X-definition 4.0

Václav Trojan

trojan@syntea.cz

JSON in X-definition

Let's have the JSON data:

```
{
    "Book" : {
       "Title" : "The Hitchhiker's guide to the Galaxy",
       "Price" : 12.3
    }
}
```

How the X-definition will look?

```
<xd:def xmlns:xd="http://www.xdef.org/xdef/4.0" name="Example" root="Book" >
 <xd:json name="Book" >
 {
    "Book" : {
       "Title" : "string()",
       "Price" : "float(0, 1000)"
    }
 }
 </xd:json>
</xd:def>
```

1. Occurrence of JSON values and objects

More complex JSON data:

```
{ "Books" :
   [
      {
          "Book" : { "Title" : "The Hitchhiker's guide to the Galaxy", "Price" : 12.3 }
      },
      {
          "Book" : { "Title" : "Introduction to X-definition"}
      }
   ]
}
```

Specification of occurrence of objects and values:

```
<xd:def xmlns:xd="http://www.xdef.org/xdef/4.0" name="Books" root="BookList" >
 <xd:json name="BookList" >
  {"Books" :
    [
       {"Book" : {$script : "occurs *",
                  "Title" : "required string()",
                  "Price" : "optional float(0, 1000)"
                 }
       }
    ]
  }
 </xd:json>
</xd:def>
```

2. Description of matrix in JSON

JSON data representing matrix 3 x 3 of integers:

```
[
  [  1,  2, 3 ]
  [ -1,  0, 111 ]
  [ 99, -5, 17 ]
]
```

Specification of occurrence of objects and values (note that also comments can be added into the JSON description):

```
<xd:def xmlns:xd="http://www.xdef.org/xdef/4.0" name="ExampleMatrix" root="Matrix" >
 <xd:json name="Matrix" >
    /* Description: Matrix 3 x 3 of integers */
    [ $script: "occurs 3",
       /* lines with integers */
       [ "occurs 3 int()" ]
    ]
 </xd:json>
</xd:def>
```

3. $oneOf specification in JSON

Let's have following JSON data:

```
[
  { "Genre": ["classic"] },
  { "Genre": ["Rock", "pop" ] },
  { "Genre": "Country" },
  { "Genre": [] }
]
```

Value of "Genre" can be either a string or an array of strings or the empty array:

```
<xd:def xmlns:xd="http://www.xdef.org/xdef/4.0" name="Example" root="Matrix" >
 <xd:json name="Matrix" >
  [
    {
      "Genre": [$oneOf: "occurs 1..10",
        "string()",
         ["occurs * string()"]
      ]
    }
  ]
 </xd:json>
</xd:def>
```

4. Example of the Java program with JSON data

```
File xdefBookList;
File jsonData;
Properties props = new Properties();
XDPool xpool; XDDocument xdoc;
Object json;
 ...

xpool = XDFacctory.compileXD(props, xdefBookList);
xdoc = xpool.createXDDocument("Books");
ArrayReporter reporter = new ArrayReporter();
json = xdoc.jparse(jsonData, "BookList", reporter);
if (reporter.errors()) {
  System.out.println(reporter);
} else {
    ...
}
```

Lexicon: language localization of XML

Let's have following XML document in the English:

```
<Contract ID = "12345">
  <Date>
     2019-05-12
  </Date>
  <Client Name = "John Smith"
     PersonalID = "0987654321"
  />
</Contract>
```

And the same XML structure e.g. in the Russian:

```
<контракт ид = "12345">
  <Дата>
     2019-05-12
  </Дата >
  <клиент имя = "John Smith"
     Персонномер = "098765432"
  />
</контракт>
```

1. Lexicon description in X-definition

Let's have following X-definition:

```
<xd:def xmlns:xd="http://www.xdef.org/xdef/4.0" name="Example" root="Contract" >
 <Contract ID = "int()">
   <Date>
     date();
   </Date>
   <Client Name       = “optional string()"
           PersonalID = “optional num()"/>
 </Contract>
</xd:def >
```

Lexicons for XML model in the X-definition above in the English and in the Russian:

```
<xd:lexicon xmlns:xd="http://www.xdef.org/xdef/4.0" language="eng" default="yes" />

<xd:lexicon xmlns:xd="http://www.xdef.org/xdef/4.0" language="rus" >
  Example#Contract                  =  контракт
  Example#Contract/@ID              =  ид
  Example#Contract/Date             =  Дата
  Example#Contract/Client           =  клиент
  Example#Contract/Client/@Name     =  имя
  Example#Contract/Client/@PersonalID =  Персонномер
</xd:lexicon>
```

2. Example of the Java program with lexicon

```
File xdefExample, lexiconEng, lexiconRus;
File dataEng, dataRus;
Properties prop = new Properties();
XDPool xpool; XDDocument xdoc;
Element elem;
...
xpool = XDFacctory.compileXD(prop, xdefExample, lexiconEng, lexiconRus);
xdoc = xpool.createXDDocument("Example");
xdoc.setLexiconLanguage("eng");
ArrayReporter reporter = new ArrayReporter();
elem = xdoc.xparse(dataEng, reporter);
if (reporter.errors()) System.out.println(reporter);
else
...
xdoc.setLexiconLanguage("rus");
elem = xdoc.xparse(dataRus, reporter);
if (reporter.errors()) System.out.println(reporter);
...
```

Where to find it :

- github: github.com/syntea/xdef
- maven: search.maven.org/search?q=g:org.xdef
- Syntea web: www.xdefinice.cz/en/download-sekce
  http://xdef.syntea.cz/tutorial

Transformation between X-definition and XML Schema format

Tomáš Šmíd

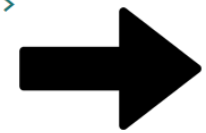
Introduction

- Complexity of XML Schema vs X-definition
- Extended X-definition validation over XML schema
- Bidirectional transformation

Transformation of XML schema

- **Lossless** transformation
- Directly 1:1
  - Some nodes are redundant

```xml
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
attributeFormDefault="unqualified"
elementFormDefault="unqualified">
    <xs:complexType name="c">
        <xs:attribute name="d" type="xs:int" use="required"/>
    </xs:complexType>
    <xs:element name="a">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="b" type="c"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```



```xml
<xd:def xd:name="schemaNode" xd:root="a"
    xmlns:xd="http://www.xdef.org/xdef/3.2">
    <c d="int"></c>
    <a>
        <b xd:script = "ref c;"></b>
    </a>
</xd:def>
```

Transformation of X-definition

- **Lossy** transformation
- Required preprocessing and postprocessing
- Many of nodes can be transformed directly
- Reference and type nodes — Lossless transformation, but sometimes not direct
- Issue with target namespace
- Many features of X-definition are unsupported
  - No equivalency exists in XML schema

Thank you

For your attention