

Stand: 23.01.2018

## XUBETRIEB-MODULNUTZUNG

Das MagicDraw UML2-Modul "XUBetrieb-Modul" wird als Modul in das Paket „Data::Externe Modelle“ eingehängt.

Damit stehen als wesentliche Bausteine die UML-Klassen mit den fachlichen Objekten von Abfallverbringung bis Zeitraum und die Codelisten von Code.Abwasserbehandlungsart bis Code.Zehrstoff aus XUBetrieb zur Verfügung.

In der folgenden Abbildung sind zwei Entitäten „Abwasserbehandlung.Klaeranlage“ und „Abwasserbehandlung.IAS“ (IAS: Individuelle AbwasserSysteme) aus der XUBetrieb-Modellkomponente „Abwasserbehandlung“ abgeleitet worden. Dabei kam für die Modellierung der Beziehung das UML-Element „Generalization“ zum Einsatz, das mit den Stereotyp «xsdRestriction» versehen wurde.\

Dadurch werden die abgeleiteten Entitäten in den per XGenerator erstellten XML-Schemata über das entsprechende XML-Schema-Konstrukt „xsdRestriction“ aus der XUBetrieb-Entität gebildet.

Bei der Nutzung der Ableitung per Generalization «xsdRestriction» müssen folgende Randbedingungen beachtet werden:

- Alle Kardinalitäten dürfen nur weiter eingeschränkt werden
- Attribute mit einer Kardinalität von 1 oder mehr dürfen nicht entfallen (implizit)
- Alle Typen dürfen nur die Orignaltypen sein oder eine Einschränkung der Originaltypen
- Es dürfen keine Attribute hinzukommen
- Es darf auch das Attribut im derived type nicht nillable sein, wenn das Attribut im base type nicht nillable ist oder nicht existiert. Das ist bei XÖV jeweils im tag value «xsdElement»nillable abgelegt.
- Die Reihenfolge der Attribute über «xsdElement»position muss exakt übereinstimmen. Achtung: Die Parserfehlermeldungen sind irreführend, wenn die Reihenfolge nicht eingehalten wird. Das betrifft Meldungen der Art "Element X ist im derived type erlaubt aber im base type nicht". Diese sind sehr wohl erlaubt, aber nicht an der entsprechenden Stelle.
- Es sollten zur Validierung sowohl die Parser Xerxes als auch Saxon eingesetzt werden, da bei unterschiedlichen Fehlerbedingungen die Qualität der Fehlermeldungen der Parser unterschiedlich ist!

# Fehler und Lösungen Baukasten-Klassen

## NDR-28: Valide W3C-XML-Schemata

- The attribute grundwasserkörper of class Data.XUKommunalabwasser.Baukasten.WRRL.0001 violates the constraint 'Property.ElementAndAttributeTypesMustBeVisible'.
- Für UML-Klasseneigenschaften verwendete «xsdNamedType»-Typen müssen in diesem «xsdSchema»-UML-Paket mit «xsdInclude» oder «xsdImport» sichtbar sein.
- **Lösung:** Es muss eine Beziehung geben, mit der das bemängelte Paket die verwendeten Datentypen in das XML-Schema einbindet. Üblicherweise ist das per «xsdImport» oder -- wenn es der selbe Namespace ist -- per «xsdInclude». Das macht man dann in der Modellübersicht, in dem man eine Dependency vom Paket auf das vererbende Paket zieht und ihr danach den entsprechenden Stereotyp zuweist. Beispiel: XUK::Berichtspflichten auf XUB::Baukasten

## NDR-23: Umgang mit Restriktionen über unterschiedliche Namensräume

- The generalization between Abwasserbehandlung and Abwasserbehandlung.0002 violates the constraint 'Generalization.TransNamespaceRestrictionImpliesUnqualifiedElements'.
- Bei Restriktionen über mehrere Namensräume müssen die UML-Klasseneigenschaften mit «xsdElement» für die Eigenschaft "form" den Wert "unqualified" aufweisen.
- **Lösung:** Im Modell des Generalization-Teils (meist eine XUBetrieb-Modellklasse) sind für die Attribute, die den «xsdElement»-Stereotyp tragen, der aus dem Stereotyp stammende Tag-Value "form" auf unqualified zu setzen. Ob das auf beiden Seiten passieren muss, ist unklar. Jedenfalls reicht es nicht, das nur auf der Seite der die Generalization nutzenden Klasse zu machen.

## Fehler und Lösungen Listen

- Alle Listen (Code.X) per Generalization «xsdRestriction» von Code abgeleiten
- In allen Listen
  - «xsdCode» tag listAgencyName der Klasse Code.X füllen
  - «xsdCode» tag listName der Klasse Code.X füllen
- Herstellung eines sauberen code-Attributs:
  - das content-Attribut in code umbenennen
  - das code-Attribut bzw. konkret – dessen Typ – mit der korrespondierenden Enumeration «xsdCodeList» verbinden

- Das Attribut code bzw. – konkret dessen aus «xsdElement» stammendes Tag form – von qualified auf unqualified setzen
- Herstellung eines sauberen listURI-Attributs:
  - listURI Attribut der Codeliste hinzufügen
  - listURI Attribut mit Stereotyp «xsdAttribute» versehen
  - listURI mit Typ anyURI versehen
  - default value des listURI Attributs mit Typ Literal String versehen
  - default value des listURI Attributs mit einer URI befüllen
  - listURI multiplicity auf 0..1 setzen
  - listURI "is read only" auf true setzen
- Herstellung eines sauberen listVersionID Attributs:
  - listVersionID Attribut wird der Codeliste hinzugefügt
  - listVersionID Attribut mit Stereotyp «xsdAttribute» versehen
  - listVersionID wird mit Typ normalizedString versehen
  - default value des listVersionID Attributs wird mit Typ Literal String versehen
    - default value des listVersionID Attributs wird mit einer Version befüllt
      - listVersionID multiplicity auf 0..1 setzen
      - listVersionID "is read only" auf true setzen
- Bei Nutzung der KoSIT-DocBook-Werkzeuge nun in Datei O/src/informationsmodell/codesUndCodelisten.xml die neue Liste mit
 

```
<include xmlns="http://www.w3.org/2001/XInclude"
href="https://www.umweltbundesamt.de/../../build/docBook/dokumente
/anhang/code..."/>
```

 eintragen

Bei Nichtbeachtung drohen folgende Fehler:

- ruleName = NDR-4: Erlaubte Einbindungsarten für Codelisten
  - message = The class Data.XUBetrieb.Baukasten.Codes.Code.Abwasserbehandlungsverfahren violates the constraint 'Class.CodeTypesConformToXOEVCodes'.
    - documentation =
 

Ein Code-Datentyp (UML-Klasse mit dem Stereotyp «xsdCode») entspricht mit den Namen, Multiplizitäten und Fix-Werten den Vorgaben zu der Einbindungsart 1, 2, 3 oder 4 von Codelisten.
- ruleName = NDR-23: Umgang mit Restriktionen über unterschiedliche Namensräume
  - message = The generalization between Code and Code.Abwasserbehandlungsverfahren violates the constraint 'Generalization.TransNamespaceRestrictionImpliesUnqualifiedElements'.
    - documentation =
 

Bei Restriktionen über mehrere Namensräume müssen die UML-Klasseneigenschaften mit «xsdElement» für die Eigenschaft "form" den Wert "unqualified" aufweisen.
- ruleName = NDR-28: Valide W3C-XML-Schemata

- message = The class

Data.XUBetrieb.Baukasten.Codes.Code.Abwasserbehandlungsverfahren violates the constraint 'Class.PropertiesOfNamedTypesAndGlobalElementsAreLocalStructuresElementsOrAttributes'.

- documentation =

Alle UML-Klasseneigenschaften einer «xsdNamedType»- oder «xsdGlobalElement»-UML-Klasse müssen als «xsdLocalStructure», «xsdElement» oder «xsdAttribute» gekennzeichnet sein. Eigenschaften ohne einen dieser Stereotypen sind nicht zulässig.

- ruleName = NDR-28: Valide W3C-XML-Schemata

- message = The attribute listURI of class

Data.XUBetrieb.Baukasten.Codes.Code.Abwasserbehandlungsverfahren violates the constraint 'Property.AttributeTypesBasedOnW3CDatatypes'.

- documentation =

Der Typ einer UML-Klasseneigenschaft mit dem Stereotyp «xsdAttribute» ist entweder direkt ein W3C-Datentyp bzw. eine Ableitung eines W3C-Datentyps.

- ruleName = NDR-28: Valide W3C-XML-Schemata

- message = The attribute listURI of class

Data.XUBetrieb.Baukasten.Codes.Code.Abwasserbehandlungsverfahren violates the constraint 'Property.InlinePropertiesMustBeTyped'.

- documentation =

Eine UML-Klasseneigenschaft mit dem Stereotyp «xsdElement» oder «xsdAttribute», die als Inline-Klasseneigenschaft innerhalb einer UML-Klasse und nicht als UML-Assoziationsende über einen Rollennamen definiert ist, muss einen Typ besitzen.

