universität wien

# Data Validation

Faculty of Computer Science
Workflow Systems and Technologies

## Interoperability SS 2023

Amolkirat Singh Mangat
Matthias Ehrendorfer
Florian Stertz

# Recap from Introduction

Interoperability is *“ … the ability of two or more systems or components to* ***exchange information*** *and to* ***use the information*** *that has been exchanged.”* (IEEE Standard Computer Dictionary)

*“ … interoperability means that two (or more)* ***systems work together unchanged*** *even though they weren't necessarily designed to work together …”* (B. Woolf, IBM)

# Introduction

*What is a data schema?*

*Why do we need data schemas?*

# Data Schema

- Defines the structure of the data/information it describes
- Example: database schema in relational databases (RDBMS)
  - allows to check if data conforms to the schema (e.g., when inserting it into a database)
  - allows to query the data with SQL
  - allows to define rules for transformation when transferring data to a new database/data storage
- This would also be desirable for data which is not structured in the first place

# Data Schema

- Data schema is not mandatory for semi-structured data but enables easier:
  - checking of exchanged data
  - querying of data
  - transformation of exchanged data
- Documents (e.g., XML) can be
  - well-formed: adheres to XML standard
  - valid: adheres to DTD (or other schema)
- Different data schemas for semi-structured data (e.g., XML, JSON):
  - XML Schemas: [1]

        Document Type Definition (DTD)
        XML Schema Definitions (XSD)
        RelaxNG
  - JSON Schema

# Document Type Definition (DTD)

- Written in Standard Generalised Markup Language (SGML)
- Specified internal or external (inside or outside the XML document)

**XML document**

```
<beings>
  <being name="Batman" level="1000">
    <trait name="stealth" score="4"/>
    <trait name="charisma" score="1"/>
    <trait name="ruthlessness" score="5"/>
  </being>
  <being name="Storm" level="9900">
    <trait name="stealth" score="3"/>
    <trait name="charisma" score="4"/>
    <trait name="ruthlessness" score="4"/>
  </being>
</beings>
```

**Schema**

```
<!ELEMENT beings (being*)>
<!ELEMENT being (trait*)>
<!ELEMENT trait (#PCDATA)>

<!ATTLIST trait name CDATA #REQUIRED>
<!ATTLIST trait score CDATA #REQUIRED>
<!ATTLIST being name CDATA #REQUIRED>
<!ATTLIST being level CDATA #REQUIRED>
```

see also https://www.w3.org/TR/WD-html40-970708/intro/sgmltut.html#howtodtd

# XML Schema Definitions (XSD)

- Is itself represented in an XML
- Extends capabilities of XML document type definitions (DTDs):
  - supports data types
  - supports namespaces
  - extensible

see also https://www.w3.org/TR/xmlschema-0/

# XML Schema Definitions (XSD)

### XML document

```
<beings>
  <being name="Batman" level="1000">
    <trait name="stealth" score="4"/>
    <trait name="charisma" score="1"/>
    <trait name="ruthlessness" score="5"/>
  </being>
  <being name="Storm" level="9900">
    <trait name="stealth" score="3"/>
    <trait name="charisma" score="4"/>
    <trait name="ruthlessness" score="4"/>
  </being>
</beings>
```

### Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:complexType name="trait">
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="score" type="xsd:positiveInteger"/>
</xsd:complexType>

<xsd:complexType name="being">
  <xsd:sequence>
    <xsd:element name="trait" type="trait" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="level" type="xsd:positiveInteger"/>
</xsd:complexType>

<xsd:complexType name="beings">
  <xsd:sequence>
    <xsd:element name="being" type="being" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="beings" type="beings"/>

</xsd:schema>
```

# RelaxNG

- Has both XML syntax and non-XML syntax
- Allows use of other datatyping languages (e.g., XSD)
- Extends capabilities of XML document type definitions (DTDs):
  - supports data types
  - supports namespaces
  - extensible

see also https://relaxng.org/ and
https://relaxng.org/spec-20011203.html

# RelaxNG

### XML document

```
<beings>
  <being name="Batman" level="1000">
    <trait name="stealth" score="4"/>
    <trait name="charisma" score="1"/>
    <trait name="ruthlessness" score="5"/>
  </being>
  <being name="Storm" level="9900">
    <trait name="stealth" score="3"/>
    <trait name="charisma" score="4"/>
    <trait name="ruthlessness" score="4"/>
  </being>
</beings>
```

### Schema

```
<grammar xmlns="http://relaxng.org/ns/structure/1.0">

<define name="trait">
  <element name= "trait">
    <attribute name="name"><text/></attribute>
    <attribute name="score"><data type="positiveInteger"
datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes"/></attribute>
  </element>
</define>

<define name="being">
  <element name= "being">
    <zeroOrMore><ref name="trait"/></zeroOrMore>
    <attribute name="name"><text/></attribute>
    <attribute name="level"><data type="positiveInteger"
datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes"/></attribute>
  </element>
</define>

<start>
  <element name= "beings">
    <zeroOrMore><ref name="being"/></zeroOrMore>
  </element>
</start>

</grammar>
```

# JSON Schema

- Allows to annotate and validate JSON documents
- Is itself a JSON document

**JSON document**

```
{"beings":
  [{"@name":"Batman", "@level":1000, "traits":
     [{"@name":"stealth", "@score":4},
       {"@name":"charisma", "@score":1},
       {"@name":"ruthlessness", "@score":5}]
    },
    {"@name":"Storm", "@level":9900, "traits":
     [{"@name":"stealth", "@score":3},
       {@name":"charisma", @score":4},
       {@name":"ruthlessness", "@score":4}]
    }]
}
```

**Schema**

```
{$schema": "https://json-schema.org/draft/2019-09/schema",
  "type": "object",
  "properties": {
    "beings": {
      "type": "array", "minItems":0, "maxItems":5,
      "items": {
        "type": "object",
        "properties": {
          "@name": {"type":"string"},
          "@level": {"type":"integer"},
          "traits": {"type":"array"}
      },
      "required": ["@name","@level","traits"]
      }
    }
  },
  "required": ["beings"]
}
```

see also https://json-schema.org/

# References

[1] https://www.w3.org/standards/xml/schema, W3C, last accessed: 23.03.2023

- additional resources mentioned on the slides:
    - https://www.w3.org/TR/WD-html40-970708/intro/sgmltut.html#howtodtd, last accessed: 23.03.2023
    - https://www.w3.org/TR/xmlschema-0/, last accessed: 23.03.2023
    - https://relaxng.org/, last accessed: 23.03.2023
    - https://relaxng.org/spec-20011203.html, last accessed: 23.03.2023