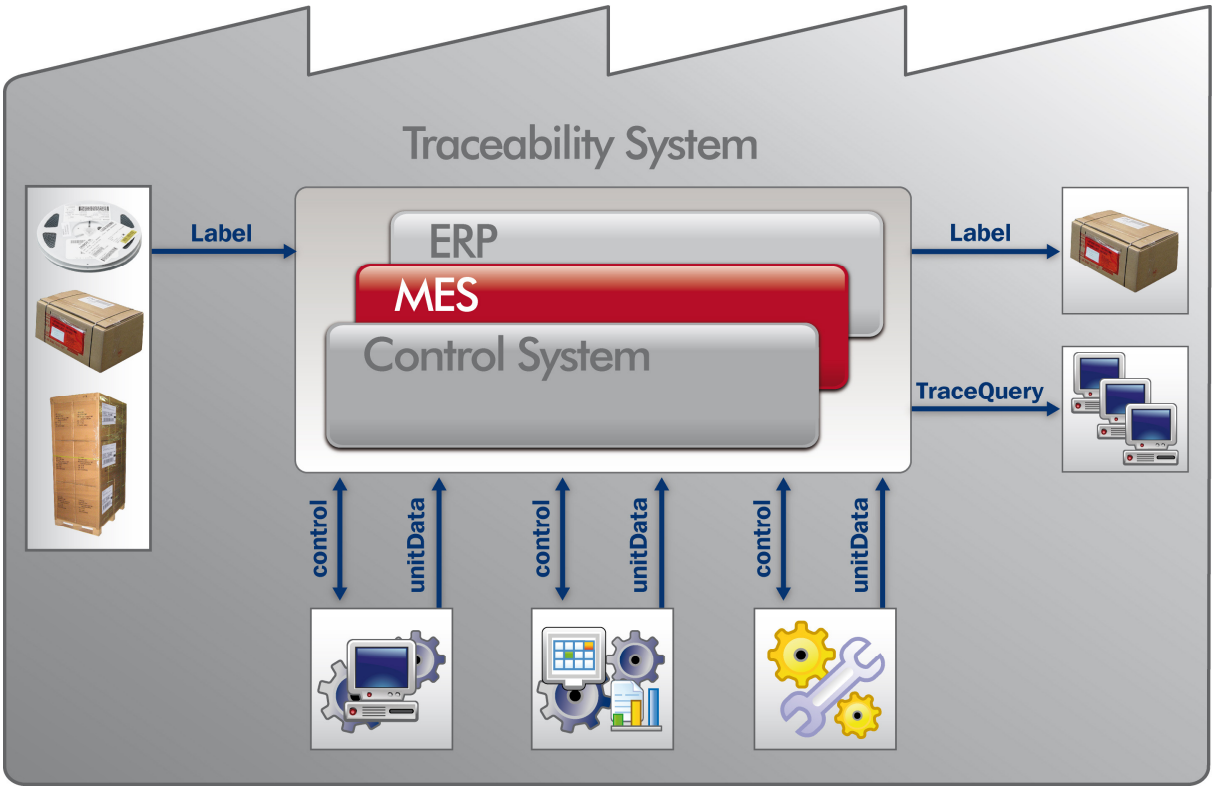
ZVEI:
Electronic Components
and Systems

Identification and Traceability in the Electrical and Electronics Industry



ZVEI Interfaces to Shopfloor

unitData

Version 1.1.0

Foreword

Work on the ZVEI manual for the entire supply and value-added chain (see MIT 1 "Guideline for Identification and Traceability) has also served as a draft for an interface to Shopfloor for connecting machines, devices and workstations.

One goal of this undertaking is to standardise the interface for processes in general.

The result was the creation of two XML-based interfaces which are freely available and recommended by ZVEI for connecting to Shopfloor:

- ***control*** for transferring data (requests and return messages) for advanced process control while a product is being processed
- ***unitData*** for transferring processing data of a product

History / changes

In MIT-2 "ZVEI-Interfaces-ChangeHistory" the history of changes of the interfaces ***control*** and ***unitData*** is described.

ZVEI – German Electrical and Electronic Manufacturers' Association e.V.

Electronic Components and Systems

Lyoner Straße 9

60528 Frankfurt am Main

Phone: 069 6302 – 276

Fax: 069 6302 – 407

E-mail: zvei-be@zvei.org

www.zvei-traceability.de

Table of contents

1 Introduction

This document describes the structure of the ZVEI standard ***unitData*** interface for transferring processing data of a product.

Various transfer protocols are available for transferring data in XML structures. They are specified in greater detail in MIT 7 "ZVEI interface transfer protocols".

The XML file described here (see MIT 14 "unitData-1.1.xml") serves as an example of the use of the XSD schema file (see MIT 9 "unitData-1.1.xsd"). To allow for concrete use of the interface, the structures between the communication partners must be co-ordinated in terms of content.

The structures consist of:

A uniform cover sheet unitData with the attributes

- Unique ID of the material being processed, for example the serial number or order number (depending on the depth of traceability) ➔ unit name
- Machine or workstation ➔ equipment
- Time of processing ➔ starttime, endtime
- Status of processing (e.g. ok or nok) ➔ state
- Optional additional information (for example plant, operation, material)

Optional system sheets for

- Tools and production resources ➔ productionResources
- Parameters and nominal values ➔ processingParameters
- Additional properties ➔ properties
- Material consumption, installation and removal information ➔ assembly, disassembly
- General measurement data ➔ measuring
- Test, diagnostic and repair data ➔ test, diagnosis, repair
- Handling workpiece carriers ➔ subUnitData
- Additional serial numbers ➔ additionalId
- Additional data ➔ additionalData
- Actions to be performed by recipient ➔ actions

1.1 Symbols used

Three different symbols are used in this documentation to emphasise important content items.

Attention!

This symbol refers to important information for which compliance is absolutely mandatory.

Explanation!

This symbol refers to explanatory information.

Tip!

This symbol identifies tips which provide faster or more efficient solutions.

1.2 Legend

[Node/attribute **]** Square brackets: ➔ optional node/attribute

< Node/attribute **>** Pointed brackets: ➔ alternative node/attribute

Explanation!

If an attribute is required (not optional), the value must be assigned (no empty string).

If an attribute is not required (optional) but is present with the value = "" (empty string), the attribute will be ignored. The attribute will then be handled as if it were not even present.

2 Explanations of XML

Data is transferred via standard compliant XML structures. The layout and format of XML structures for each interface are saved in an XSD schema file.

Explanation!

All information about XML can be found in http://www.w3.org/XML/. XML specifications with translations into various languages are also available at the same location.

Additional sites with information related to this topic are:

- WIKIPEDIA, the free encyclopaedia http://de.wikipedia.org/wiki/XML

2.1 XPath (addressing nodes and attributes)

The XML Path Language (XPath) is a query language developed by the W3 consortium to address parts of an XML document. An XPath expression addresses parts of an XML document, which is considered as a tree.

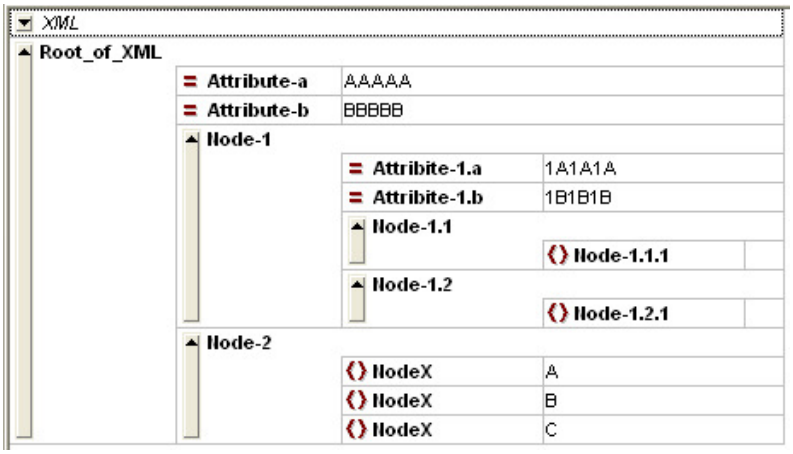


Fig 1: Example of an XPath

Examples of XPath expressions for the XML structure from Fig 1:

- **/Root_of_XML** selects the root element "Root_of_XML" of the XML structure.
- **/*** selects the root element independently of the name
  (every well formed XML document has exactly one root element)
- **/Root_of_XML /Node-2/NodeX** selects *all* "NodeX" elements within the "Node-2" node.
  In this example 3 elements are addressed: NodeX=A, NodeX=B, NodeX=C
- **child::*** selects all child elements of the current node
- **child:: NodeX** selects all "NodeX" children of the current node
- **./*** selects all subelements of the current node
- **/Root_of_XML/attribute::Attribute-a** (abbreviated notation **/Root_of_XML/@Attribute-a**) selects the attribute "Attribute-a" of root element "Root_of_XML" and addresses the value "AAAA"
- **/Root_of_XML/attribute::*** (abbreviated notation **/Root_of_XML/@***) selects all attributes of root element "Root_of_XML"
- **attribute::Attribute-1.a** (abbreviated notation **@Attribute-1.a**) selects all attributes "Attribute-1.a" of the current node
- **attribute::*** (abbreviated notation **@***) selects all attributes of the current node

Explanation!

All information related to XPath can be found at http://www.w3.org/TR/xpath20.

Additional sites with information related to this topic are:

- ZVON.org, The Guide to the XML Galaxy: http://www.zvon.org/xxl/XPathTutorial
- WIKIPEDIA, the free encyclopaedia: http://de.wikipedia.org/wiki/XPath

2.2 Serialisers and parsers

When sending, receiving and processing XML structures, current format-supporting XML processing tools must be used. XML tools must also be taken into consideration in the XSD schema file specified in the XML file. This ensures compliance with XML specifications.

- XML serialisers check compliance with XML specifications when data structures are sent
- XML parsers check compliance with XML specifications when data structures are received

Attention!

Before an interface is integrated, the sender must check to ensure compliance with XML specifications using a current XML parser, XML editor or XML checker. The XML tool that is used must also check to ensure the XSD schema file complies with the XML schema.

2.2.1 Formats in the XML file

The format of data transferred as an XML structure must comply with the requirements of the XML schema in the XSD schema file.

2.2.1.1 Character formats

The character set for character formats is optionally defined as the "encoding" attribute in the first node of the XML structure.

Fig 2: XML encoding

Examples of the "encoding" attribute:

<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="iso-8859-1"?>

Explanation!

The standard character set is loaded without the "encoding" attribute being specified.

Attention!

If the character sequences are intended to include special characters and umlauts, it is essential to specify the corresponding character set in the "encoding" attribute.

2.2.1.2 Numeric formats

Attention!

In general the number formats are given from the XSD schema files

In addition to the XML standard formats long and double in the XSD schema file ZVEI Version.xsd-common-specific formats are defined as simpleType.

Available number formats:

Format	Source	Description
long	Standard XML	Natural number together with negatives Examples: … -4; -3; -2; -1; 0; 1; 2; 3; …
double	Standard XML	Floating point number Examples: -1.5; -1; -0.5; 0; 0.5; 1; 1.5; 3.1415; 2.71828  **Explanation**: In the double floating-point format a point is to be used as the decimal point (international notation) and not a comma as in the German spelling.
positiveDouble	ZVEI-common simpleType	Floating point number > 0 Beispiele: 0.5; 1; 1.5; 3.1415; 2.71828  **Explanation:** The number 0 is not permitted.
measureDataType	ZVEI-common simpleType	Value in a special notation (see 2.2.1.2.1 measureDataType)

2.2.1.2.1 measureDataType

In some cases it makes sense for the sake of resolution or readability to display a value in a certain notation.

This can be done with the simpleType measureDataType. The attribute "measureDataType" holds the type of data and the attribue "value" holds the value. Example: measureDataType="decimal" value="3.1415".

Measurement and numerical values with the attribute "measureDataType" can have the following data types:

decimal, exponential, metricPrefix, hexadecimal, binary, string

Available measureDataType

measureDataType	Example	Description
decimal	0.031	Integer or floating point number  **Explanation:** In the decimal format a point is to be used as the decimal point (international notation) and not a comma as in the German spelling.
exponential	3.1E-2	Integer or floating point number, and directly afterwards the symbol E for exponent and the exponent itself as an integer  **Explanation:** A negative exponent as a minus sign. A positive exponent as a plus sign or not sign.  **Explanation** Between the number, and the symbol E, and the exponent no caracters (also no white spaces!) are allowed
metricPrefix	1µ (oder auch 31u)	Integer or floating point number, and directly afterwards the metric prefix.  **Explanation** Between the number, and the metric prefix no characters (also no white spaces!) are allowed **Prefix** **Symbol** **Multiplier** **Exp** yotta Y 1,000,000,000,000,000,000,000,000 10^{24} zetta Z 1,000,000,000,000,000,000,000 10^{21} exa E 1,000,000,000,000,000,000 10^{18} peta P 1,000,000,000,000,000 10^{15} tera T 1,000,000,000,000 10^{12} giga G 1,000,000,000 10^{9} mega M 1,000,000 10^{6} kilo k 1 10^{3} hecto h 100 10^{2} deca da 10 10^{1} 1 10^{0} deci d 0.1 10^{-1} centi c 0.01 10^{-2} milli m 0.001 1^{0-3} micro µ (u) 0.000001 10^{-6} nano n 0.000000001 10^{-9} pico p 0.000000000001 10^{-12} femto f 0.000000000000001 10^{-15} atto a 0.000000000000000001 10^{-18} zepto z 0.000000000000000000001 10^{-21} yocto y 0.000000000000000000000001 10^{-24}
hexadecimal	1F	Integer in hexadecimal notation
binary	00011111	Integer in binary notation
string	any string	Arbitrary sequence of characters

2.2.1.3 Date formats

Dates and times must be indicated in the ISO 8601-compliant format. The optional delimiters according to the ISO standard (-[hyphen] when writing a date, :[colon] when writing a time) must be fully indicated. Dates must be written using the following notation:

YYYY-MM-DD

Times must always be specified in "Coordinated Universal Time" (UTC) using the following notation:

hh:mm:ss

Attention!

The value range for representing seconds is 00 - 60! This makes it possible to represent leap seconds.

If a time is requested (for example "starttime"), it must be fully indicated, including the time zone information, as follows:

YYYY-MM-DDThh:mm:ss+hh:mm

Examples:

2009-12-01T11:01:00+01:00
means December 1, 2009, 11:01:00 in German local time (standard time).
This corresponds to December 1, 2009 10:01:00 in UTC.
+01:00 means + 1 hour compared to coordinated universal time (UTC).

2009-07-03T11:01:00+02:00
means July 3, 2009, 11:01:00 in German local time (daylight saving time).
This corresponds to July 3, 2009 09:01:00 in UTC.
+02:00 means + 2 hours compared to coordinated universal time (UTC).

A description of the ISO 8601-compliant time stamp format is available from:

http://www.cl.cam.ac.uk/~mgk25/iso-time.html

http://www.w3.org/TR/xmlschema-2/#isoformats
http://www.w3.org/TR/NOTE-datetime

Attention!

Time synchronisation is required between the communication partners. A suitable time server must be made available by the system operator for this purpose.

3 XML schema

The XML schema is stored in an XSD file.

3.1 "unitData-1.1.xsd"

The illustration below shows the schema of the ZVEI interface ***unitData*** (see MIT 9 "unitData-1.1.xsd"). XSD schema files "ZVEI-common-1.1.xsd" (see MIT 10 "ZVEI-common-1.1.xsd") and "ZVEI-testRepair-1.1.xsd" (see MIT 11 "ZVEI-testRepair-1.1.xsd") are integrated into the ***unitData*** schema . All XSD schema files must reside in the same directory.

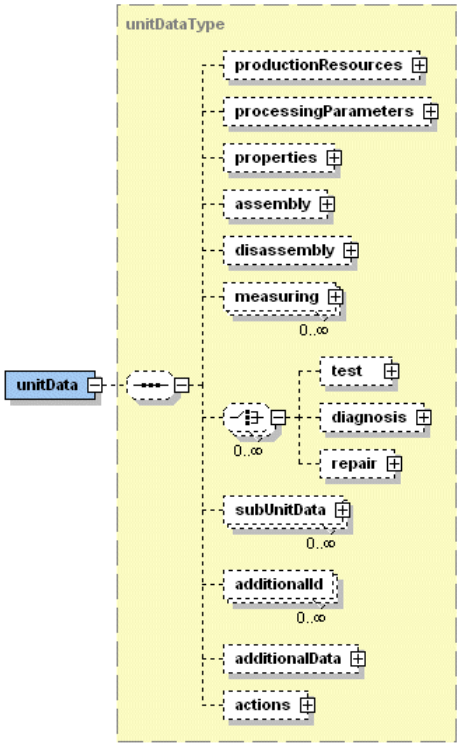


Fig 3: "unitData-1.1.xsd": Schema

A uniform cover sheet unitData with the attributes

- Unique ID of the material being processed, for example the serial number or order number (depending on the depth of traceability) ➔ unit name
- Machine or workstation ➔ equipment
- Time of processing ➔ starttime, endtime
- Status of processing (e.g. ok or nok) ➔ state
- Optional additional information (for example plant, operation, material)

Optional system sheets for

- Tools and production resources ➔ productionResources
- Parameters and nominal values ➔ processingParameters
- Additional properties ➔ properties
- Material consumption, Installation and removal information ➔ assembly, disassembly
- General measurement data ➔ measuring
- Test, diagnostic and repair data ➔ test, diagnosis, repair
- Handling workpiece carriers ➔ subUnitData
- additional serial numbers ➔ additionalId
- Additional data ➔ additionalData
- Actions to be performed by recipient ➔ actions

3.1.1 [productionResources]

Tools and production resources that were used while processing a product.

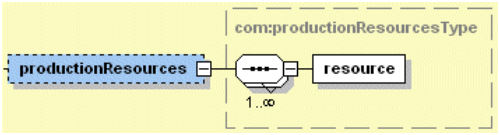


Fig 4: "unitData-1.1.xsd": Schema of productionResources

3.1.2 [processingParameters]

Parameters and target values that were used while processing a product.

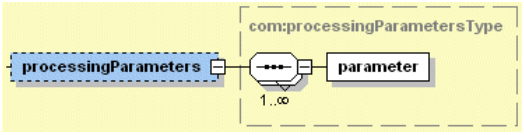


Fig 5: "unitData-1.1.xsd": Schema of processingParameters

3.1.3 [properties]

Additional properties for processing a product.

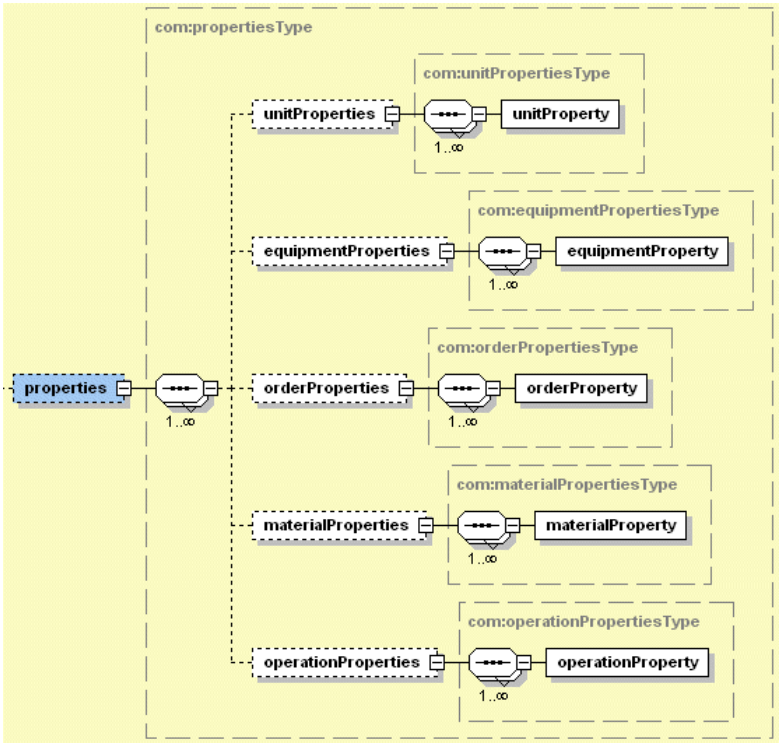


Fig 6: "unitData-1.1.xsd": Schema of properties

Additional properties

- for the unit (serial number of the material being processed)
- for the equipment (machine, manual workstation, production line or production cell)
- for the order
- for the material (product)
- for the operation

3.1.4 [assembly]

Data from material consumption and installed material (components).

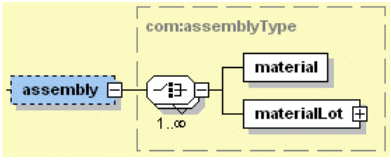


Fig 7: "unitData-1.1.xsd": Schema of assembly

3.1.5 [disassembly]

Data from disassembled material (components).

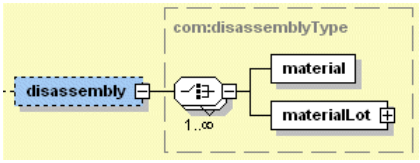


Fig 8: "unitData-1.1.xsd": Schema of disassembly

3.1.6 [measuring]

Measurement data referring directly to processing of the product.

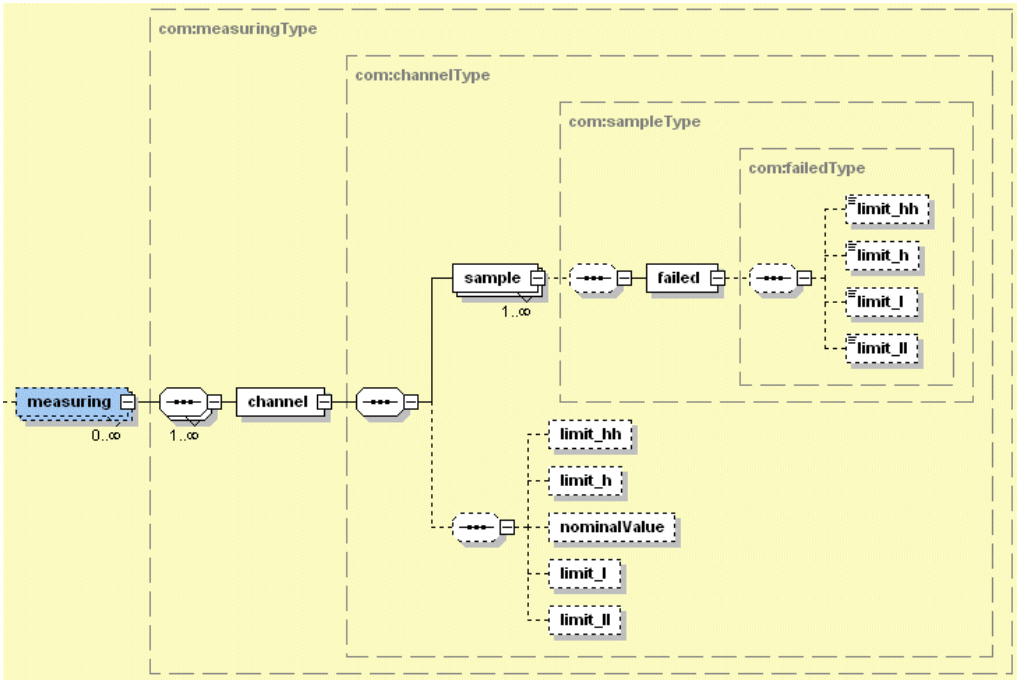


Fig 9: "unitData-1.1.xsd": Schema of measuring

3.1.7 [test]

Test data referring to a tested product.

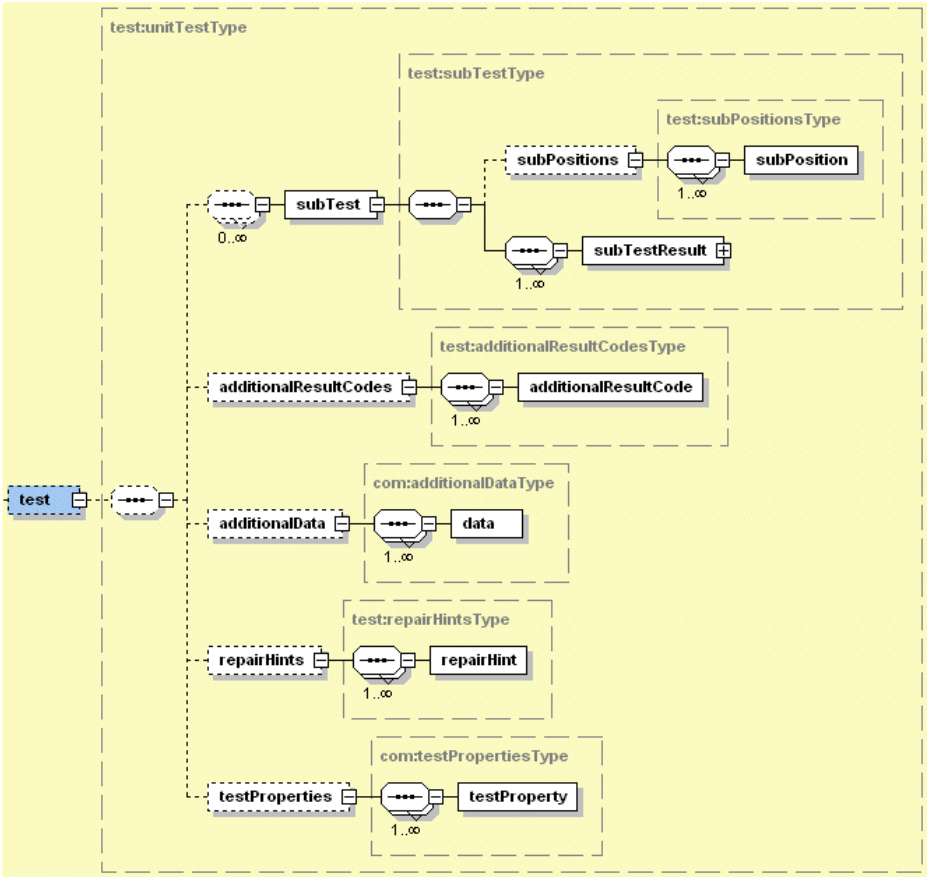


Fig 10: "unitData-1.1.xsd": Schema of test

3.1.7.1 subTest/subTestResult

Details of the result of subtests for a test that was performed.

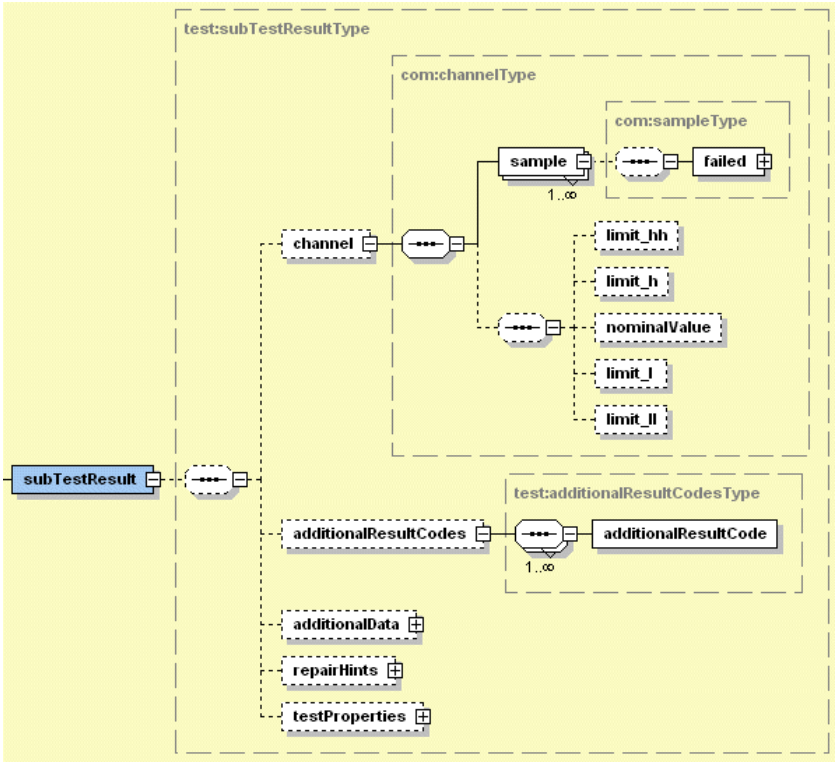


Fig 11: "unitData-1.1.xsd": Schema of test/subTest/subTestResult

3.1.8 [diagnosis]

Data for diagnostics (analysis, classification) of test data for a main test.

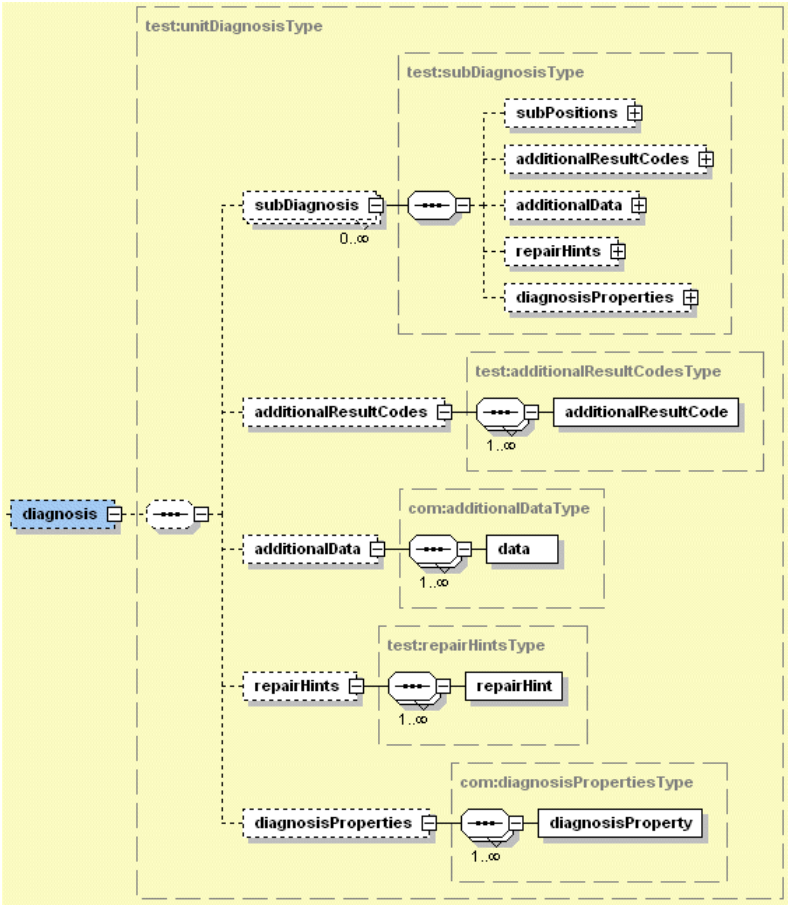


Fig 12: "unitData-1.1.xsd": Schema of diagnosis

3.1.8.1 [subDiagnosis]

Data for diagnostics (analysis, classification) of test data for a subtest.

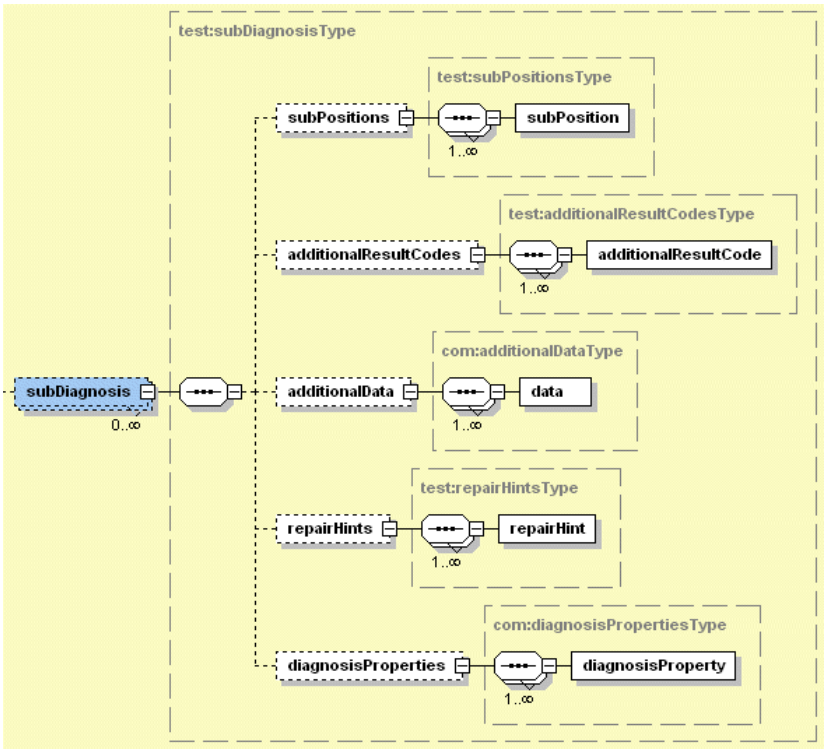


Fig 13: "unitData-1.1.xsd": Schema of subDiagnosis

3.1.9 [repair]

Data for a repair performed on a product for a main test.

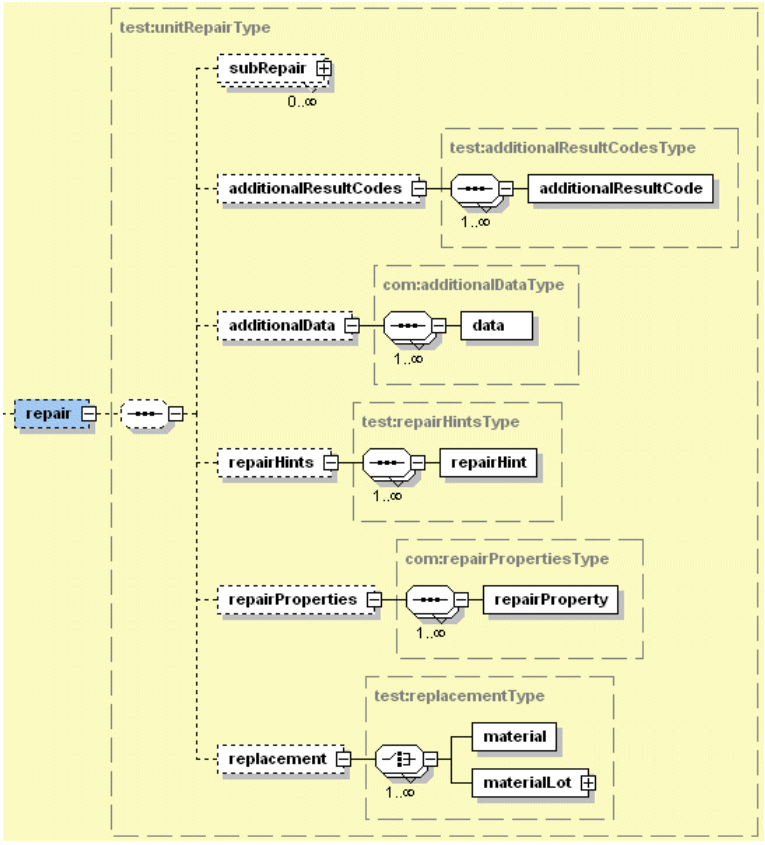


Fig 14: "unitData-1.1.xsd": Schema of repair

3.1.9.1 [subRepair]

Data for a repair performed on a product for a subtest.

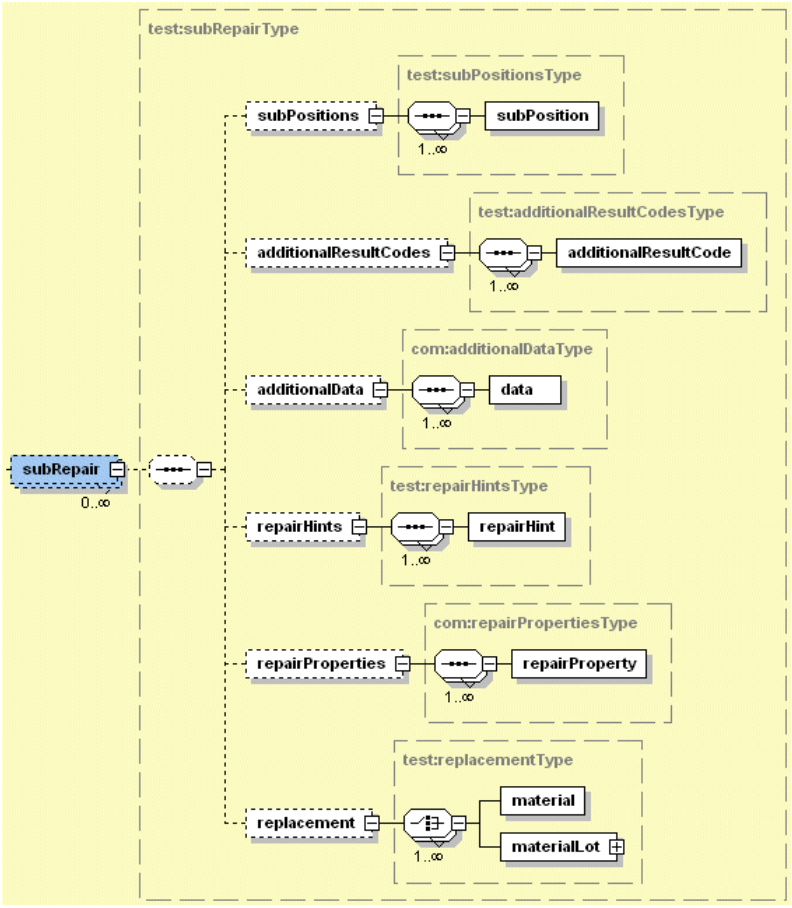


Fig 15: "unitData-1.1.xsd": Schema of subRepair

3.1.10 [subUnitData]

Optional list of products (subassemblies) in a workpiece carrier.

Explanation!

Products can be combined in a workpiece carrier for shared processing. If the processing data of the individual products may differ, the data can be transferred individually for each product as a subUnitData. An example is the combination of independent circuits to a board (virtual workpiece carrier).

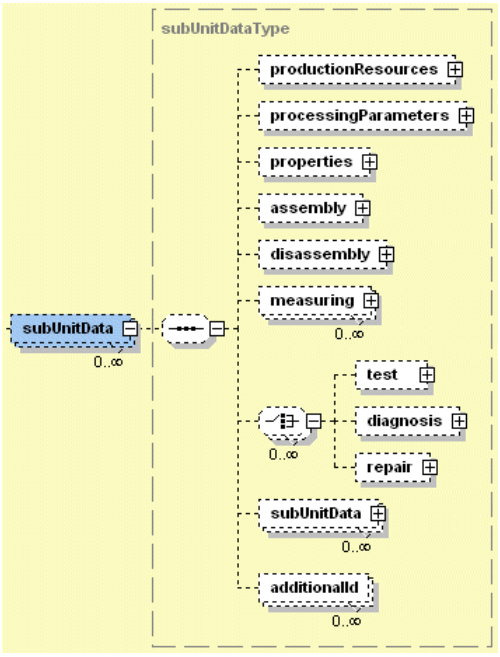


Fig 16: "unitData-1.1.xsd": Schema of subUnitData

3.1.10.1 [other subnodes]

Other optional subnodes with processing data may be located under the "subUnitData" node (similar to 3.1 "unitData-1.1.xsd")

- productionResources
- processingParameters
- properties
- assembly
- disassembly
- measuring
- test
- diagnosis
- repair
- subUnitData
- additionalId

3.1.11 [additionalId]

Optional list of additional serial numbers.

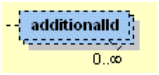


Fig 17: "unitData-1.1.xsd": Schema of additionalId

3.1.12 [additionalData]

Optional list of additional project-specific data.

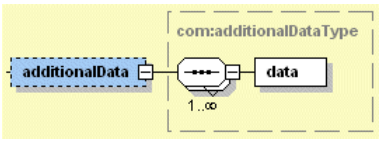


Fig 18: "unitData-1.1.xsd": Schema of additionalData

3.1.13 [actions]

Optional list of actions to be performed.

Explanation!

To allow for concrete use of the interface, the structures between the communication partners must be co-ordinated in terms of content.

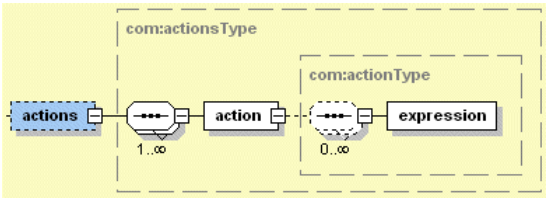


Fig 19: "unitData-1.1.xsd": Schema of actions

4 XML root: unitData

General properties of the XML root

Fig 20: XML root /unitData, general

Attribute	Format	Description
xmlns:xsi	URL	Link to the XMLSchema instance Example: http://www.w3.org/2001/XMLSchema-instance **Explanation:** No connection to the Internet is required.
xsi:noNamespaceSchemaLocation	XSD file	Path (directory and name) of the XSD reference schema file **Explanation:** The reference schema file can be saved locally (same directory as the XML file) or in a general directory. If the reference schema file is saved in the same directory as the XML file, the directory does not need to be indicated. **Attention!** The path (directory and name) of the XSD reference schema file must be configurable in the sender's system.
[locale]	String	Optional language setting for messages and descriptions. **Explanation:** The language must be specified in English in lower case letters. Examples: german, english, spanish, chinese, french, hungarian, romanian.
[senderID]	String	Identification of the sender (eg computer name - company - software version). This attribute can be used to check the software version of the sender during software integration.

Table 1: XML root /unitData/attribute::* (general)

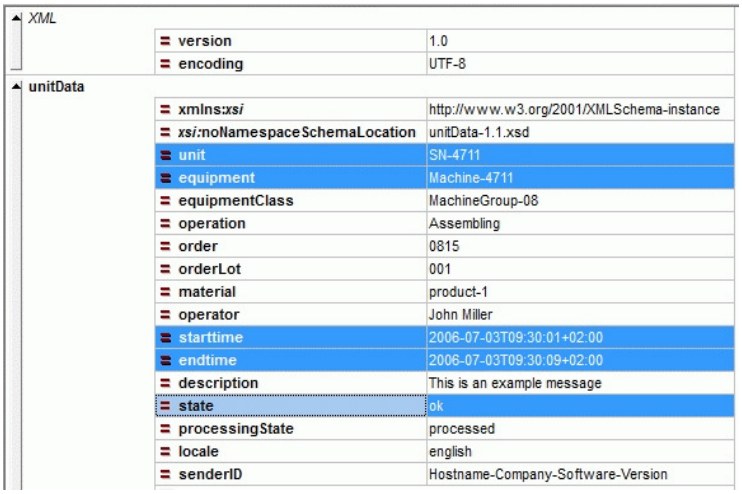


Fig 21: XML root /unitData

Attribute	Format	Description
unit	**String**	**Serial number of the material being processed**
[unitType]	String	Type of material being processed (for example Device, MaterialLot)
[unitSide]	String	Processed side of product (for example top or bottom side during testing)
[plant]	String	Plant in which the material was processed
equipment	**String**	**Name of unique designation of the machine, manual workstation, production line or production cell**
[equipmentClass]	String	Name of unique designation of a group of machines or manual workstations This attribute can be used in addition to the "equipment" attribute to transfer the group affiliation of a machine.
[operation]	String	Name of the operation or work process performed
[order]	String	Order number
[orderLot]	String	Lot number of the order
[material]	String	Material number of the manufactured product
[materialVersion]	String	Material version of the manufactured product
[materialVariant]	String	Material variant of the manufactured product
[operator]	String	Processor, machine technician
starttime	**DateTime**	**Date and time at which processing of the product was started for the relevant operation.**
[endtime]	DateTime	Date and time at which processing of the product was completed for the relevant operation.
[duration]	DateTime	Duration of product processing.
[arrivaltime]	DateTime	The time at which the product arrives in the machine (not necessarily identical with the starttime).
[departuretime]	DateTime	The time at which the processed product has left the machine again (not necessarily identical with the endtime).
[description]	String	Comments regarding processing of the product, messages and error messages that were generated
state	**String**	**Overall status of processing of the product, ok or nok**  **Explanation:** The codes for the status of processing should be configurable. Some examples are 'ok', 'nok', 'BadBoard', 'repaired'. The specific designation must be co-ordinated with the system operator.
[processingState]	String	In addition to the 'state' attribute for the status of processing of the product, a process-relevant status can optionally be assigned with the new attribute 'processingState'. Example: state=ok, processingState=installed
[locale]	String	Optional language setting for messages and descriptions.
[senderID]	String	Identification of the sender (eg computer name - company - software version). This attribute can be used to check the software version of the sender during software integration.

Table 2: XML root /unitData/attribute::*

4.1 [productionResources]

Optional transfer of tools and production resources that were used while processing a product.

4.1.1 resource

List of tools and production resources that were used.

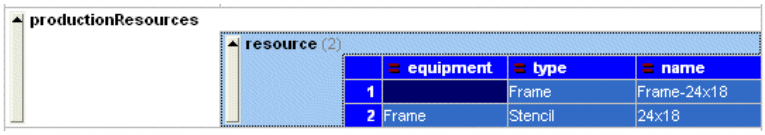


Fig 22: node /unitData/productionResources/resource

Attribute	Format	Description
[equipment]	String	Optional possibility of assigning tools and production resources to different areas within a machine. **Explanation:** If no equipment is specified, the parameters or setpoints will be assigned to the equipment of the "unitData" node.
[position]	String	Optional entry for the position of a tool within a machine
type	**String**	**Type of the production resource**
name	**String**	**Name (instance) of the production resource**
[state]	String	Status of the tool or production resource, ok or nok

Table 3: node /unitData/productionResources/resource/attribute::*

4.2 [processingParameters]

Optional transfer of parameters and setpoints that were used while processing a product.

4.2.1 parameter

List of parameters and setpoints that are used.

processingParameters

parameter (3)

	equipment	name	value	UnitOfMeasure
1		Program	Assemble-4711	
2		ProgramVersion	1.7	
3	24x18	Temperature	20.2	°C

Fig 23: node /unitData/processingParameters/parameter

Attribute	Format	Description
[equipment]	String	Optional possibility of assigning parameters and setpoints to different areas within a machine. **Explanation:** If no equipment is specified, the parameters or setpoints will be assigned to the equipment of the "unitData" node.
[position]	String	Optional entry for the position within a machine to which a parameter or setpoints apply.
name	**String**	**Type of the parameter or setpoint**
value	**String**	**Value of the parameter or setpoint**
[UnitOfMeasure]	String	Unit of the parameter or setpoint **Attention!** Always required for numeric parameters and setpoints. **Explanation:** The dimension must be entered as an SI unit of measure (see 5.4 List of relevant terms and abbreviations). In addition to SI units of measure, the following units of measure are also supported: pcs (pieces), ° (degrees), m² (square metres), l (litres), % (percent), db (decibels)
[measureDataType]	String	If the parameter or setpoint is a numeric value, the format of the numeric value can be indicated. Possible numeric formats: decimal, exponential, metricPrefix, hexadecimal, binary, string (default: decimal) (see 2.2.1.2.1 measureDataType)
[state]	String	Status of the parameter, ok or nok

Table 4: node /unitData/processingParameters/parameter/attribute::*

4.3 [properties]

Additional properties for processing a product.

4.3.1 [xxxProperties]

Grouping of properties by

- unitProperties
  Additional properties for the unit (serial number of the material being processed).
- equipmentProperties
  Additional properties for the equipment (machine, manual workstation, production line or production cell).
- orderProperties
  Additional properties for the order.
- materialProperties
  Additional properties for the material.
- operationProperties
  Additional properties for the operation.

4.3.1.1 xxxProperty

Description of a property.

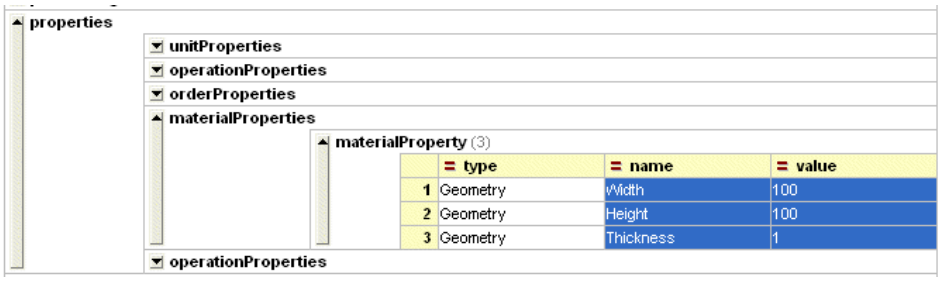


Fig 24: node /unitData/properties/xxxProperties/xxxProperty

Attribute	Format	Description
[type]	String	Type of the property
name	**String**	**Name of the property**
value	**String**	**Value of the property**
[UnitOfMeasure]	String	Unit of measure of the property **Explanation:** The dimension must be entered as an SI unit of measure (see 5.4 List of relevant terms and abbreviations).In addition to SI units of measure, the following units of measure are also supported: pcs (pieces), ° (degrees), m² (square metres), l (litres), % (percent), db (decibels)
[measuringDataType]	String	Optional data type of a property • decimal • exponential • metricPrefix • hexadecimal • binary • string **Explanation:** The data type of a property is only relevant if a UnitOfMeasure is indicated for the property. In that case the default for the data type is decimal.
[state]	String	Optional status for the property

Table 5: node /unitData/properties/xxxProperties/xxxProperty/attribute::*

4.4 [assembly]

Data from installed material (components).

Explanation!

Data about components that were replaced during a repair; can also alternatively be transferred with the repair data (see 4.7 [repair])

4.4.1 <material>

List of installed articles (components) without lot reference (batch reference).

Explanation!

Material data without lot reference (batch reference) provide only very limited options for evaluation (e.g. for process locking, tracking, material management or detailed planning). When transferring material data, a lot reference (batch reference) is therefore preferable (see 4.4.2 <materialLot>).

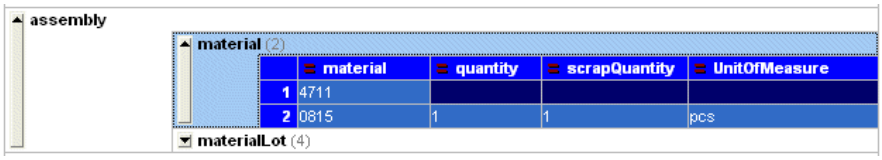


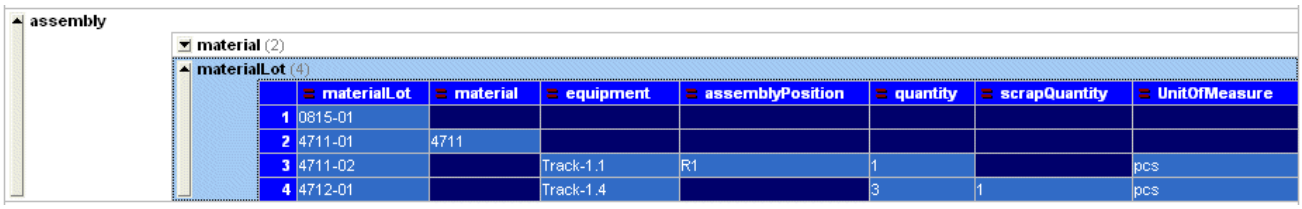
Fig 25: node /unitData/assembly/material

Attribute	Format	Description
[type]	String	Type of the installed material (component)
material	**String**	**Material number of the installed material (component)**
[materialVersion]	String	Version of the installed material
[equipment]	String	The location at which the installed material was fitted or stored during installation (for example an equipment vehicle)
[position]	String	The position at which the installed material was fitted or stored during installation
[assemblyPosition]	String	The location at which the installed material was fitted
[quantity]	Double	Quantity of the installed material (component)
[scrapQuantity]	Double	This attribute can optionally be used to report scrapping of components that occurs or is discovered while a product is being processed (faulty parts or rejects).
[UnitOfMeasure]	String	Unit of measure **Explanation:** The dimension must be entered as an SI unit of measure (see 5.4 List of relevant terms and abbreviations). In addition to SI units of measure, the following units of measure are also supported: pcs (pieces), ° (degrees), m² (square metres), l (litres), % (percent), db (decibels)
[state]	String	Status, ok or nok

Table 6: node /unitData/assembly/material/attribute::*

4.4.2 <materialLot>

List of installed lots (components, batch)



	materialLot	material	equipment	assemblyPosition	quantity	scrapQuantity	UnitOfMeasure
1	0815-01						
2	4711-01	4711					
3	4711-02		Track-1.1	R1	1		pcs
4	4712-01		Track-1.4		3	1	pcs

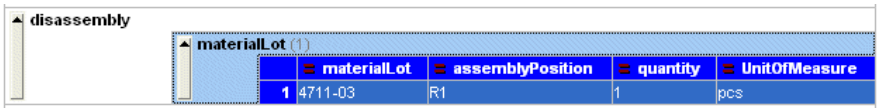
Fig 26: node /unitData/assembly/materialLot

Attribute	Format	Description
[type]	String	Type of installed lot or material (component)
materialLot	**String**	**Number of a lot (batch) of the installed material (component)**
[material]	String	Material number of the installed material (component)
[materialVersion]	String	Version of the installed material
[equipment]	String	The location at which the installed material was fitted or stored during installation (for example an equipment vehicle)
[position]	String	The position at which the installed material was fitted or stored during installation
[assemblyPosition]	String	The location at which the installed material was fitted
[quantity]	Double	Quantity of the installed material (component)
[scrapQuantity]	Double	This attribute can optionally be used to report scrapping of components that occurs or is discovered while a product is being processed (faulty parts or rejects).
[UnitOfMeasure]	String	Unit of measure **Explanation:** The dimension must be entered as an SI unit of measure (see 5.4 List of relevant terms and abbreviations). In addition to SI units of measure, the following units of measure are also supported: pcs (pieces), ° (degrees), m² (square metres), l (litres), % (percent), db (decibels)
[state]	String	Status, ok or nok

Table 7: node /unitData/assembly/materialLot/attribute::*

4.5 [disassembly]

Data from disassembled material (components). This node makes it possible to record disassembled components when components are replaced (service and repair) (see also 4.9 [repair]).



	materialLot	assemblyPosition	quantity	UnitOfMeasure
1	4711-03	R1	1	pcs

Fig 27: node /unitData/disassembly/materialLot

The "disassembly" node corresponds in format and attributes to the "assembly" node (see 4.4 [assembly]).

4.6 [measuring]

Measurement data referring directly to processing of the product.

Attention!

The measurement data listed here describe only the environment and conditions that play a role in processing a product (for example oven temperatures or relative humidity).

The 4.7 [test] node should be used for test data measured directly on the product.

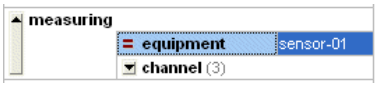


Fig 28: node /unitData/measuring

Attribute	Format	Description
[equipment]	String	Optional possibility of assigning measurement values to various ranges or sensors within a machine. The unique identity or number of the location at which the measurement values were recorded (for example the ID or name of a sensor) **Explanation:** If no equipment is specified for the "measuring" node, the measurement values are assigned to the equipment of the "unitData" node. (see Table 2: XML root /unitData/attribute::*)

Table 8: node /unitData/measuring/attribute::*

4.6.1 channel

List of measurement channels

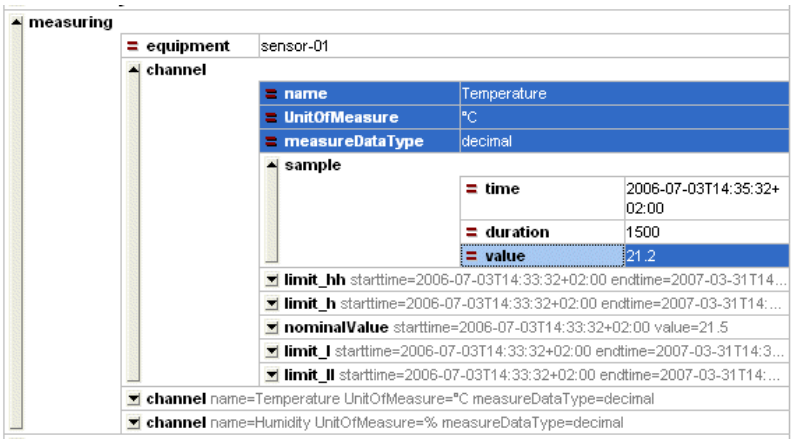


Fig 29: node /unitData/measuring/channel

The channel node is described in MIT 6 "ZVEI interface TestAndRepair" in the "channel" section.

4.7 [test]

Test data referring to a tested product.

Explanation!

Test data are divided into one overall test and optionally into one or more subtests. To transfer test details (for example the designation of a test position) a subtest must be defined in the interface.

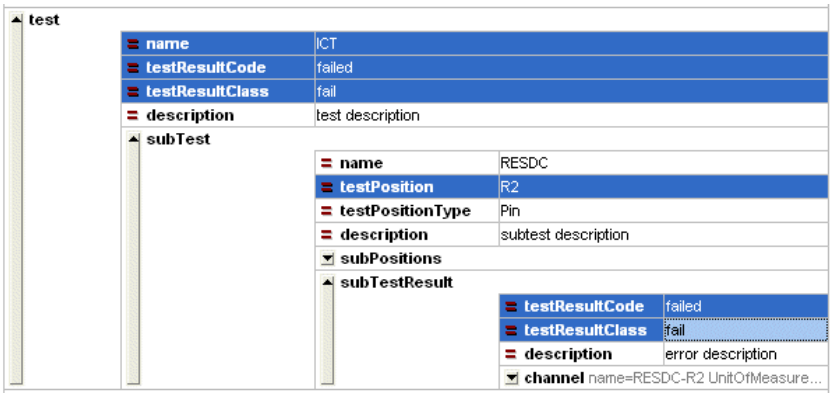


Fig 30: node /unitData/test

Attribute	Format	Description
name	**String**	**Unique name of a test or test sequence**
testResultCode	**String**	**Result of the overall test** **Explanation:** The codes for the possible test results should be configurable. Some examples of standard designations include: 'passed', 'failed', 'aborted'. The specific designation must be co-ordinated with the system operator.
[testResultClass]	String	Optional attribute for classification of test results. The following classes can be specified: 'pass', 'certifiedPass', 'fail', 'interrupt', 'unknown'. If no classification is transferred, the default 'unknown' is assumed.
[description]	String	Optional description of a test

Table 9: node /unitData/test/attribute::*

Explanation!

Further details about the "test" node are described in MIT 6 "ZVEI interface TestAndRepair".

4.8 [diagnosis]

Data related to diagnostics (analysis and classification) of test data.

Explanation!

To transfer diagnostics data, a reference to the test for which the analysis or classification was performed is required.

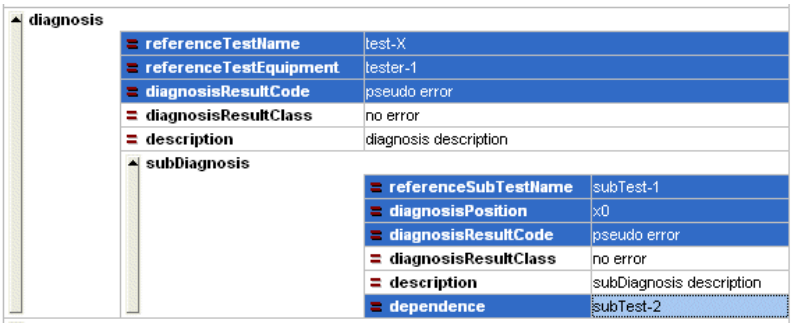


Fig 31: node /unitData/diagnosis

Attribute	Format	Description
referenceTestName	**String**	**The name of the test of which the results were the basis for diagnostics.**
[referenceTestEquipment]	String	Optional name or unique designation of the test machine or test station of which the test results were the basis for diagnostics.
diagnosisResultCode	**String**	**Result of diagnostics.** **Explanation:** The codes for the results of diagnostics should be configurable. The specific designation must be co-ordinated with the system operator.
[diagnosisResultClass]	String	Optional attribute for classifying diagnostic results. The following classes can be specified: 'fault', 'pseudoFault', 'testFault', 'consecutiveFault', 'unknown'. When no classification is passed then the default 'unknown' applies.
[dependence]	String	These optional attributes can be used to refer a secondary error to a main error.
[description]	String	Optional description or comments about the repair that was performed

Table 10: node /unitData/diagnosis/attribute::*

Explanation!

Further details about the "diagnosis" node are described in MIT 6 "ZVEI interface TestAndRepair".

4.9 [repair]

Data about a repair performed on a product.

Explanation!

To transfer repair data, a reference to the test of which the results were the basis for the repair is required.

Explanation!

Data transferred in the "repair" node about components that were replaced during a repair must not be transferred additionally in the "assembly" node.

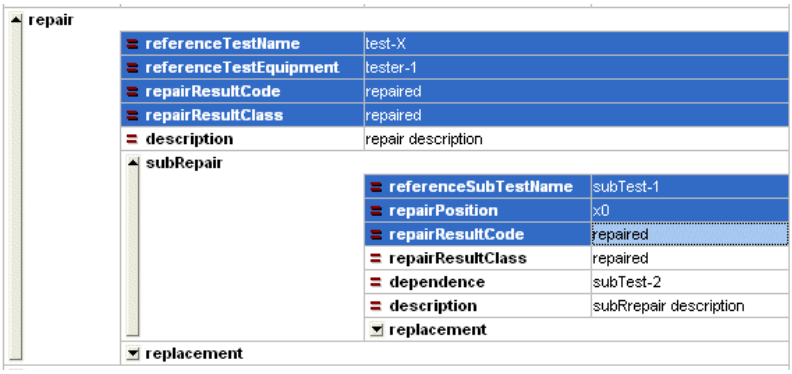


Fig 32: node /unitData/repair

Attribute	Format	Description
referenceTestName	**String**	**The name of the test of which the results were the basis for repair.**
[referenceTestEquipment]	String	Optional name or unique designation of the test machine or test station of which the test results were the basis for the repair.
repairResultCode	**String**	**Result of the repair.** **Explanation:** The codes for the results of the repair should be configurable. The specific designation must be co-ordinated with the system operator.
[repairResultClass]	String	Optional attribute for classifying repair results. The following classes can be specified: 'successful', 'failed', 'interrupt', 'unknown'. When no classification is passed then the default 'unknown' applies.
[dependence]	String	These optional attributes can be used to refer a secondary error to a main error.
[description]	String	Optional description or comments about the repair that was performed

Table 11: node /unitData/repair/attribute::*

Explanation!

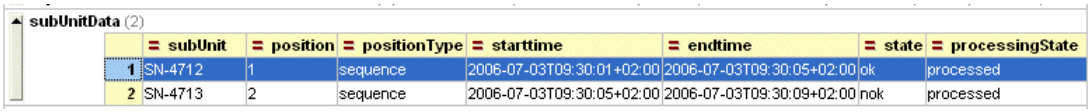
Further details about the "repair" node are described in MIT 6 "ZVEI interface TestAndRepair".

4.10 [subUnitData]

Optional list of products (subassemblies) in a workpiece carrier.

Explanation!

Products can be combined in a workpiece carrier for shared processing. If the processing data of the individual products may differ, the data can be transferred individually for each product as a subUnitData. An example is the combination of independent circuits to a board (virtual workpiece carrier).



subUnitData (2)

	subUnit	position	positionType	starttime	endtime	state	processingState
1	SN-4712	1	sequence	2006-07-03T09:30:01+02:00	2006-07-03T09:30:05+02:00	ok	processed
2	SN-4713	2	sequence	2006-07-03T09:30:05+02:00	2006-07-03T09:30:09+02:00	nok	processed

Fig 33: node /unitData/subUnitData

Attribute	Format	Description
<subUnit>	**String**	**Serial number of a processed product in a workpiece carrier** **Explanation:** Either the serial number of the processed product or the serial number of the workpiece carrier must be specified, combined with the number in the workpiece carrier (position).
[subUnitType]	String	Type of material being processed (for example Device, MaterialLot)
[subUnitSide]	String	Side of processed material (for example top or bottom side during testing)
<position>	**String**	**Position of the subproduct (subassembly) in the workpiece carrier**
<positionType>	**String**	**Type of the position in the workpiece carrier** • sequence • line,column • coordinates (x, y)
[material]	String	Material number of the manufactured product
[materialVersion]	String	Material version of the manufactured product
[materialVariant]	String	Material variant of the manufactured product
[starttime]	DateTime	Date and time at which processing of the product was started.
[endtime]	DateTime	Date and time at which processing of the product was completed.
[description]	String	Comments regarding processing of the product (subassembly) in a workpiece carrier, messages and error messages that were generated
state	**String**	**Overall status of the processing of a product (subassembly) in a workpiece carrier, ok or nok.** **Explanation:** The codes for the status of processing should be configurable. Some examples of standard designations include: 'ok', 'nok', 'BadBoard', 'repaired'. The specific designation must be co-ordinated with the system operator.
[processingState]	String	In addition to the 'state' attribute for the status of processing of the product, a process-relevant status can optionally be assigned with the new attribute 'processingState'. Example: state=ok, processingState=installed

Table 12: node /unitData/subUnitData/attribute::*

4.10.1 [other subnodes]

Other optional subnodes with processing data may be located under the "subUnitData" node (similar to 4 XML root: unitData)

- productionResources
- processingParameters
- properties
- assembly
- disassembly
- measuring
- test
- diagnosis
- repair
- subUnitData
- additionalId

4.11 [additionalId]

Optional list of additional serial numbers.

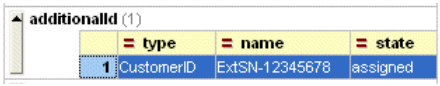


Fig 34: node /unitData/additionalId

Attribute	Format	Description
type	**String**	**Type of an additional serial number (for example customer serial number, equipment serial number)**
name	**String**	**Additional serial number**
[state]	String	Optional status for an additional serial number (for example reserved or assigned)  **Explanation:** The codes for the status should be configurable. Some examples of standard designations include: 'ok', 'nok', 'aborted', 'BadBoard', 'repaired, 'passed', 'failed'. The specific designation must be co-ordinated with the system operator.

Table 13: node /unitData/additionalId/attribute::*

4.12 [additionalData]

Optional list of additional data.

Explanation!

This node can be used to transfer project-specific data. The actual data or simply a link to the data (for example a URL or file path) can be transferred.

4.12.1 data

List of additional project-specific data.

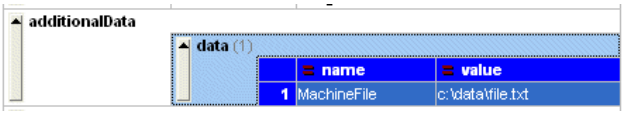


Fig 35: node /unitData/ additionalData /data

Attribute	Format	Description
[type]	String	Type of the additional data (for example measurement value, file, external source)
name	**String**	**Name of the additional data**
value	**String**	**Value or reference (for example a URL or file path)**

Table 14: node /unitData/additionalData/data/attribute::*

4.13 [actions]

Optional list of actions to be performed.

Explanation!

To allow for concrete use of the interface, the structures between the communication partners must be co-ordinated in terms of content.

4.13.1 action

An action to be performed.

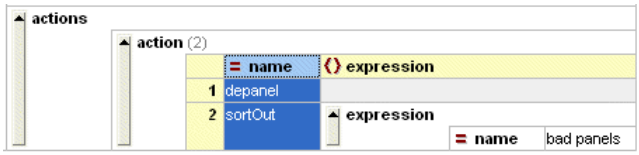


Fig 36: node /unitData/actions/action

Attribute	Format	Description
name	**String**	**Name of the action to be performed** **Explanation:** The names of possible actions to be performed should be configurable. Some examples are 'depanel', 'scrap'

Table 15: node /unitData/actions/action/attribute::*

4.13.2 [expression]

Optional list of expressions to restrict or more narrowly define an action to be performed. The expression may be a simple string used to describe an object or a regular expression.

Explanation!

A regular expression (abbreviated as regex or regexp) is a special text sequence used to describe a search pattern (similar to a wildcard). Wildcards such as *.txt are well known in file management.

The regex equivalent to *.txt is `(?i)\.txt$` where
- (?i) means ("ignore case")
- $ means „End of line“ or „end of string“ - depending on the context

Additional information about regular expressions is available on the Internet, for example from http://www.regular-expressions.info .

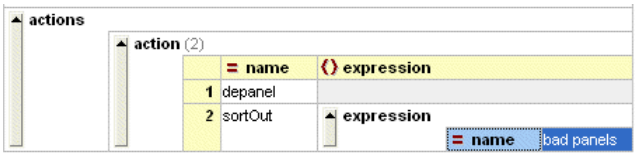


Fig 37: node /unitData/actions/action/expression

Attribute	Format	Description
name	**String**	**Designation of an expression or regular expression.**

Table 16: node /unitData/actions/action/expression/attribute::*

4.13.3 Examples

The following table shows some examples of the use of the node actions.

Action	Expression	Beschreibung
depanel		Separation of individual circuits (workpiece carriers)
relocate		Change the order of a unit

5 Appendix

5.1 List of relevant documents

MIT 1 "Guideline for Identification and Traceability"	ZVEI manual for the entire supply and value-added chain
MIT-2 " ZVEI-Interfaces-ChangeHistory "	This document describes the history of changes to the interfaces control and unitData.
MIT-3 "ZVEI-Interfaces-Overview"	This document provides an overview of possible applications of the ZVEI control and unitData interfaces during the manufacturing of a product
MIT 4 "ZVEI interface control"	This document describes the structure of the standard ***control*** interface for transferring data (requests and return messages) in process control (advanced process control) during the processing of a product.
MIT 5 "ZVEI interface unitData"	This document describes the structure of the standard ***unitData*** interface for transferring data to process a product.
MIT 6 "ZVEI interface TestAndRepair"	Description of the substructures of ZVEI standard interfaces for transferring test and repair data
MIT 7 "ZVEI interface transfer protocols"	Description of transfer protocols for transferring XML structures for ZVEI standard interfaces
MIT 8 "control-1.1.xsd"	XSD schema as file for ZVEI standard interface control
MIT 9 "unitData-1.1.xsd"	XSD schema as file for ZVEI standard interface unitData
MIT 10 "ZVEI-common-1.1.xsd"	XSD schema as file for general types that are used in various ZVEI standard interfaces
MIT 11 "ZVEI-testRepair-1.1.xsd"	XSD schema as file for general types used to forward test and repair data that are used in various ZVEI standard interfaces
MIT 12 "control_Request-1.1.xml"	Sample request for the ZVEI standard control interface as a file
MIT 13 "control_Response-1.1.xml"	Sample return message for the ZVEI standard control interface as a file
MIT 14 "unitData-1.1.xml"	Example of the ZVEI standard unitData interface as a file

5.2 List of illustrations

5.3 List of tables

5.4 List of relevant terms and abbreviations

Term	Description
Product	Definition according to GPSG / ProdHG Description of a material that is produced Examples: • Electronic flat module • Mechanical assembly • Device
Parts list	Description of the type and number of components included in a product
Work sequence	Description of a work plan with the production steps necessary to manufacture a product.
Work process	Description of an individual production step (work or process step) that is required as part of the work sequence to manufacture a product. A work process does not need to be assigned yet to any specific equipment.
Equipment	Resources used to manufacture a product such as • Production machines • Test machines • Manual workstations • Production lines • Production cells • Handling systems • Scanners and readers • Terminals (GUI) • Signal devices (traffic signals)
PCB	Unfitted independent circuit
Flat module	Fitted independent circuit
Panel	A combination of independent circuits to form a board (virtual workpiece carrier); single or multiple panels depending on the number of independent circuits
Panel blank	Unprocessed single or double panel
Serial number	Unique number of a manufactured part
Component	A component (component, assembly, bulk material) is used for a product (for example mounted, fitted)
Batch	Physical combination of individual parts, for example roller, rod and tray in a smallest packaging unit.
Lot	Quantity of parts that have a barcode (batch). There are two types of lots, supplier lot and incoming goods lot. Supplier lots can be subdivided into smaller incoming goods lots.
MaterialLot (lot number or batch number)	The unique number of a lot or batch. The number of a supplier lot is already assigned by the supplier at delivery. The numbers of incoming goods lots can be (re)assigned in incoming goods. If incoming goods lots are combined in a smallest packaging unit it is equivalent to a batch.
SI unit	Abbreviation for: "Système International d'Unités" SI units are part of the International System of Units for Natural Sciences. This is the standard system as defined by law in most countries. It describes seven fundamental SI units: • Length: Metre (m) • Weight: Kilogram (kg) • Time: Second (s) • Current strength: Ampere (A) • Temperature: Kelvin (K) • Amount of substance: Mole (mol) • Light intensity: Candela (cd) Numerous other units are derived from these basic units, for example Newton, Pascal or Joule.