



# MID Insight'09

Better Together - Teamwork mit Modellen

## Integration von openArchitectureWare in Innovator am Beispiel eines WSDL/XSD-Generators für SOA- Plattformen

Nürnberg, 10.11.2009

# INHALT

1. **Was ist openArchitectureWare?**
2. **Die Integration von oAW in Innovator**
3. **Erstellung von oAW-Schablonen**
4. **Die WSDL/XSD-Generierung**
5. **Einsatz von oAW am Beispiel**



**Was ist openArchitectureWare?**

**openArchitectureWare ist ein Generator-Framework zum Erstellen von Textdateien aus Modellen.**

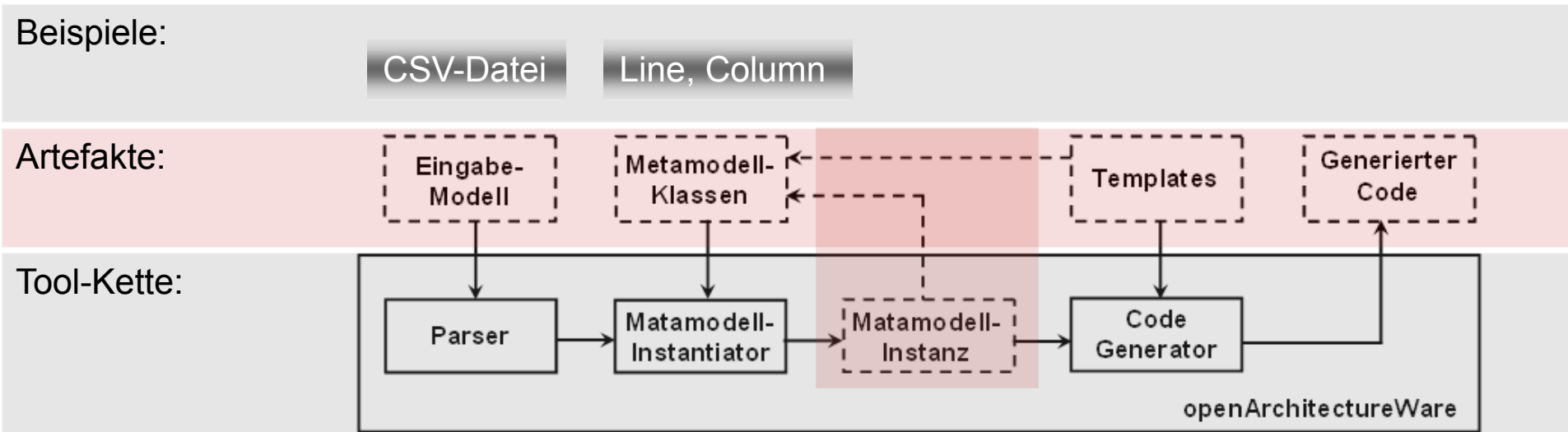
## openArchitectureWare

- Plattform für modellgetriebene Softwareentwicklung
- Aktuell: Version 5
- Seit Version 5 im „Eclipse Modeling Project“ integriert und somit Bestandteil von Eclipse
- [www.openarchitectureware.org](http://www.openarchitectureware.org)
- Deutschsprachiges Forum: [oaw.itemis.de](http://oaw.itemis.de)



**oAW arbeitet Workflow-basiert.**

## Der grundlegende oAW-Workflow



## oAW arbeitet schablonenbasiert.

### Generierung von Textdateien aus Schablonen

- Plain Text in den Schablonen erscheint eins zu eins im Generat
- Die Generierung wird innerhalb der Schablone durch die Sprache Xpand gesteuert
  - Gliederung der Schablonen in Blöcke und Aufruf dieser Blöcke
  - Steuerung der Dateierzeugung
  - Bedingungen
  - Schleifen
  - Funktionen zum Zugriff auf Listen
  - Typsystem
  - Codeschutz-Blöcke
  - ...
- Einfacher Zugriff auf Modellelemente aus der Schablone heraus
- Hilfsfunktionen (sog. Extensions) lassen sich mit der Sprache Xtend erzeugen
- Von den Schablonen aus können über Hilfsfunktionen statische Methoden eigener Java-Klassen aufgerufen werden



## Die Integration von oAW in Innovator

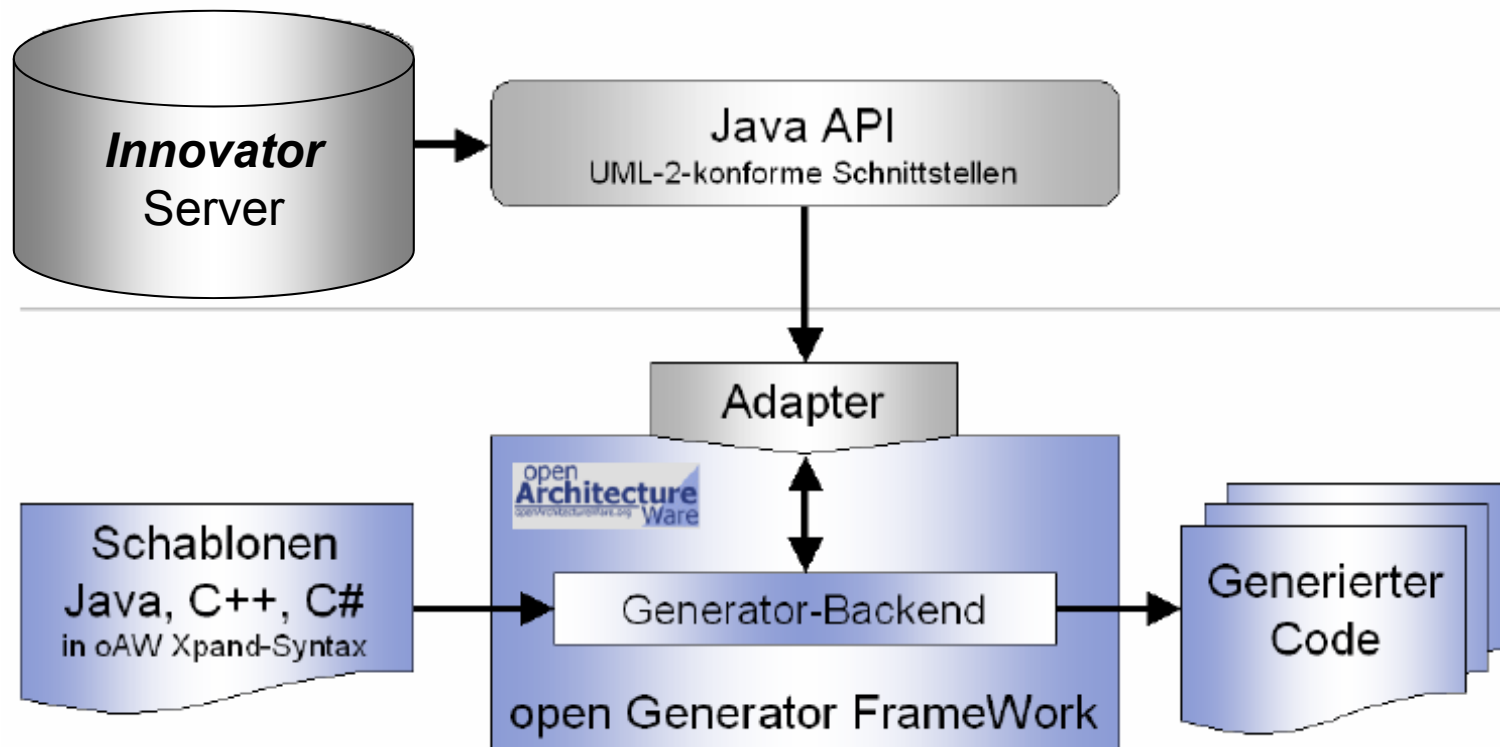
---

**MID stellt ein integriertes oAW-Framework zur Verfügung, um Text-zu-Modell-Transformationen durchzuführen.**

- Das integrierte oAW-Framework greift über das Innovator-Metamodell direkt auf das Innovator-Repository zu
- Zur Zeit Unterstützung der oAW-Version 3.4.1
- oAW wird langfristig Bestandteil von Innovator bleiben

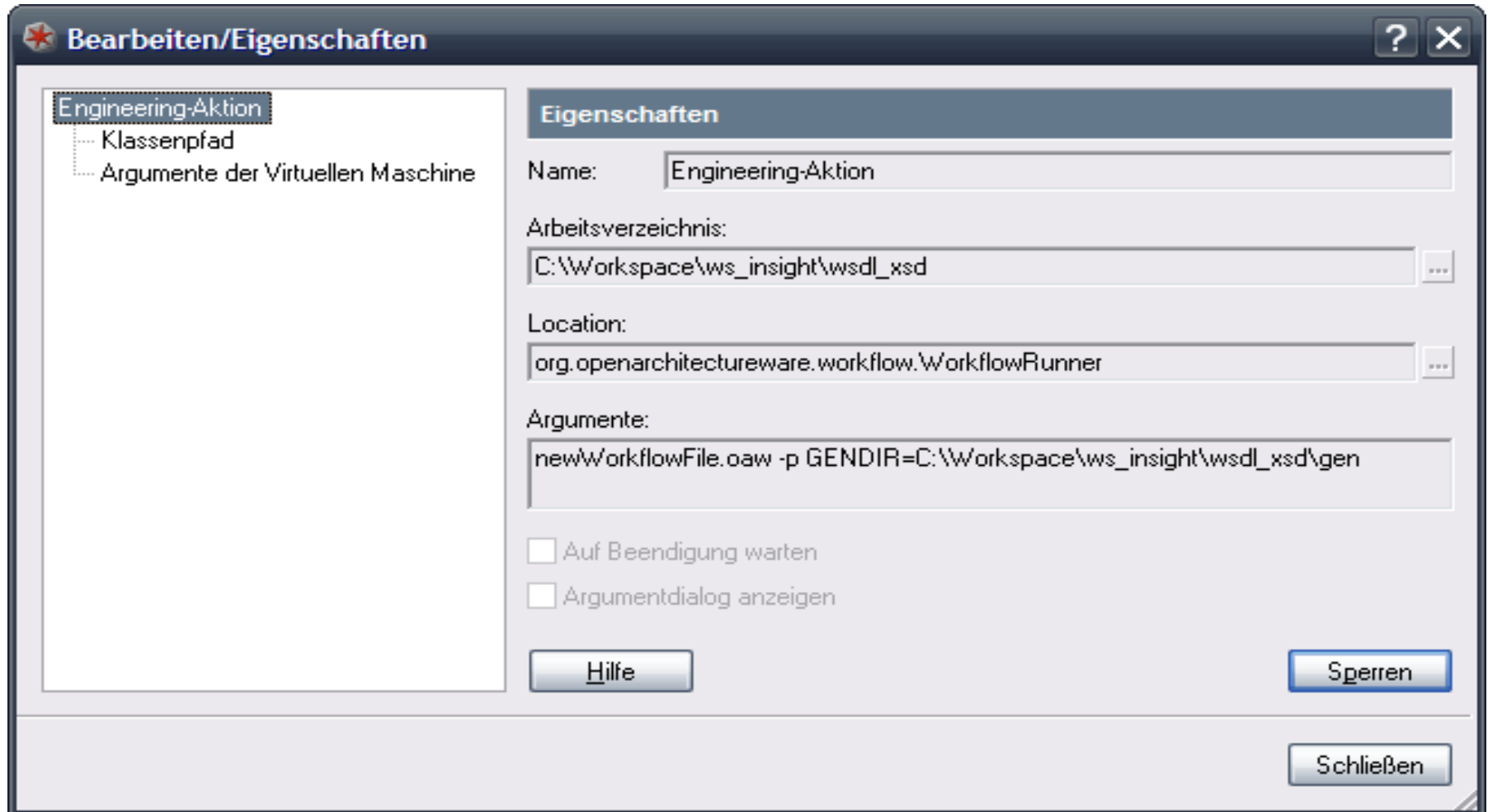
**oAW ist über einen Adapter direkt mit dem Innovator-Repository verbunden.**

## Informationsfluss und Komponenten von Innovator und oAW



## Die oAW-Generierung wird als Engineering-Aktion konfiguriert.

### Konfiguration der Engineering-Aktion



**Zur Syntaxprüfung kann das Innovator-Metamodell in Eclipse bekannt gemacht werden.**

## Hinweise zur Eclipse-Einrichtung

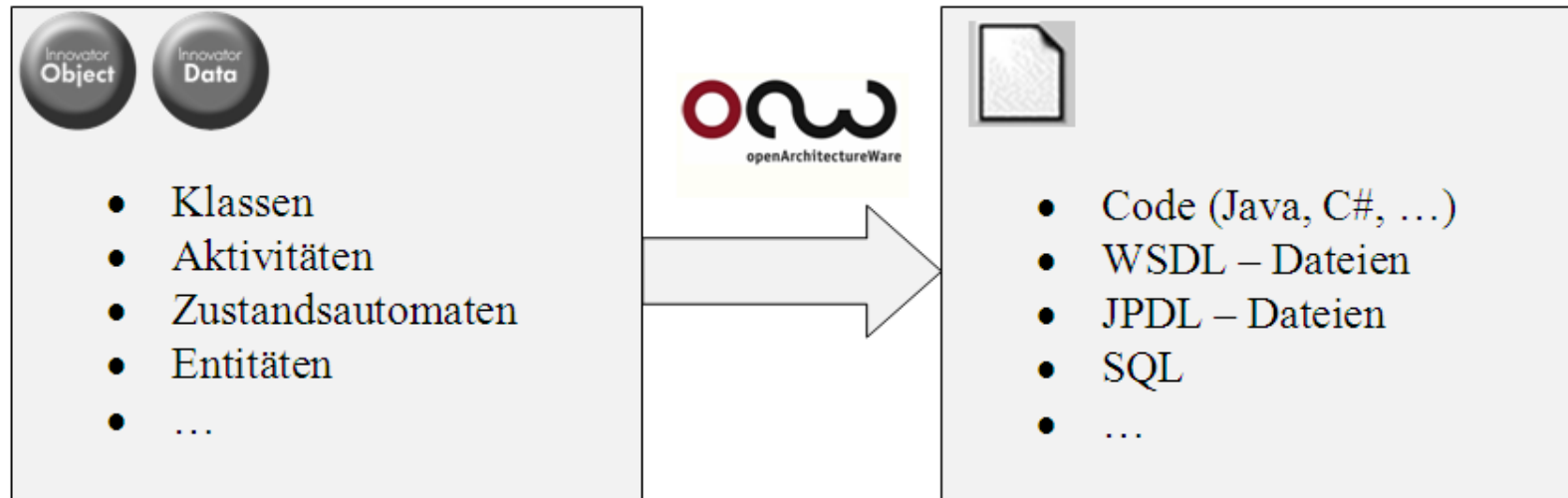
- Plugin über Update-Manager installierbar
  - <http://eclipse.innovator.de/updatesite/10.1.1/>
- Zur Zeit unterstützt das Plugin die oAW-Version 3.4.1 → Eclipse 3.4 (Ganymede) verwenden
- Schablonen müssen sich in einem Source-Folder befinden, um sie zu prüfen
- Neue Projekte als Java-Projekte erstellen und nachträglich die oAW-Nature hinzufügen
- In Projekteigenschaften das Innovator-Metamodell statt des EMF-Metamodells aktivieren



## **Erstellen von Generatoren**

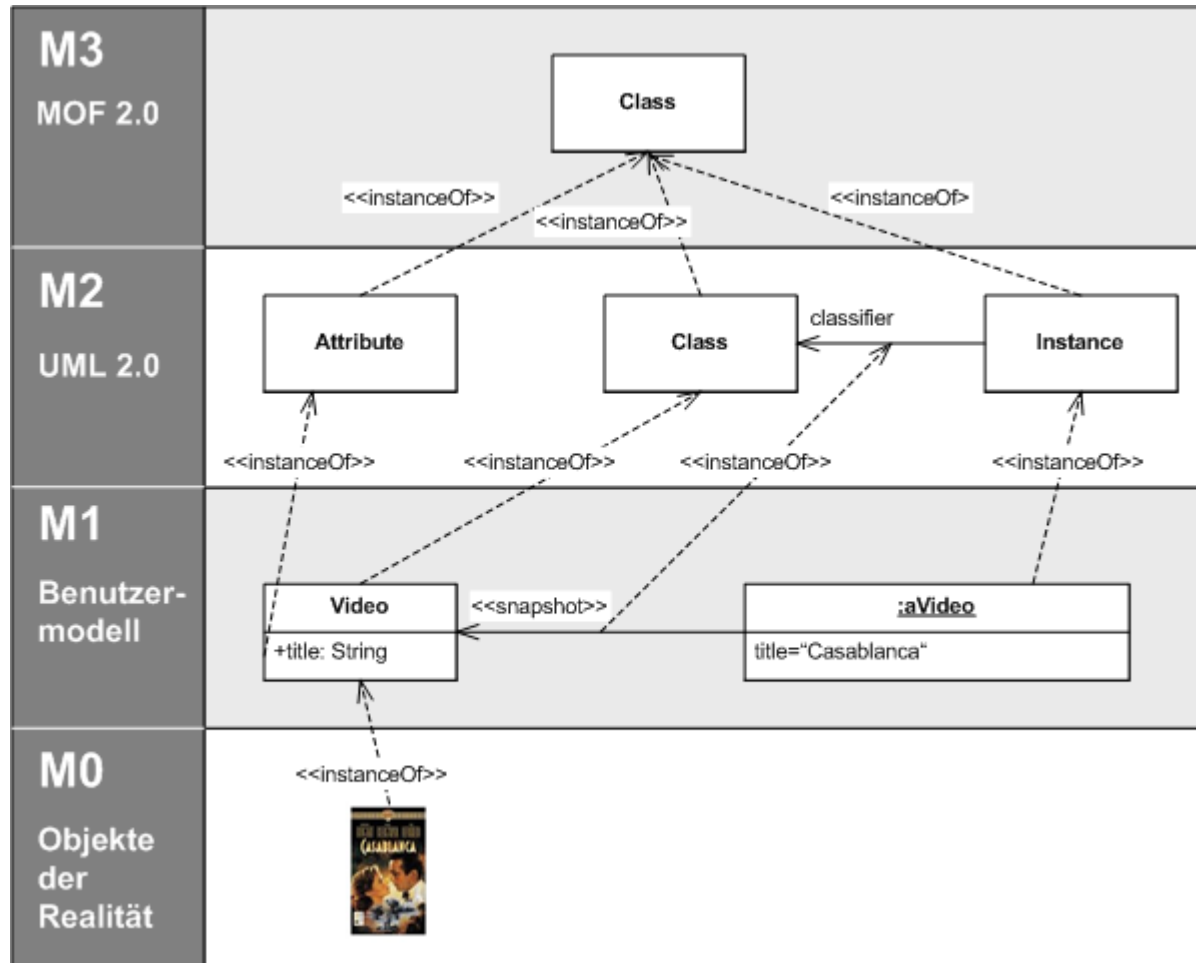
Ziel der Generatoren ist die Erstellung von Text-Dokumenten aus Innovator-Modellen.

## openArchitectureWare

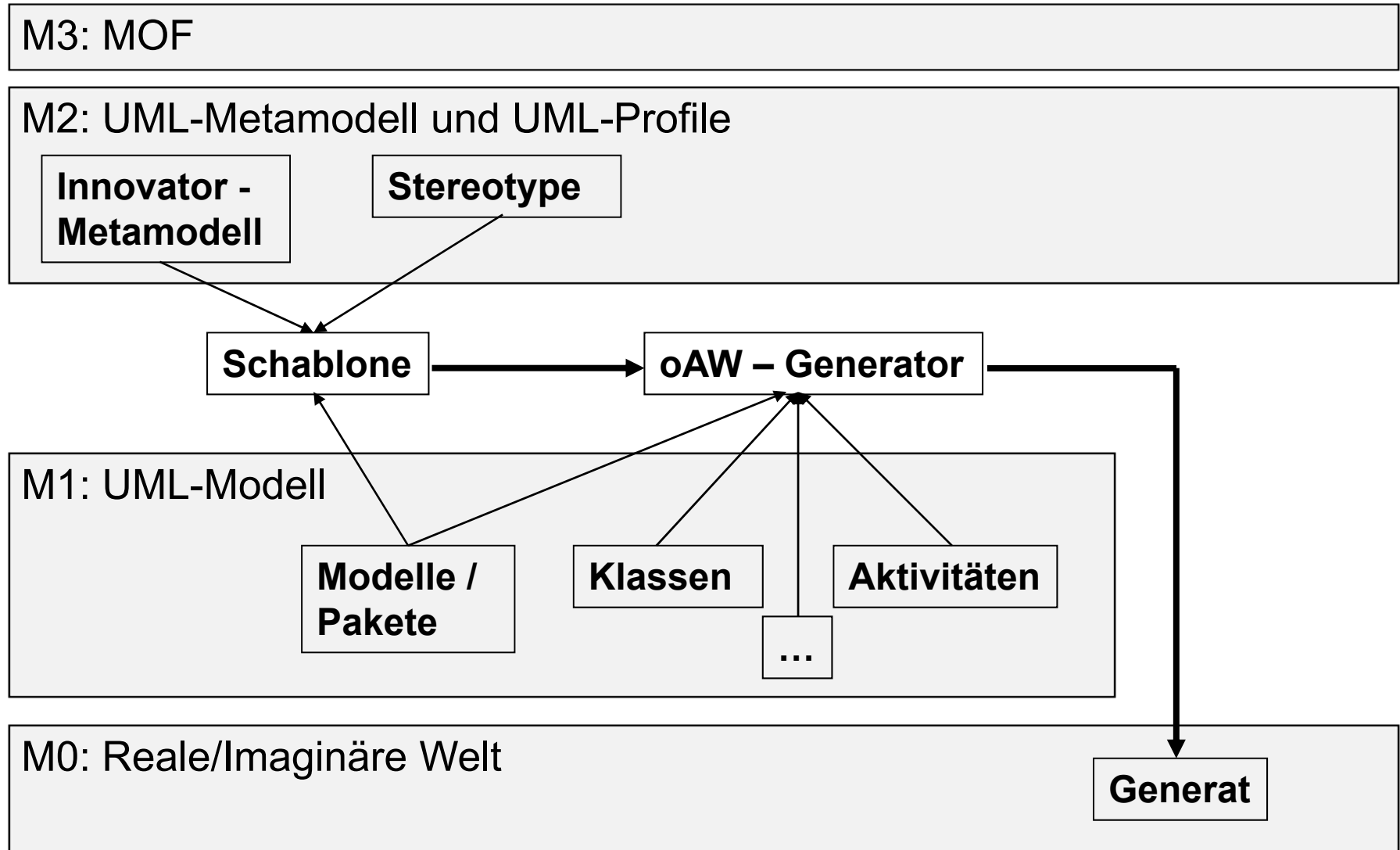


## Den Innovator-Modellen liegt das UML-Metamodell zugrunde.

### Die Schichten des UML-Metamodells



Die Generatoren benötigen Informationen aus dem Metamodell und aus dem Modell.



Aus den oAW-Schablonen heraus kann direkt auf das Innovator-Modell zugegriffen werden.

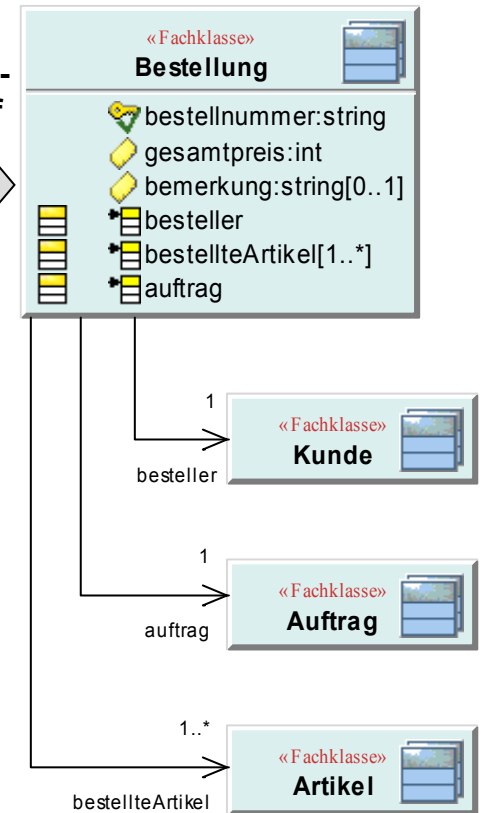
## Zugriff auf Modellelemente in einer oAW-Schablone

```
«DEFINE complexTypeForBox(boolean generateObjectType) FOR CLClass-»  
<xsd:complexType name="«Name»Type">  
  <xsd:sequence>  
    «EXPAND Property::xsdProperty FOREACH OwnedAttribute-»  
  </xsd:sequence>  
</xsd:complexType>  
«ENDDFINE»
```

Generierung

```
<xsd:complexType name="BestellungType">  
  <xsd:sequence>  
    <xsd:element name="bestellnummer" type="xsd:string"/>  
    <xsd:element name="gesamtpreis" type="xsd:int"/>  
    <xsd:element name="bemerkung" minOccurs="0" type="xsd:string"/>  
    <xsd:element name="besteller" type="tns:KundeType"/>  
    <xsd:element name="bestellteArtikel" maxOccurs="unbounded",  
      type="tns:ArtikelType"/>  
    <xsd:element name="auftrag" type="tns:AuftragType"/>  
  </xsd:sequence>  
</xsd:complexType>
```

Modell-  
zugriff

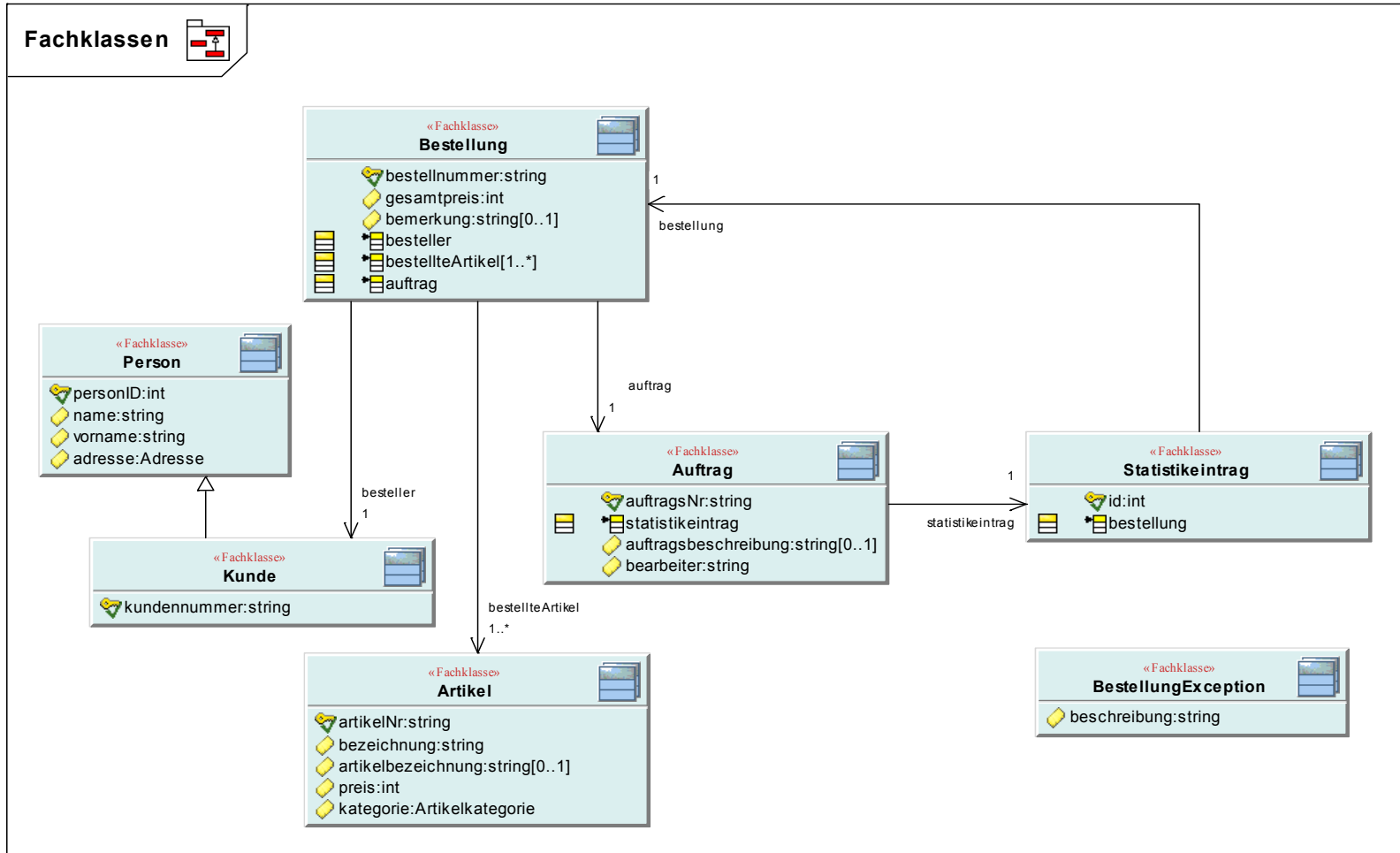




## Die WSDL/XSD – Generierung

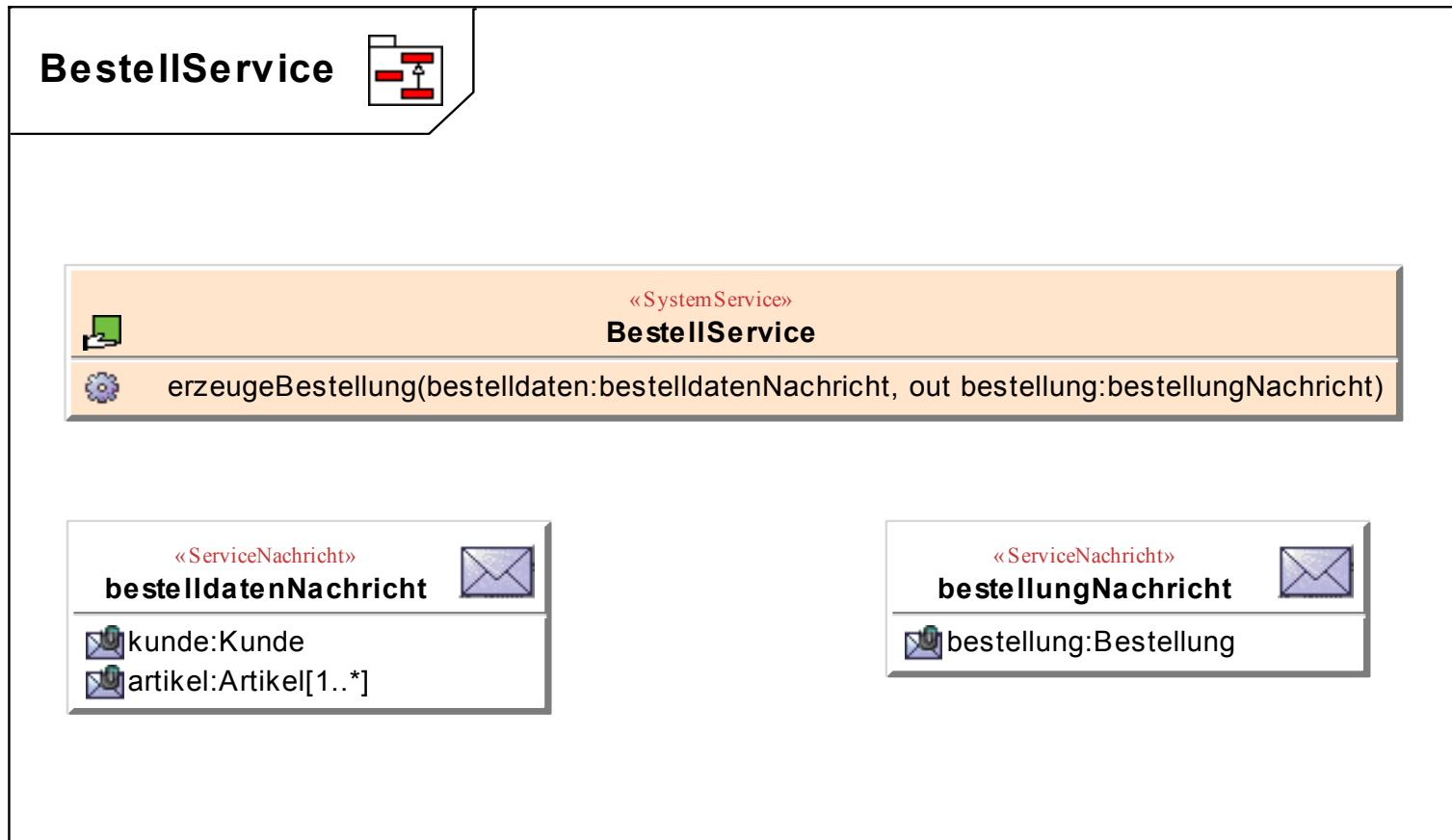
## Ausgangspunkt ist die Modellierung von Fachklassen und Services.

### Modellierung der Fachklassen



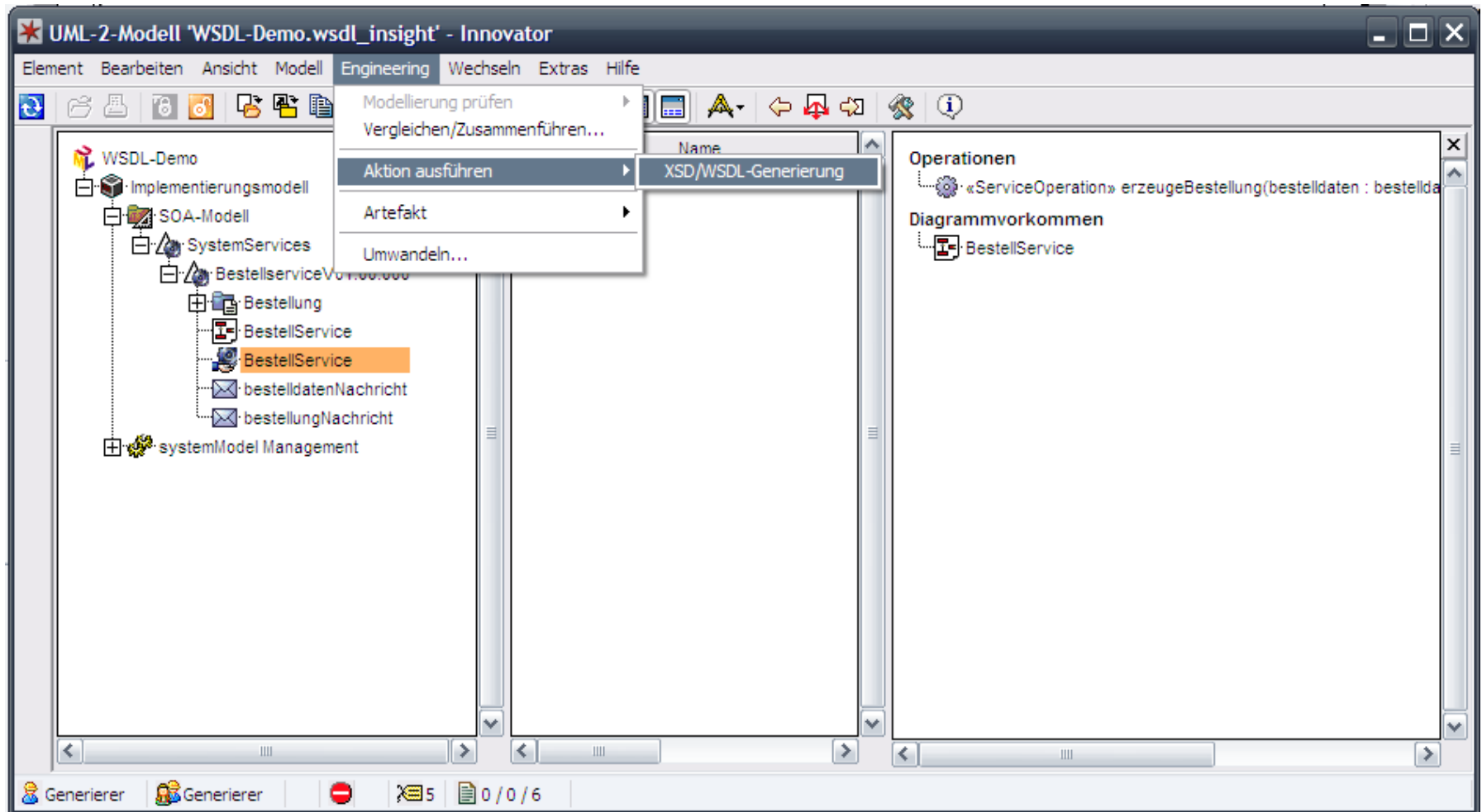
Ausgangspunkt ist die Modellierung von Fachklassen und Services.

## Modellierung des Services



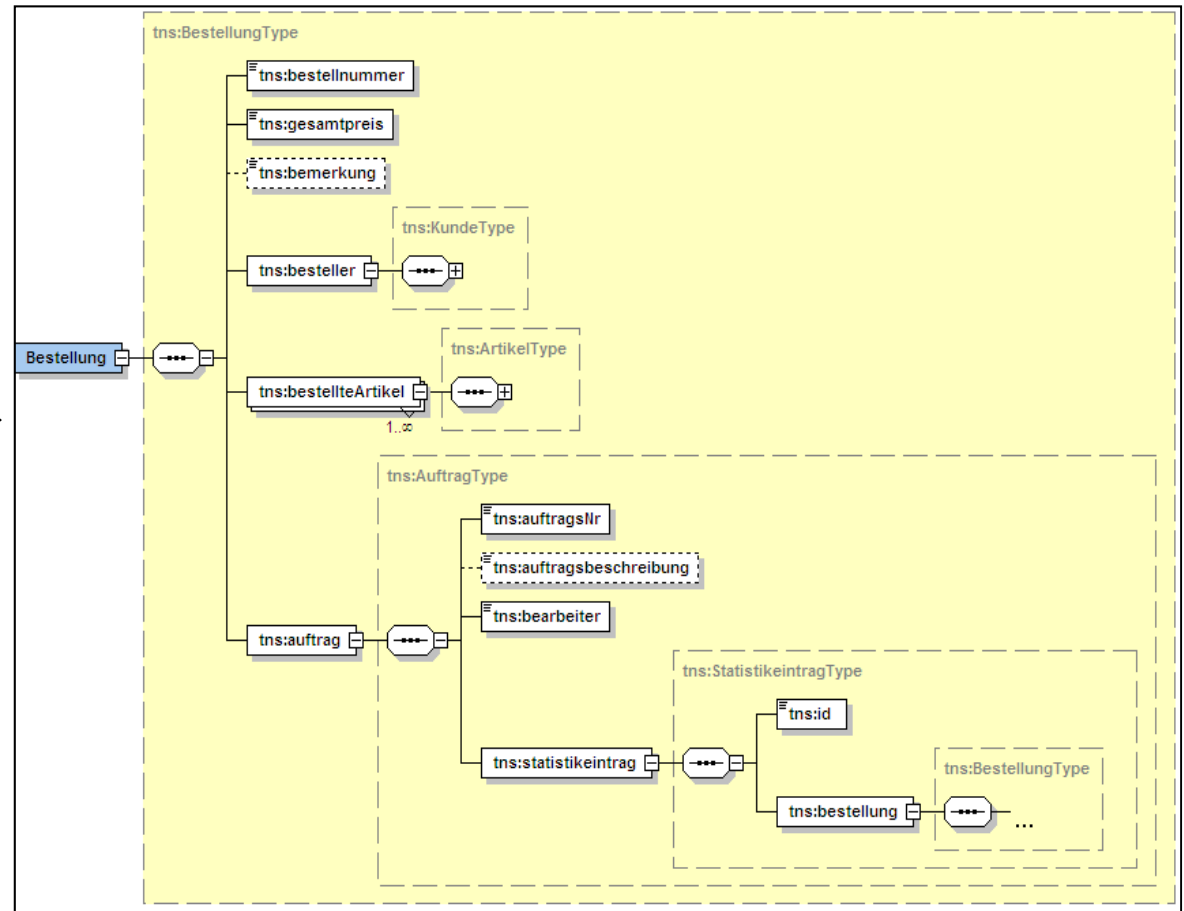
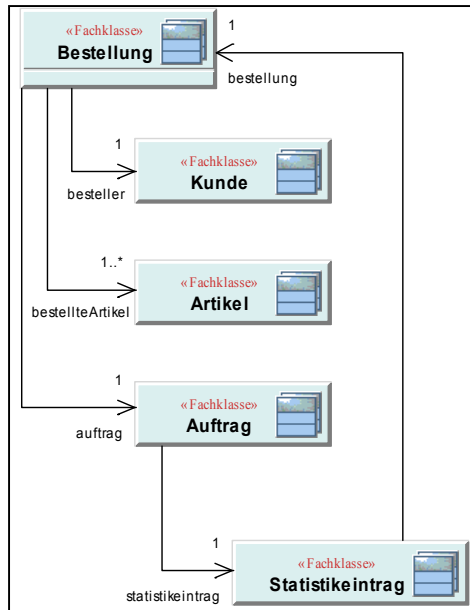
Die Generierung wird mit geeigneter Selektion als Engineering-Aktion aufgerufen.

## Ausführung der Generierung



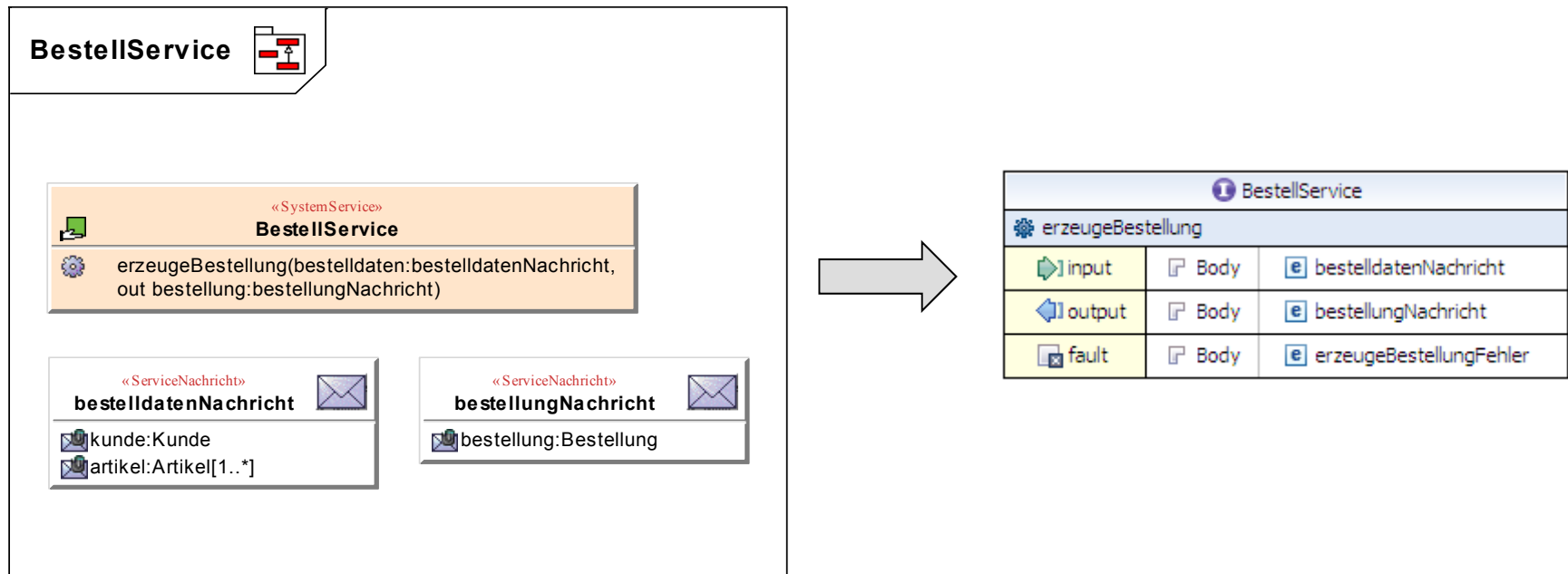
Die Klassen des Modells werden als ComplexTypes in der XSD-Datei generiert.

## Beispiel für generierte XSD-Dateien



Die Informationen der modellierten Services werden in der WSDL-Datei generiert.

## Beispiel für generierte WSDL-Dateien



## Die Generierung erfolgt in mehreren Teilschritten.

### Ablauf der WSDL/XSD - Generierung

1. Start der Engineering-Aktion in Innovator
2. Aufruf der Root-Schablone durch den oAW-Workflow
3. Zusammenstellen der zu generierenden Klassen
4. Zusammenstellen der zu generierenden Datentypen
5. Generierung der XSD-Datei
6. Generierung der WSDL-Datei
7. Aufruf des Beautifiers



Ansprechpartner: René Beckert



**Hauptverwaltung Nürnberg**

**MID GmbH**

Eibacher Hauptstraße 141  
90451 Nürnberg  
Telefon: +49 (0)911 - 968 36-0  
Telefax: +49 (0)911 - 968 36-10

**Niederlassung Stuttgart**

**MID GmbH**

Silberburgstraße 187  
70178 Stuttgart  
Telefon: +49 (0)711 - 633 859-0  
Telefax: +49 (0)711 - 633 859-10

[www.mid.de](http://www.mid.de)