Schema validation error

- The error (bpmn-miwg.github.io/bpmn-miwg-tools/xsd/ARIS Architect 10.0.14/ARIS Architect 10.0.14-B.2.0-roundtrip.html)

ARIS Architect 10.0.14

B.2.0

ANALYSIS: ARIS Architect 10.0.14-B.2.0-roundtrip (Tool: xsd)
1.Error Line 1802: cvc-complex-type.2.4.c: The matching wildcard is strict, but no declaration can be found for element 'deco:BPMNEnvelopeDecorator'.

INFO : Validation failed: Warnings: 0, Errors: 1, Fatal Errors: 0

BPMN File

```
1799            </bpmndi:BPMNEdge>
1800            <bpmndi:BPMNEdge bpmnElement="_09e7cb23-4a1b-4165-b93a-cf635c223ee5" messageVisibleKind="initiating" id="Edge__09e7ck
1801                <di:extension>
1802                    <deco:BPMNEnvelopeDecorator>
1803                        <dc:Bounds height="19.84" width="28.35" x="168.09" y="246.61"/>
1804                    </deco:BPMNEnvelopeDecorator>
1805                </di:extension>
1806                <di:waypoint x="248.03" y="140.88"/>
1807                <di:waypoint x="248.03" y="256.54"/>
1808                <di:waypoint x="116.79" y="256.54"/>
1809                <di:waypoint x="116.79" y="488.13"/>
1810                <bpmndi:BPMNLabel labelStyle="LabelStyle_0_0" color:color="#000000">
1811                    <dc:Bounds x="144.57" y="269.57" width="71.15" height="9.64"/>
1812                </bpmndi:BPMNLabel>
1813            </bpmndi:BPMNEdge>
1814            <bpmndi:BPMNEdge bpmnElement="_f61be5ab-acb2-4348-a9a0-bdfdde0c42ad" id="Edge__f61be5ab-acb2-4348-a9a0-bdfdde0c42ad"
```

Extension element is used for BPMN 2.0 Extension for Envelope Decorator Bounds

(https://github.com/bpmn-miwg/bpmn-deco)

BPMN 2 Schemas

DI.xsd

```
<xsd:complexType abstract="true" name="DiagramElement">
    <xsd:sequence>
        <xsd:element name="extension" minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

Attribute processContents not set

Semantic.xsd

```
<xsd:element name="extensionElements" type="tExtensionElements" />
<xsd:complexType name="tExtensionElements">
    <xsd:sequence>
        <xsd:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
```

Attribute processContents set to “lax“

Schema validation

- If processContents is not set, then “strict“ is used as default.
- ProcessContents “strict” means that the extension element will be validated.
  This requires a schema for the extension.
- ProcessContents “lax” means that the extension element will be only validated if a schema is provided, otherwise the element will not be validated.

XSD Schema (1)

W3C Recommendation

XML Representation Summary: `any` Element Information Item

```
<any
  id = ID
  maxOccurs = (nonNegativeInteger | unbounded)  : 1
  minOccurs = nonNegativeInteger : 1
  namespace = ((##any | ##other) | List of (anyURI | (##targetNamespace | ##local)) )  : ##any
  processContents = (lax | skip | strict) : strict
  {any attributes with non-schema namespace . . .}>
  Content: (annotation?)
</any>
```

A particle containing a wildcard, with properties as follows (unless `minOccurs=maxOccurs=0`, in which case the item corresponds to no component at all):

Particle Schema Component

Property	Representation
{min occurs}	The ·actual value· of the `minOccurs` [attribute], if present, otherwise `1`.
{max occurs}	*unbounded*, if the `maxOccurs` [attribute] equals *unbounded*, otherwise the ·actual value· of the `maxOccurs` [attribute], if present, otherwise `1`.
{term}	A wildcard as given below:

Wildcard Schema Component

Property	Representation
{namespace constraint}	Dependent on the ·actual value· of the `namespace` [attribute]: if absent, then *any*, otherwise as follows: **##any** *any* **##other** a pair of *not* and the ·actual value· of the `targetNamespace` [attribute] of the <schema> ancestor element information item if present, c ***otherwise*** a set whose members are namespace names corresponding to the space-delimited substrings of the string, except 1 if one such substring is `##targetNamespace`, the corresponding member is the ·actual value· of the `targetNamespace` [attribute] of the · element information item if present, otherwise ·absent·. 2 if one such substring is `##local`, the corresponding member is ·absent·.
{process contents}	The ·actual value· of the `processContents` [attribute], if present, otherwise *strict*.
{annotation}	The annotation corresponding to the <annotation> element information item in the [children], if present, otherwise ·absent·.

XSD Schema (2)

W3C Recommendation

3.10.3 Constraints on XML Representations of Wildcards

Schema Representation Constraint: Wildcard Representation OK

In addition to the conditions imposed on <any> element information items by the schema for schemas, the corresponding particle and model group must satisfy the conditions set out in Constraints on Model Group Schema Components (§3.8.6) and Constraints on Particle Schema Components (§3.9.6).

3.10.4 Wildcard Validation Rules

Validation Rule: Item Valid (Wildcard)

For an element or attribute information item to be locally ·valid· with respect to a wildcard constraint its [namespace name] must be ·valid· with respect to the wildcard constraint, as defined in Wildcard allows Namespace Name (§3.10.4).

When this constraint applies the appropriate **case** among the following must be true:

1 **If** {process contents} is *lax*, **then** the item has no ·context-determined declaration· with respect to Assessment Outcome (Element) (§3.3.5), Schema-Validity Assessment (Element) (§3.3.4) and Schema-Validity Assessment (Attribute) (§3.2.4).
2 **If** {process contents} is *strict*, **then** the item's ·context-determined declaration· is *mustFind*.
3 **If** {process contents} is *skip*, **then** the item's ·context-determined declaration· is *skip*.

Questions, Solutions

- **Question**

1. Is the schema of the BPMN-Deco-Extension included in the schema validation?

- **If not, possible solutions**

1. Adapt BPMN-Schema DI.xsd (processContents="lax")
2. Adapt Schema Validation of BPMN-MIWG-Tests (include the schema for BPMN-Deco-Extension)
3. Adapt the BPMN-Deco-Extension (do not use the extension element)
4. Do not support the BPMN-Deco-Extension (by ARIS)