Koordinierungsstelle
für IT-Standards

Freie
Hansestadt
Bremen

XML-MUTATE

A declarative approach to XML Mutation and Test Management

Dr. Renzo Kottmann

Prague 2020-02-15

Acknowledgement

- Coordination Office for IT-Standards
  - Fabian Büttner
  - Frank Steimke
  - Lutz Rabe
  - Anna Dopatka
- init AG
  - Andreas Penski
  - Victor del Campo

About me

- Dipl. Ing. Informatics and MSc Marine Microbiology
- PhD Bioinformatics
- My main topics:
  - Data Integration
  - Databases
  - Data Standardization
  - Data Retrieval
- I love coding and <structured-data/>

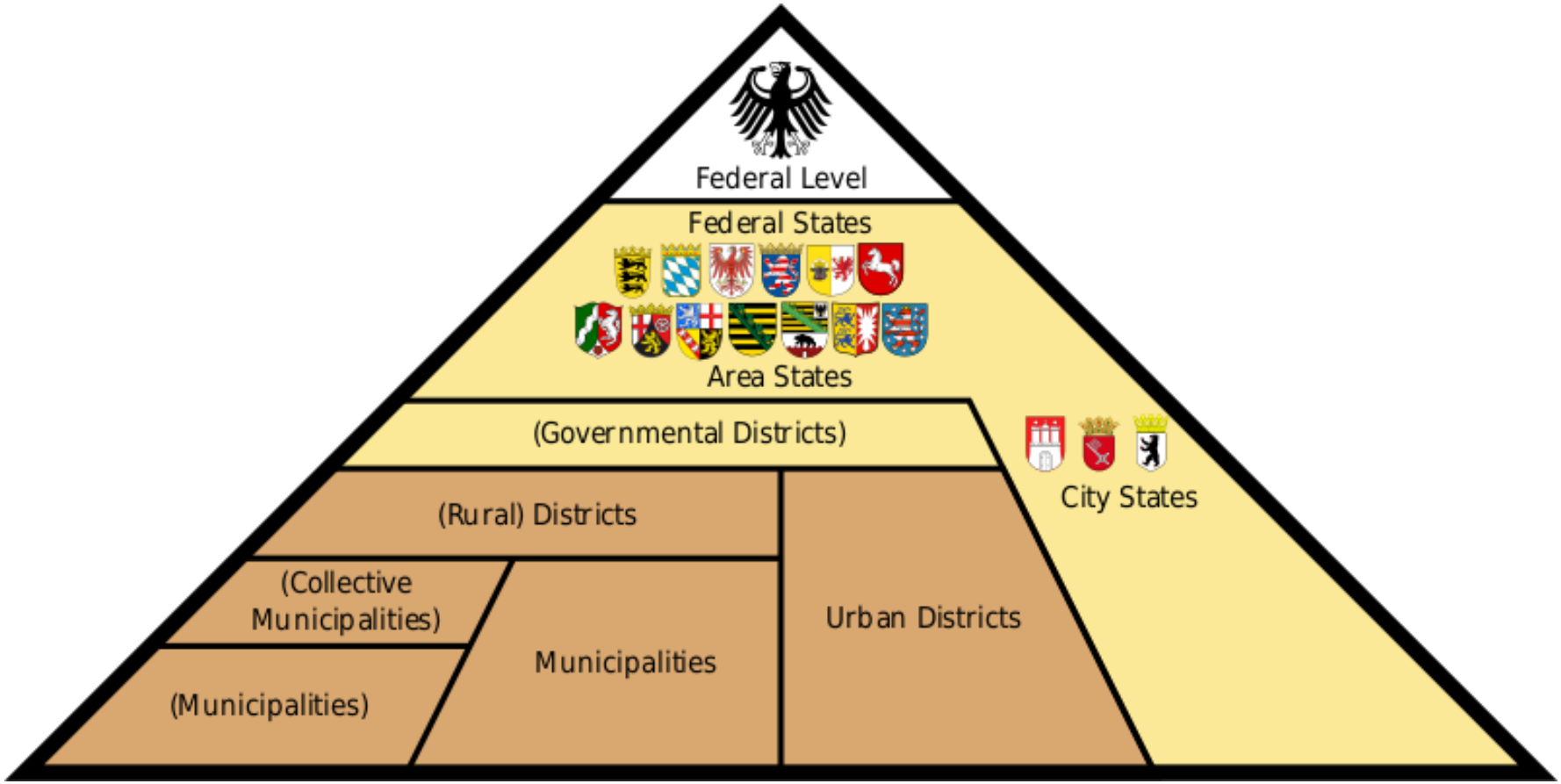
About this presentation

Koordinierungsstelle für IT-Standards

```
<!-- @renzokott What the heck is this? -->
<data-driven>
  <?xmute mutator="code" values="13-99"
          schema-valid schematron-invalid="r1"
          description="True Negative"
          id="t1"
          tags="mandatory simple"?>
  <monthday>02-15</monthday>
</data-driven>
<!– @renzokott Why did you do this? -->
<!– @xmlprague what do you think dear
#xmlprague attendees? -->
```

Our Ecosystem: Distributet System

Koordinierungsstelle für IT-Standards



Source: https://en.wikipedia.org/wiki/File:Administrative_divisions_of_Germany.svg

Coordination Office for IT-Standards

Koordinierungsstelle für IT-Standards

- Develops, operates and maintains standards for the public sector
- Focused on interoperability and data exchange
- Publishes and maintains standardization methodology XÖV (XML for Public Entities)
  - > 20 standards based on XÖV
- Contributes to national and European standardization activities
  - Member of DIN and CEN
- Developes and maintaince XRechnung
  - XML based invoices for Business to Government
- Hosts experts groups and steering committees
- Hosts XOEV conference

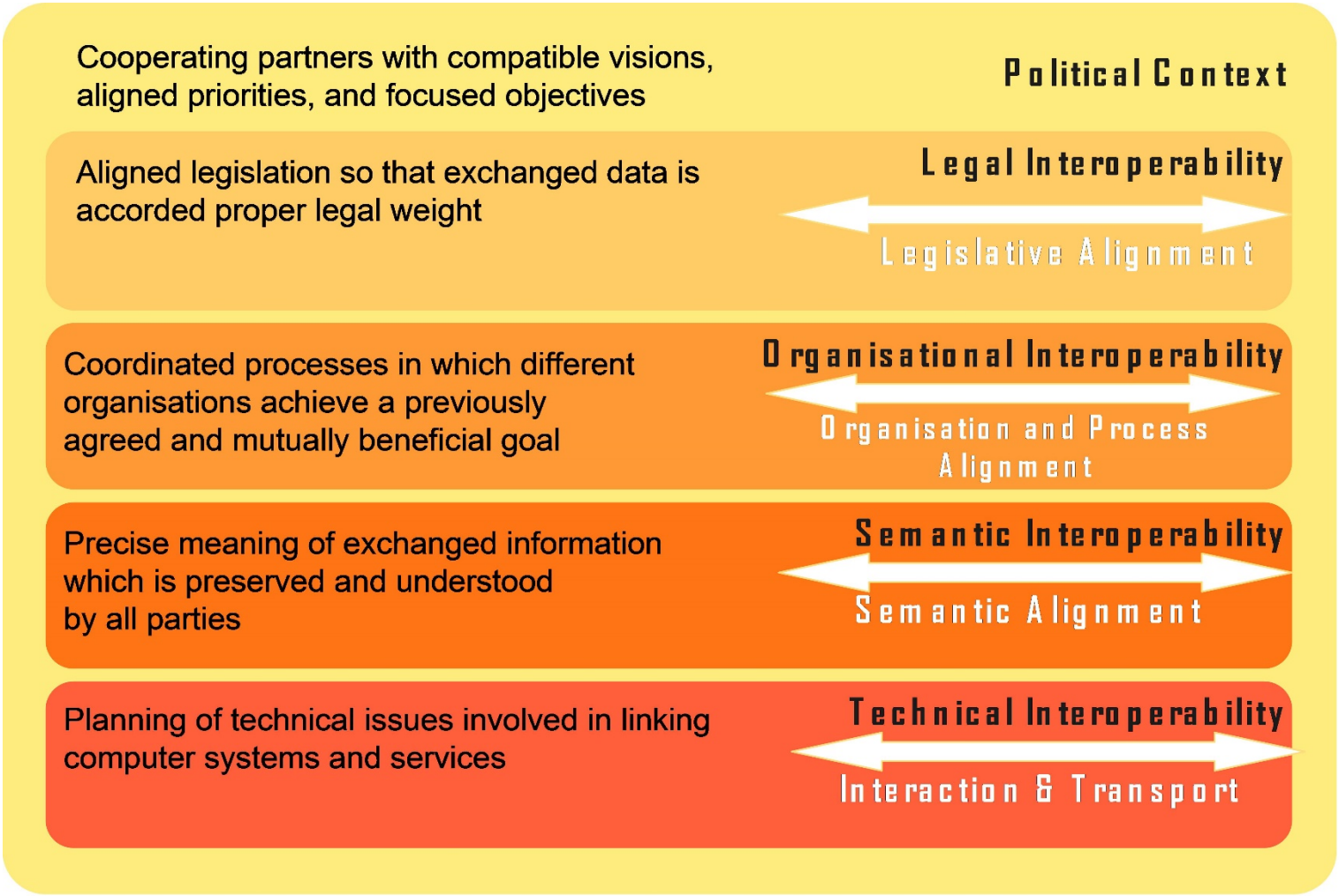
Coordination Office for IT-Standards

Koordinierungsstelle für IT-Standards

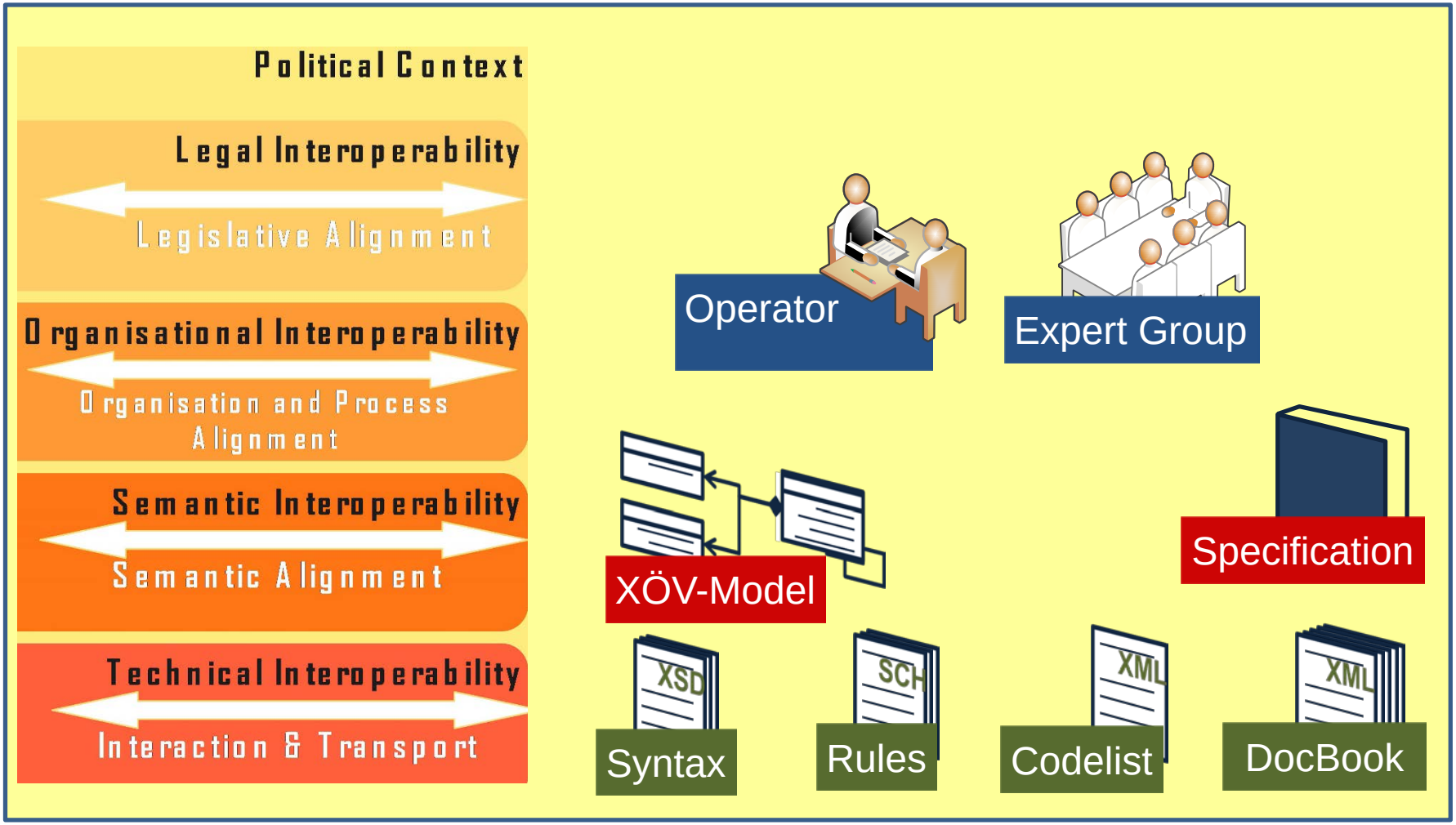
- Examples of current mandates and tasks:
  - For Federal Government of Germany
    - Operation of XRechnung
  - For federal state of North Rhine-Westphalia
    - Development of XGewerbe
  - For city state of Free Hanseatic City of Bremen
    - Development XFamilie
- Other activities
  - Innovation, methodology development, project acquisition and management
- 17 colleagues
- 3.5 m EUR Budget

Working on all levels of interoperability

Koordinierungsstelle für IT-Standards



Components and activities



XÖV „Standard for Standard Development“

Koordinierungsstelle für IT-Standards

- Includes well established
  - Methods
    - Test methodology
  - Rules
  - Guidelines
  - Tools and infrastructure components
- Completely XML based

XRechnung – A B2G Standard

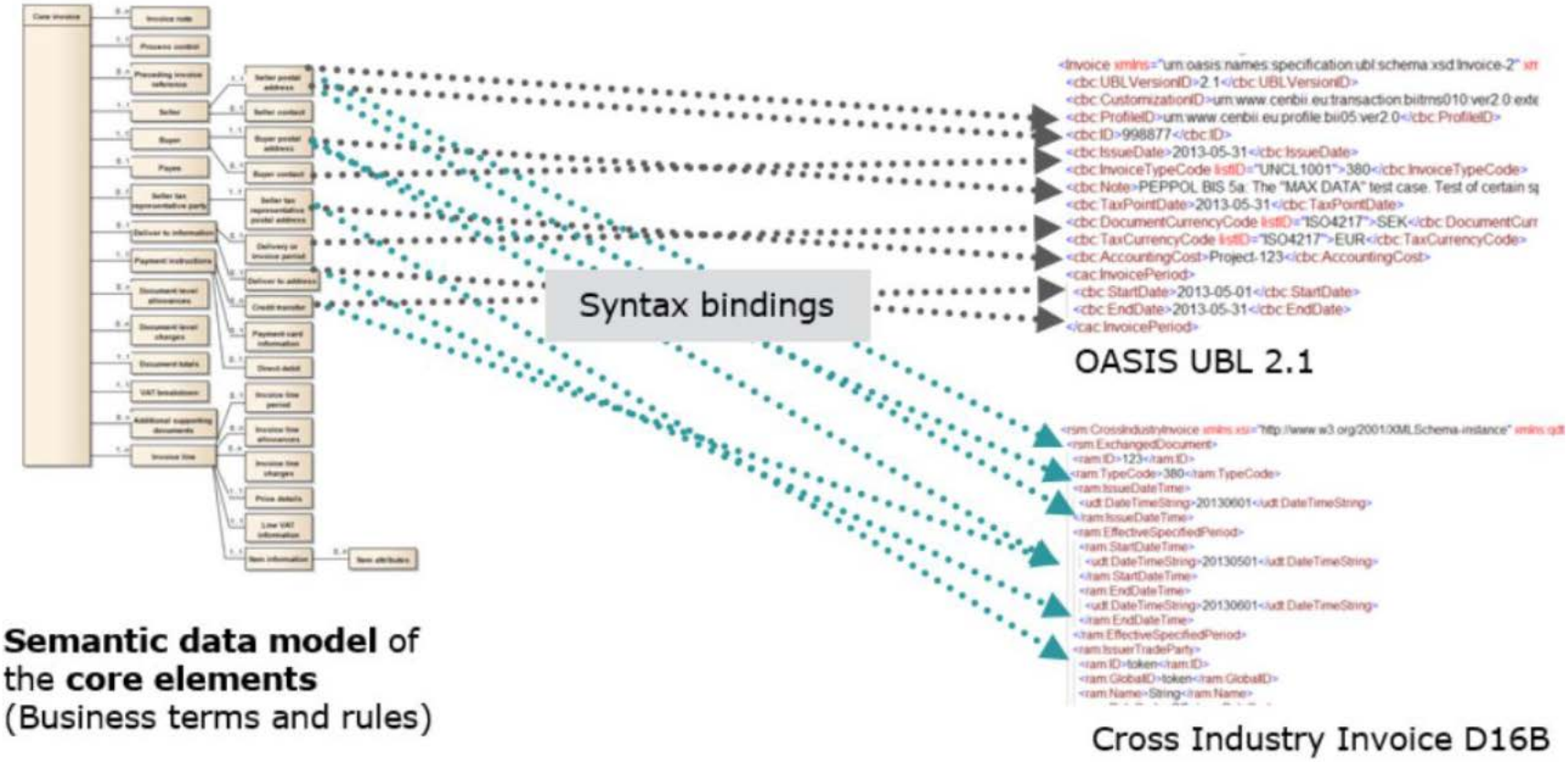
Koordinierungsstelle für IT-Standards

- By end of this year all entities of public administration will accept:

- Invoices
  - Which is a concrete usage of the European EN16931 norm
  - With special rules to comply to German laws and regulations

European Norm

Koordinierungsstelle
für IT-Standards



Semantic data model of
the **core elements**
(Business terms and rules)

Complex Validation Scenarios

Koordinierungsstelle für IT-Standards

- OASIS UBL 2.1
  1. XML Schema
     1. Invoices
     2. CreditNote
  2. CEN Schematron
  3. XRechnung Schematron

- Cross Industry Invoice
  1. XML Schema
  2. CEN Schematron
  3. Xrechnung Schematron

Note on Agile Standards

Koordinierungsstelle für IT-Standards

- We are open
  - Everyone can comment on the specification at any time and we start working on it
  - https://projekte.kosit.org/xrechnung/xrechnung
- Lean organization
  - Keeping process clear and simple
- Automatize and simplify publishing process
  - We are moving in this direction, but it needs TESTS, TESTING, TESTS and did I say TESTS?

Quality Control (kind of mission critical)

Koordinierungsstelle für IT-Standards

- A lot can go wrong
- What can go wrong goes wrong
- Complex overlapping rules
  - > thousand rules
  - Have to have consistent behavior across different syntaxes

Test Management a year ago

Koordinierungsstelle für IT-Standards

- We had 16 example invoices with real world data
  - Based on a custom test framework
  - They get validated with current XML Schemas and Schematrons
- Manual checking
  - Very good
  - But costs time, time and a lot of communication

Koordinierungsstelle
für IT-Standards

Freie
Hansestadt
Bremen

In need of better Testmanagement

Aim

Koordinierungsstelle für IT-Standards

- Have negative test cases
- Automatically generate many test cases
- Many test cases from a singe XML instance
- Validate each XML instance against
  - Many XML Schemas and
  - Many Schematrons
  - at the same time
- Fully automated testing

Data-driven testing

A new integrated approach

- Some tools exist covering specific aspects
- Need a combination of
  1. Generation by Mutation
  2. Evaluation of validation by Expectation
  3. Declarative Annotation

Mutation

Running Example

Koordinierungsstelle für IT-Standards

- Business rule:
  - It is mandatory to record month and day of an event

Running Example

Koordinierungsstelle für IT-Standards

- Business rule:
  - It is mandatory to record month and day of an event

- Specification by example:

```
<?xml version="1.0"?>
<data-driven>
  <monthday>02-15</monthday>
</data-driven>
```

Generate Test Cases

Koordinierungsstelle für IT-Standards

- Original (valid) instance:

```
<data-driven>
  <?xmute mutator="remove" ?>
  <monthday>02-15</monthday>
</data-driven>
```

- Mutation generates new XML instance:

```
<data-driven>
  <?xmute mutator="remove" ?>
</data-driven>
```

Koordinierungsstelle
für IT-Standards

Possible Mutators

Name	# Mutants	Description
empty	1	Deletes the text content of an element or attribute.
add	1	Adds an element or attribute.
remove	1	Removes an element or attribute.
rename	1	Renames an element or attribute.
change-text	m	Changes text content of an element or attribute.
whitespace	m	Exchanges text content of an element or attribute with random whitespace content.
identity	1	Keeps element as is.
code	m	Exchanges text content of an element with a list of code words one by one.
alternative	m	Uncomments each comment one by one. Allows to e.g. test XML Schema choices [xsd-choice].
random-element-order	m	Randomizes child element order of an element.

Evaluation of validation

by Expectation

Evaluation of XML Schema validation

Koordinierungsstelle für IT-Standards

- Declare **Expectation** about validation outcome

```
<data-driven>
  <?xmute mutator="remove" schema-valid?>
  <monthday>02-15</monthday>
</data-driven>
```

Given XML Schema

Koordinierungsstelle für IT-Standards

```
<xs:schema  …>
  <xs:element name="data-driven">
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element name="monthday">
          <xs:simpleType>
            <xs:restriction
base="xs:normalizedString“ />
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

- Outcome of:

```
<data-driven>
  <?xmute mutator="remove"
              schema-valid ?>
  <monthday>02-15</monthday>
</data-driven>
```

Running Example: But it is mandatory!

Koordinierungsstelle für IT-Standards

- Business rule:
  - It is mandatory to record month and day of an event

- Specification by (two) examples:

```
<data-driven>
  <?xmute mutator="remove" schema-invalid ?>
  <monthday>02-15</monthday>
</data-driven>
```

Given XML Schema

Koordinierungsstelle für IT-Standards

```
<xs:schema  …>
  <xs:element name="data-driven">
    <xs:complexType>
      <xs:sequence minOccurs="1">
        <xs:element name="monthday">
          <xs:simpleType>
            <xs:restriction
base="xs:normalizedString" />
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

- Outcome of:

```
<data-driven>
  <?xmute mutator="remove"
              schema-invalid ?>
  <monthday>02-15</monthday>
</data-driven>
```

Real Power of data-driven expectations

- We can express all combinations of what we expect:
  - A validation has to
    - accept valid content as **True Positive (TP)**
    - exclude invalid content as **True Negative (TN)**
  - and **does not** have to
    - accept invalid content as **False Positive (FP)**,
    - exclude valid content **False Negative (FN)**

Validation Result vs. Expectation	Valid	invalid
Valid	+ (TP)	- (FP)
Invalid	- (FN)	+ (TN)

Running Example: What's the format?

Koordinierungsstelle für IT-Standards

- Business rule:
  - It is mandatory to record month and day of an event
    - Express Month and Day in that order with leading zeros.

- Specification by (two) examples:

```
<data-driven>
  <?xmute mutator="remove" schema-invalid?>
  <monthday>02-15</monthday>
</data-driven>
```

Given Schematron

Koordinierungsstelle für IT-Standards

```
<rule context="/data-driven/monthday">
      <assert
test="matches(.,'([0-1][1-9])-([1-3][0-9])')“
role="fatal“
 id="r1"
       >Not a valid month-day
pattern</assert>
   </rule>
```

- Outcome of:

```
<data-driven>
 <?xmute mutator="remove"
          schema-invalid ?>
 <?xmute mutator="identity"
          schema-valid
          schematron-valid="r1" ?>
 <monthday>02-15</monthday>
</data-driven>
```

Running Example: What's the format?

Koordinierungsstelle für IT-Standards

- Specification by (three) examples:

```
<data-driven>
  <?xmute mutator="remove" schema-invalid ?>
  <?xmute mutator="identity" schema-valid schematron-valid="r1" ?>
  <monthday>02-15</monthday>
</data-driven>
```

Validation Result vs. Expectation	valid	invalid
valid	02-15 as + (TP)	- (FP)
invalid	- (FN)	+ (TN)

How does Schematron work?

```
<data-driven>
  <?xmute mutator="remove" schema-invalid ?>
  <?xmute mutator="identity" schema-valid schematron-valid="r1"
        description="True Positive"?>
  <?xmute mutator="code" values="02-30" schema-valid schematron-invalid="r1"
        description="False Positive" ?>
  <?xmute mutator="code" values="02-03" schema-valid schematron-valid="r1"
        description="False Negative" ?>
  <?xmute mutator="code" values="13-99" schema-valid schematron-invalid="r1"
        description="True Negative" ?>
  <monthday>02-15</monthday>
</data-driven>
```

Validation Result vs. Expectation	valid	invalid
valid	02-15 as + (TP)	02-30 as - (FP)
invalid	02-03 as - (FN)	13-99 as + (TN)

Test Metadata

Three attributes

Koordinierungsstelle für IT-Standards

```
<data-driven>
  <?xmute mutator="code" values="13-99"
            schema-valid schematron-invalid="r1"
            description="True Negative"
            id="t1"
            tags="mandatory simple"?>
  <monthday>02-15</monthday>
</data-driven>
```

- description: free text for humans
- id: uniquely identify test case
- tags: for keywords/categorization/grouping

Declarative Approach

Simple Mutation and Testing Language

Summary

- Declarative approach
  - What to do and what to expect
- Simple language
  - List of Key/Value items
- Implementation independent
- XML Processing instruction
  - Does not interfere with XML validation
  - Allows to base testing on valid instances

Focus on test writing

- Test writer writes example XML and uses the data to ask:
  - “Do the XML Schema and or Schematron express everything for my business need?”
  - “Is the validation correct?”
    - Or “Does it accept wrong data and excludes correct data?”

xml-mutate

A first prototype implementation

Status

- *xml-mutate* for XML **Muta**ting and **Te**sting
- Written in Java
- a command line interface
- Writes a mutation and test report to console.
- The source code is available on GitHub.

CLI and Output

- java -jar xml-mutate.jar
  --schema example/expectation/monthday.xsd
  --schematron s=example/expectation/monthday.sch
  --target test-generated
  example/expectation/monthday.xml

```
Generated 5 mutations from example/expectation/monthday.xml

|#  |Mutation        |Line|Exp|XSD|Exp|Sch|Exp    |Error Message                    |Description    |
|1  |[remove] 1      |3   |N  |Y  |N  |Y  |NA     |                                 |               |
|2  |[identity, noop|5   |Y  |Y  |Y  |Y  |Y      |                                 |True Positive  |
|   |] 2             |    |   |   |   |   |       |                                 |               |
|3  |[code] monthday|7   |N  |Y  |Y  |Y  |r1:N   |Failed expectation assert for r1 |False Positive |
|   |.xml-3-02-30    |    |   |   |   |   |       |                                 |               |
|4  |[code] monthday|9   |N  |Y  |Y  |N  |r1:N   |Failed expectation assert for r1 |False Negative |
|   |.xml-4-02-03    |    |   |   |   |   |       |                                 |               |
|5  |[code] monthday|11  |Y  |Y  |Y  |N  |Y      |                                 |True Negative  |
|   |.xml-5-13-99    |    |   |   |   |   |       |                                 |               |

Mutations run: 5 . Failures: 3 . Errors: 0

------------------------------------------------------
Result: FAILURE
Generated 5 mutations. Passed: 2 . Failed: 3 . Error: 0
```

Mutators

Name	Implementation status
empty	available
add	in planning
remove	available
rename	in planing
change-text	available
whitespace	available
identity	available
code	available
alternative	available
random-element-order	in planning

Conclusion

Koordinierungsstelle für IT-Standards

- Prevents need for developing custom test frameworks
- Simplifies test driven development
  - Unit-, acceptance- and regression testing
  - Test writer can focus on business data and rules
- Integrated test data generation, expectations and test metadata
  - It is simple to add new features
- Implementation independent
  - Alternative processors can implement different feature sets
  - Use different programming languages and technologies.
  - Hence, making it possible that this approach becomes more widely accepted and adopted

Discussion

Koordinierungsstelle für IT-Standards

```
<data-driven>
  <?xmute mutator="code" values="13-99"
          schema-valid schematron-invalid="r1"
          description="True Negative"
          id="t1"
          tags="mandatory simple"?>
  <monthday>02-15</monthday>
</data-driven>
<!– @xmlprague dear #xmlprague attendees
    Is it useful?
    Worth going further with this?
 -->
```

THANK YOU!

Koordinierungsstelle für IT-Standards

Dr. Renzo Kottmann

renzo.kottmann@finanzen.bremen.de

kosit@finanzen.bremen.de

www.xoev.de/de/xrechnung

www.xoev.de / www.osci.de

Disclaimer

- Das Logo XRechnung (im Folgenden als Logo bezeichnet) ist eine eingetragene Design-Marke der Freie Hansestadt Bremen, vertreten durch den Senator für Finanzen (Designanmeldung 402019200826.1 vom 14.08.2019, Veröffentlichung im Designblatt voraussichtlich am 06.09.2019). Die Verwendung des Logos XRechnung ist grundsätzlich genehmigungspflichtig.
- Die Verwendung des Logos für kommerzielle Zwecke ist grundsätzlich nicht gestattet. Ausnahmen bedürfen zwingend der Zustimmung der Rechteinhaberin.