**INTERNATIONAL STANDARD ISO/IEC 15909-2:2011**

TECHNICAL CORRIGENDUM 1

Published 2013-12-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

# Systems and software engineering — High-level Petri nets —

## Part 2: **Transfer format**

TECHNICAL CORRIGENDUM 1

*Ingénierie des systèmes et du logiciel — Réseaux de Petri de haut niveau —*

*Partie 2: Format de transfert*

*RECTIFICATIF TECHNIQUE 1*

Technical Corrigendum 1 to ISO/IEC 15909-2:2011 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

This corrigendum proposes changes to correct the several defects, which came up during more intensive use and the implementation of new tools supporting ISO/IEC 15909-2:2011. Several errors have been identified, which need to be fixed. These ambiguities mostly concern mismatches between the UML models and their mapping to XML syntax. Some changes are made for clarity reasons only or to make the intention of ISO/IEC 15909-2:2011 explicit.

Altogether, there are 14 defects that are corrected in this corrigendum.

**ICS 35.080** **Ref. No. ISO/IEC 15909-2:2011/Cor.1:2013(E)**



Published in Switzerland

# 1 Scope

## 1.1 Update of the affected type URIs

This Technical Corrigendum introduces two new URIs that identify the type of Petri net in the version according to this Technical Corrigendum. There is one new URI for High-level Petri Net Graphs and one for Symmetric Nets (the other net types are not affected by this Technical Corrigendum and, therefore, do not need to be changed).

In order to be consistent, the respective URI of the affected Petri net types is explicitly added to the respective clauses defining them.

## 1.2 Defect 1/2

This Technical Corrigendum changes the name of the reference “variableDecl” from class Variable to class VariableDecl to the new name “refvariable” (Figure 7 of Clause 5.3.2) and adjust the mapping of Table 8 of Clause 7.3.1 accordingly. This aligns the naming of the UML model with its name in the XML syntax and corrects the mapping, which was incorrect.

## 1.3 Defect 3

This Technical Corrigendum changes the defining occurrences of the class “BuiltInConst” to “BuiltInConstant” (in Figure 7 of Clause 5.3.2) in order to make it consistent with its uses later in other UML class diagrams and the RELAX/NG syntax.

## 1.4 Defect 4

This Technical Corrigendum changes the classes from which class Equality and Inequality in Figure 11 of Clause 5.3.4 are derived.

## 1.5 Defect 5

This Technical Corrigendum moves the name attribute that occurs in the sub classes of Declaration to the class Declaration itself in Figure 7 of Clause 5.3.2. The name attributes are removed from classes SortDecl, VariableDecl and OperatorDecl in Figure 7. This way, also the class Unparsed in package ArbitraryDeclarations in Figure 20 of Clause 5.3.11 has a name attribute.

## 1.6 Defect 6

This Technical Corrigendum corrects the data types of some attributes in the packages defining the PNML data types FiniteIntRanges, Strings, and Lists (in Figures 14 in Clause 5.3.8 and Figures 18 and 19 and in Clause 5.3.11).

## 1.7 Defect 7

This Technical Corrigendum changes an OCL constraint the class diagram of Figure 19 in Clause 5.3.11, which concerns the append operator on lists.

## 1.8 Defect 8

This Technical Corrigendum changes the mapping of Table 8 of Clause 7.3.1 for the class FiniteIntRangeConstant. The range attribute should be deleted.

### 1.9 Defect 9

This Technical Corrigendum makes explicit in Clause 7.1.1 that it is legal to define namespaces in PNML elements of a PNML document.

### 1.10 Defect 10

This Technical Corrigendum updates the definition of CyclicEnumeration class in Figure 12 of Clause 5.3.6 with two optional attributes, to prevent unecesseray enumeration of large intervals in their corresponding XML declaration. Table 8 in Clause 7.3.1 and the RELAX NG grammar in Annex B.2.5 are updated correspondingly.

### 1.11 Defect 11

This Technical Corrigendum updates the definition of Successor and Predecessor classes in Figure 13, Clause 5.3.7, to prevent the definition of a $n^{th}$ successor or predecessor to become a deep nested tree of those operations. Table 8 in Clause 7.3.1 and the RELAX NG grammar in Annex B.2.6 are updated correspondingly.

### 1.12 Defect 12

This Technical Corrigendum changes the RELAX/NG grammar for the IntegerOperators in Annex B.2.9 in order to fix an error.

### 1.13 Defect 13

This Technical Corrigendum removes the use of UML primitive data types and replaces it with XMLSchemaDatatypes so that all primitive data types consistently are XMLSchemaDataTypes. This affects Figure 4 in Clause 5.2.6, Figure 11 in Clause 5.3.3 and Figure 17 in Clause 5.3.11.

## 2 Changes

### 2.1 Update of the affected type URIs

At the end of Clause 5.3.1, as a last paragraph, insert

> The URI identifying a Petri net to be of type Place/Transition Net is
>
> `http://www.pnml.org/version-2009/grammar/ptnet`

At the end of Clause 5.3.10, as a last paragraph, insert

> The URI identifying a Petri net to be of type Symmetric Net is
>
> `http://www.pnml.org/version-2012/grammar/symmetricnet`

Replace all occurrences of the URI

> `http://www.pnml.org/version-2009/grammar/symmetricnet`

by the URI

> http://www.pnml.org/version-2012/grammar/symmetricnet

This concerns:

- The grammar “symmetricnet.pntd” in Annex B.2.14 at line 48
- The grammar “highlevel-net.pntd” in Annex B.2.15 at line 30
- The PNML listing in Annex C at line 2

At the end of Clause 5.3.11, as a last paragraph, insert

The URI identifying a Petri net to be of type High-level Petri Net Graph is

```
http://www.pnml.org/version-2012/grammar/highlevelnet
```

Replace all occurrences of the URI

```
http://www.pnml.org/version-2009/grammar/highlevelnet
```

by the URI

```
http://www.pnml.org/version-2012/grammar/highlevelnet
```

This concerns:

- The grammar “highlevelnet.pntd” in Annex B.2.15 at line 38

At the end of Clause 5.3.12, as a last paragraph, insert

The URI identifying a Petri net to be of type Place/Transition net as High-level Net Graph is

```
http://www.pnml.org/version-2009/grammar/pt-hlpng
```

## 2.2 Defect 1/2

In the UML class diagram of Figure 7 in Clause 5.3.2, the role of the association “variableDecl” of class VariableDecl is renamed to “refvariable”. In Table 8 in Clause 7.3.1 the last column of the line for Terms::Variable the entry “variabledecl: IDREF” is changed to “refvariable: IDREF”. This row in the table should read now:

| | | |
|---|---|---|
| Terms::Variable | variable | refvariable: IDREF |

## 2.3 Defect 3

In the UML class diagram of Figure 7 in Clause 5.3.2, the class “BuiltInConst” is renamed to “BuiltInConstant”.

### 2.4 Defect 4

In the UML class diagram in Figure 11 of Clause 5.3.4, the classes Equality and Inequality should be derived from class Boolean::Operator. The class Term::Operator can be deleted from this diagram since it is no longer used here.

### 2.5 Defect 5

In the UML class diagram of Figure 7 in Clause 5.3.2, add a name attribute to class Declaraion and remove the name attributes from the SortDecl, VariableDecl and OperatorDecl.

### 2.6 Defect 6

In the UML class diagram of Figures 14 in Clause 5.3.8 replace the type of the attribute value of FiniteRangeConstant and the type of the attributes start and end of class FiniteIntRange by the XML Schema type XMLSchemaDataTypes::Integer

In the UML class diagram of Figure 18 in Clause 5.3.11 replace the type of the attributes start and length in class Substring by XMLSchemaDataTypes:NonNegativeInteger

In the UML class diagram of Figure 19 in Clause 5.3.11 replace the type of the attributes start and length in class Sublist by XMLSchemaDataTypes:NonNegativeInteger and replace attribute “Index” of class MemberAtIndex by “index” (change to non-capital) and change the type to XMLSchemaDataTypes:NonNegativeInteger.

### 2.7 Defect 7

The OCL constraint for the class Append in the class diagram of Figure 19 in Clause 5.3.11 is replaced by

**context** Append **inv**:
self.input->size() = 2 and
self.input.->at(2).oclIsKindOf(List) and
self.output.oclIsKindOf(List)

### 2.8 Defect 8

In the mapping of Table 8 of Clause 7.3.1 for the class FiniteIntRangeConstant, remove therange:IDREF attribute from the last column. The row for FiniteIntRangeConstant should now read:

| | | |
|---|---|---|
| FiniteIntRanges::FiniteIntRangeConstan<br>t | finiteintrangeconstant | value: integer |

### 2.9 Defect 9

Insert as new last paragraph in Clause 7.1.1 the following text:

> In PNML documents, it is legal that namespaces are defined in other PNML elements, wherever XML would allow the definition of namespaces. These namespaces, however, should only be used by tool-specific extensions, which are not supposed to be shared among different tools. The default PNML namespace applies to the whole document.

## 2.10 Defect 10

In the UML class diagram of Figure 12 in Clause 5.3.6, add to class FiniteEnumeration two optional attributes (multiplicity 0..1) named *start* and *end,* having the datatype XMLSchemaDataTypes::Integer. Moreover, add the following OCL constraint to the diagram of Figure 12 :

**context** FiniteEnumeration **inv**:
( not self.elements.isUndefined()  and self.start.oclIsUndefined() and self.end.oclIsUndefined() ) or
( self.elements.isUndefined() and self.start <= self.end )

After the last paragraph of Clause 5.3.6 insert the following paragraph.

If the elements feature of *FiniteEnumeration* is undefined, the attributes start and end must be defined and start must be a value less or equal than the value of end. In that case, the set associated with the *FiniteEnumeration* sort is all integer values from start to end (including the start and end values). If the elements feature is defined, start and end should not be defined.

In Table 8 in Clause 7.3.1, replace the line for FiniteEnumerations::FiniteEnumeration (page 34) by

| | | |
|---|---|---|
| FiniteEnumerations::FiniteEnumeration | finiteenumeration | start, end: integer |

To the grammar for FiniteEnumerations in Annex B.2.5, apply the following changes :

Insert the following RELAX NG definition at line 36 :

```
<define name="FiniteEnumeration.range">
   <a:documentation>
      The content of a Finite Enumeration may also be specified as a
       range When the enumeration consists of a large interval
       (e.g. 1..5000), it is advised to use the attributes start and
       end provided in this definition.
   </a:documentation>
   <attribute name="start">
         <data type="integer"/>
   </attribute>
   <attribute name="end">
         <data type="integer"/>
   </attribute>
</define>
```

Replace the lines 37—44 by

```
<define name="FiniteEnumeration.content">
   <a:documentation>
       The content of a Finite Enumeration is composed of constant.
       They must be either enumerated, or specified as a range using
       FiniteEnumeration.range. Don't use both. In case both are used, the
       initial default definition and interpretation prevails, i.e. the
       enumerated values are considered rather than the attributes specifying
       a range.
   </a:documentation>
    <choice>
     <ref name="FiniteEnumeration.range"/>
     <zeroOrMore>
           <ref name="FEConstant"/>
     </zeroOrMore>
    </choice>
</define>
```

 

## 2.11 Defect 11

In the UML class diagram of Figure 13 in Clause 5.3.7, add an optional attribute (multiplicity 0..1) named *power* with data type XMLSchemaDataTypes::PositiveInteger to the classes *Successor* and *Predecessor*.

In Table 8 in Clause 7.3.1, replace the lines for CyclicEnumerations::Successor and for CyclicEnumerations::Predecessor (page 34) by

| | | |
|---|---|---|
| CyclicEnumerations::Successor | successor | power: integer |
| CyclicEnumerations::Predecessor | predecessor | power: integer |

Update the grammar for Cyclic Enumerations in Annex B.2.6 with the following:

Insert the following RELAX NG definition at line 62:

```
<define name="CyclicEnumOperator.power">
   <a:documentation>
       A cyclic enumeration operator may specify a rank, indicating for
       instance the nth successor or predecessor of a cyclic enumeration
       constant.
   </a:documentation>
   <attribute name="power">
         <data type="positiveInteger"/>
   </attribute>
</define>
```

Replace lines 63-70 by:

```
<define name="Successor">
   <a:documentation>
       Defines the 'successor' operator.
   </a:documentation>
    <element name="successor">
      <optional>
            <ref name="CyclicEnumOperator.power"/>
            </optional>
      <ref name=" CyclicEnumOperator.content"/>
    </element>
</define>
```

Replace lines 72-79 by:

```
<define name="Predecessor">
   <a:documentation>
       Defines the 'predecessor' operator.
   </a:documentation>
    <element name="predecessor">
      <optional>
            <ref name="CyclicEnumOperator.power"/>
            </optional>
      <ref name=" CyclicEnumOperator.content"/>
    </element>
</define>
```

## 2.12 Defect 12

In the grammar in Appendix B.2.9:

Replace the line in the operators LessThan (line 129), LessThanOrEqual (line 138), GreaterThan (line 147), GreaterThanOrEqual (line 156), Addition (line 165), Subtraction (line 174), Multiplication (line 183), Division (line 192), Modulo (line 201)

```
<ref name="IntegerOperator"/>
```

by the following line

```
<ref name="IntegerOperator.content"/>
```

## 2.13 Defect 13

Add class Boolean to package XMLSchemaDataTypes in Figure 4 of Clause 5.2.6. In the paragraph after the Figure 4, add the following sentence after the first sentence:

*Boolean* refers to the boolean values *true* and *false*.

In Figure 11 in Clause 5.3.3, replace the type of the value attribute of class BooleanConstant by XMLSchemaDataTypes::Boolean and remove the comment about the use of UML PrimitiveTypes package from the package.

In Figure 17 in Clause 5.3.11, replace the type of the value attribute of NumberConstant by XMLSchemaDataTypes::Integer and remove the comment about the use of UML PrimitiveTypes package from the package.