

XSD Visualizer Plugin: Simplify XSD Understanding (even for Beginners)

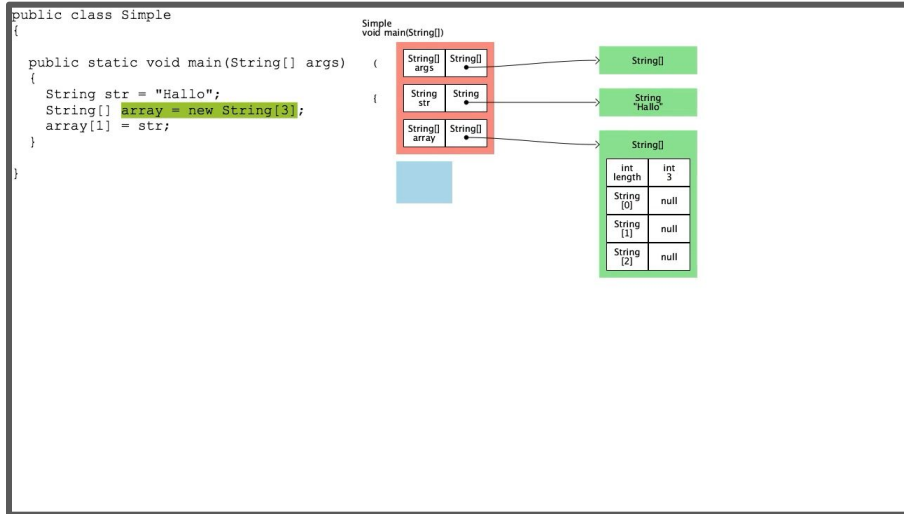
Declarative Amsterdam 2023

Introduction



- Sven Reinck
- From Hamburg, Germany
- Graduated as Master of Computer Science in 2007
- Freelance IT-Trainer since 2012
 - Java, Kotlin, Groovy, C++, OpenGL, JavaFX, GoLang

Javis (Java Visualizer)



- Developed since 2014
- Founded FLUXparticle in 2019
- Developed XSD Visualizer plugin since 2021
- Focus on visualization
- Not a daily user of XML

Data vs Document

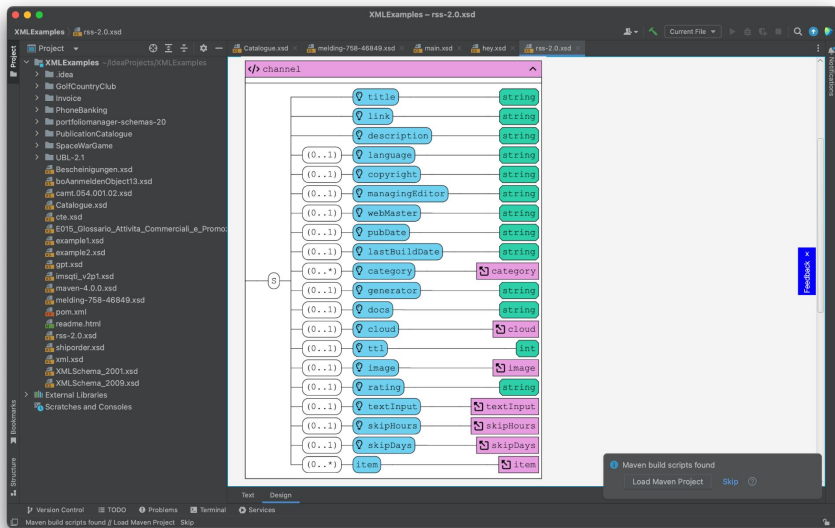
```
<XMLSchemaDependencies>
  <Declaration name="identifiedType" type="TYPE"/>
  <Declaration name="beans" type="ELEMENT">
    <Dependency ref="description" type="USES"/>
    <Dependency ref="import" type="USES"/>
    <Dependency ref="alias" type="USES"/>
    <Dependency ref="bean" type="USES"/>
    <Dependency ref="beans" type="USES"/>
  </Declaration>
  <Declaration name="description" type="ELEMENT"/>
  <Declaration name="import" type="ELEMENT"/>
  <Declaration name="alias" type="ELEMENT"/>
  <Declaration name="beanElements" type="GROUP">
    <Dependency ref="description" type="USES"/>
    <Dependency ref="meta" type="USES"/>
    <Dependency ref="constructor-arg" type="USES"/>
    <Dependency ref="property" type="USES"/>
    <Dependency ref="qualifier" type="USES"/>
    <Dependency ref="lookup-method" type="USES"/>
    <Dependency ref="replaced-method" type="USES"/>
  </Declaration>
</XMLSchemaDependencies>
```

```
<chapter>
  <title>My title goes here</ title>

  <para>Paragraph text goes here.</ para>

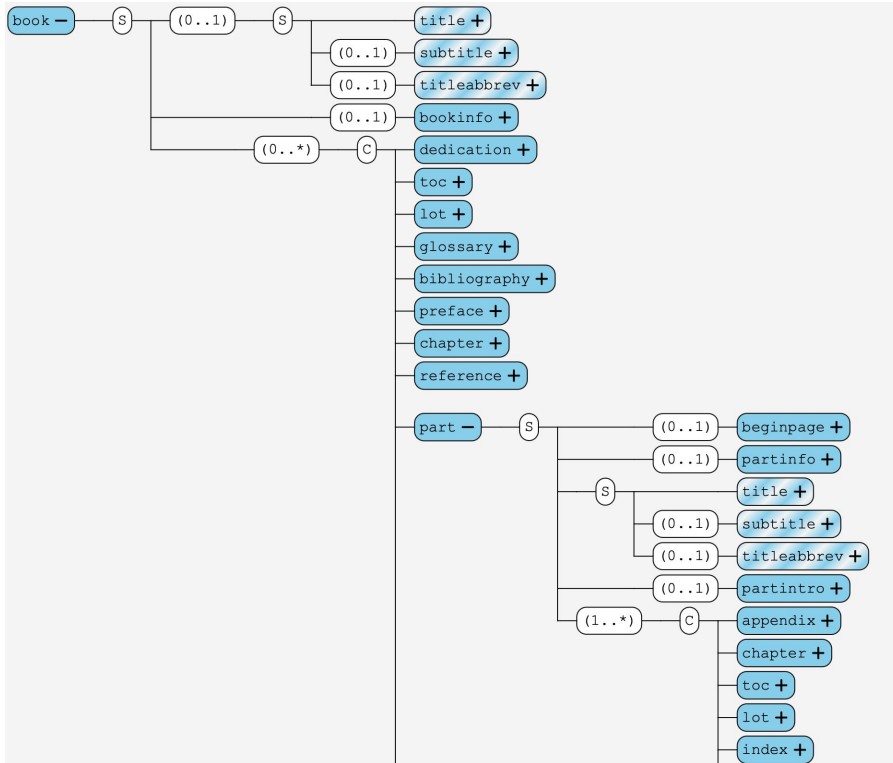
  <section>
    <title>A section title</ title>
    <para>
      More paragraph text.
      Some in < emphasis>italics</ emphasis>.
    </ para>
  </ section>
</ chapter>
```

XSD Visualizer Plugin



- Directly where you need it
- Visualization is tightly packed
- Annotations as a tooltip (Light bulb)

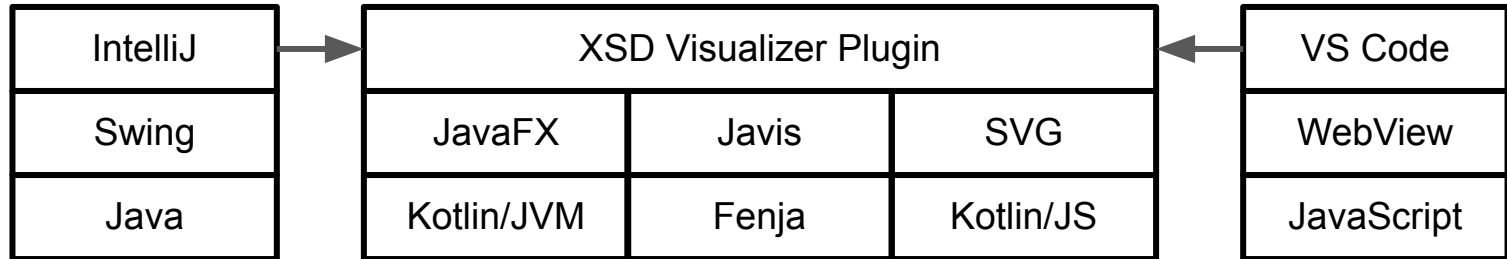
Future Developments: Tree View



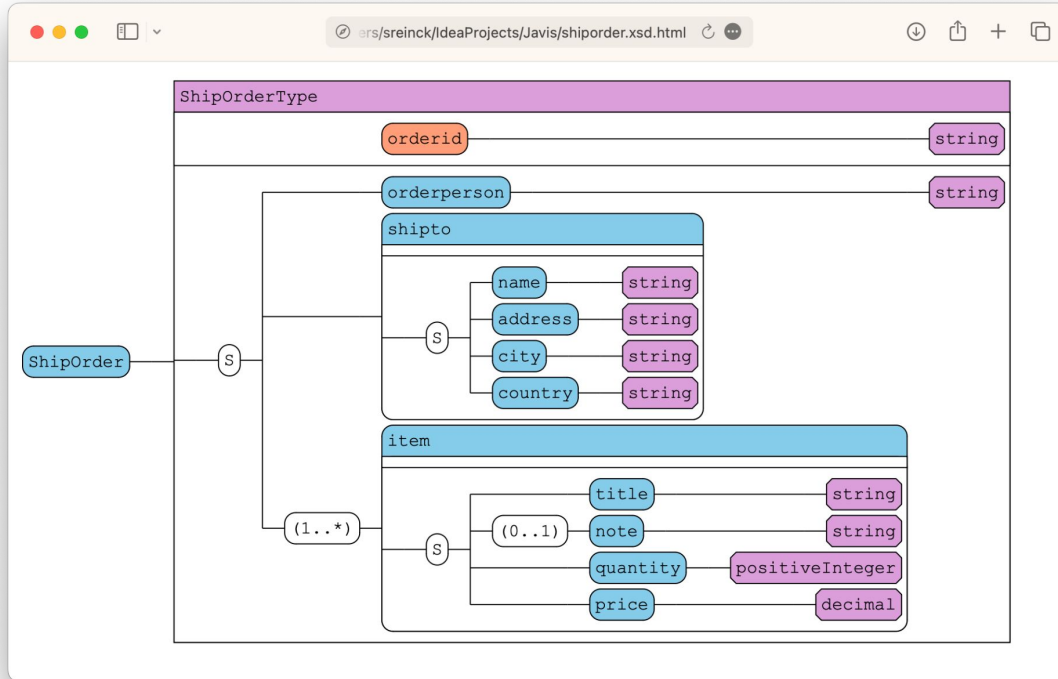
- Similar to Near&Far

Demo

Technical Details



HTML/SVG Export



Other Schema Formats



- Schematron?
- Relax NG
- JSON Schema

Schema Design in AR/VR?



IntelliJ: "XSD / WSDL Visualizer"

The screenshot displays the IntelliJ IDEA interface with the XSD/WSDL Visualizer open for the file `rss-2.0.xsd`. The visualizer shows a tree structure of the XML Schema Definition (XSD) elements. The root element is `channel`, which is highlighted in pink. Below it, various elements are listed, each with its cardinality and data type. The elements and their types are:

- `title` (string)
- `link` (string)
- `description` (string)
- `language` (string)
- `copyright` (string)
- `managingEditor` (string)
- `webMaster` (string)
- `pubDate` (string)
- `lastBuildDate` (string)
- `category` (string)
- `generator` (string)
- `docs` (string)
- `cloud` (cloud)
- `tts` (int)
- `image` (image)
- `rating` (string)
- `textInput` (textInput)
- `skipHours` (skipHours)
- `skipDays` (skipDays)
- `item` (item)

The visualizer also shows the cardinality of each element, such as `(0..1)` for most elements and `(0..*)` for `category` and `item`. The `category` and `item` elements are highlighted in pink, indicating they are complex types. A notification at the bottom right of the IDE states: "Maven build scripts found" with buttons for "Load Maven Project" and "Skip".

Visual Studio Code: "SchemaViz"

The image displays the Visual Studio Code interface with an XML Schema (XSD) file named `shiporder.xsd` open. The left pane shows the raw XML code, and the right pane shows a graphical visualization of the schema structure.

```
1 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
2
3   <xs:element name="shiporder">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element name="orderperson" type="xs:string"/>
7         <xs:element name="shipto">
8           <xs:complexType>
9             <xs:sequence>
10              <xs:element name="name" type="xs:string"/>
11              <xs:element name="address" type="xs:string"/>
12              <xs:element name="city" type="xs:string"/>
13              <xs:element name="country" type="xs:string"/>
14            </xs:sequence>
15          </xs:complexType>
16        </xs:element>
17        <xs:element name="item" maxOccurs="unbounded">
18          <xs:complexType>
19            <xs:sequence>
20              <xs:element name="title" type="xs:string"/>
21              <xs:element name="note" type="xs:string" minOccurs="0"/>
22              <xs:element name="quantity" type="xs:positiveInteger"/>
23              <xs:element name="price" type="xs:decimal"/>
24            </xs:sequence>
25          </xs:complexType>
26        </xs:element>
27      </xs:sequence>
28    <xs:attribute name="orderid" type="xs:string" use="required"/>
29  </xs:complexType>
30 </xs:element>
31 </xs:schema>
```

The visualization on the right shows the hierarchical structure of the schema. The root element is `shiporder`, which contains an `orderid` attribute (string) and a sequence of elements. The sequence includes an `orderperson` element (string) and a `shipto` complex type. The `shipto` type contains a sequence of elements: `name` (string), `address` (string), `city` (string), and `country` (string). Additionally, the root sequence contains an `item` complex type that occurs unboundedly (indicated by `(1..*)`). The `item` type contains a sequence of elements: `title` (string), `note` (string, optional, indicated by `(0..1)`), `quantity` (positiveInteger), and `price` (decimal).

Thanks for listening!



- Please fill out the Survey:
<https://www.fluxparticle.com/xsdvisualizer/>
(QR Code)
- E-Mail: sreinck@fluxparticle.com
- Slack: XML.com (Sven Reinck)
- Twitter: @FLUXparticleCOM