Handreichung zur technischen Umsetzung der XInneres-Basisnachricht mit JAXB

Fassung vom 15.06.2020

Bei der Umsetzung der Basisnachricht aus dem XInneres-Basismodul mit JAXB besteht ein Problem, das zu *Exceptions* führen kann. Diese Handreichung informiert über das Problem und Lösungsmöglichkeiten.

*Hintergrund:* Ein API in Java oder .NET (und vergleichbaren typsicheren, objektorientierten Programmiersprachen) kann nicht eins-zu-eins einem XML Schema entsprechen, wenn XML-Restrictions auf komplexen Typen zum Einsatz kommen. Dieses Problem ist grundsätzlich unauflösbar: Es können im Allgemeinen nicht die Restriction-Beziehungen durch Subtyp-Beziehungen in der Programmiersprache dargestellt und gleichzeitig passende Schnittstellen für die abgeleiteten OO-Typen erreicht werden.

# Problembeschreibung

Folgende Nachrichtenkonstruktion ist im XInneres-Basismodul vorgegeben:



Im weiteren Verlauf des Dokuments werden die in der Abbildung aufgeführten Typen wie folgt bezeichnet:

* Bn = Nachricht.G2G (aus Basismodul)
* Fn = Nachricht.G2G (aus Fachmodul)
* Bnk = Nachrichtenkopf.G2G (aus Basismodul)
* Fnk = Nachrichtenkopf.G2G (aus Fachmodul)

Die Typen *Bn* und *Bnk* wurden im Basismodul abstrakt deklariert, da sie nicht direkt in den Fachmodulen verwendet werden dürfen. Es müssen Ableitungen in den Fachmodulen entsprechend der obigen Abbildung mit den Typen *Fn* und *Fnk* gebildet werden.

Es hat sich gezeigt, dass JAXB mit dieser Konstruktion Probleme hat. Nach Erzeugung der JAVA-Klassen mit *xjc (kompilieren)* ergibt sich eine Form der sogenannten Kovarianz. Zur Laufzeit versucht JAXB daraufhin beim Unmarshalling einer XML-Nachricht, die abstrakte Klasse *Bnk* zu instanziieren, womit eine *Instantiation-Exception* entsteht.

# Problemlösungsvariante ohne JAXB-Bindung-Datei

Eine einfache Umgehung des Problems besteht darin, die Deklaration abstrakt des Typs Bnk aus der Schemadatei xinneres-basisnachricht.xsd manuell zu entfernen, bevor die JAVA-Klassen mit dem JAXB-Tool xjc erzeugt werden:

Vorher: <xs:complexType name="Nachrichtenkopf.G2G" **abstract="true"**>

Nachher: <xs:complexType name="Nachrichtenkopf.G2G">

# Problemlösungsvariante mit JAXB-Bindingsdatei

Der JAXB-Compiler bietet mit sogenannten Bindings-Dateien die Möglichkeit, das Unmarshaling von XML-Dokumenten zu beeinflussen. Solch ein Binding kann eingesetzt werden, um das Problem ohne eine manuelle Modifizierung der Schamadatei *xinneres-basisnachricht.xsd* zu umgehen.

Die das Problem lösende Bindings-Datei schreibt beim Unmarshalling vor, immer wenn der Typ eines Feldes Bnk ist, eine manuell vorkompilierte Instanz des Nachrichtenkopfes zu verwenden. Damit muss keine abstrakte Klasse instanziiert werden.

Hierfür muss in JAVA eine Klasse manuell erstellt werden, die von der abstrakten Klasse *Bnk* ableitet. Die JAVA-Klasse sieht wie folgt aus:

**package** de.osci.xinneres.basisnachricht.impl;

**import** de.osci.xinneres.basisnachricht.\_4.NachrichtenkopfG2G;

**public** **class** NachrichtenkopfImpl **extends** NachrichtenkopfG2G {

}

Das folgende JAXB-Binding bindet statt der Klasse *Bnk* die manuell erzeugte Klasse *NachrichtenkopfImpl*. Diese Klasse ist nicht abstrakt, kann damit instanziiert und durch die Vererbungsbeziehung zu *Bnk* stattdessen verwendet werden (Substitutionsprinzip). Die *InstantiationException* beim unmarshalling der XML-Nachricht tritt damit nicht mehr auf.

<jxb:bindings version="1.0"

 xmlns:jxb="http://java.sun.com/xml/ns/jaxb"

 xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"

 xmlns:xs="http://www.w3.org/2001/XMLSchema">

<jxb:bindings schemaLocation="xinneres-basisnachricht.xsd" >

 <jxb:bindings node="//xs:complexType[@name='Nachrichtenkopf.G2G']">

 <jxb:class

 name="Nachrichtenkopf.G2G"

 implClass="de.osci.xinneres.basisnachricht.impl.NachrichtenkopfImpl" />

 </jxb:bindings>

</jxb:bindings>

</jxb:bindings>