

# **Schematron for Information Architects**

# Contents

<b>Schematron for Information Architects.....</b>	<b>3</b>
Introduction.....	3
Many XML Technologies.....	3
But not so much time.....	3
Business Rules.....	3
Definition.....	3
Examples.....	3
Enforcing business rules.....	3
Schematron.....	3
About.....	3
Implementation.....	4
Use cases.....	4
Information Architects.....	4
Role.....	4
Challenge.....	4
Simple use of Schematron.....	4
Abstract patterns.....	4
Example abstract pattern.....	4
Pattern instantiation.....	5
Example pattern instantiations.....	5
Recommended best practice.....	5
Roles.....	5
Integrate Schematron within a style guide.....	5
Why do you need a style guide?.....	5
Encoding rules within the style guide.....	6
Encoding rules in DITA.....	6
Link the rule to the style guide topic.....	6
The DIM Project.....	6
About.....	6
Deliverables.....	7
Demo.....	7
Conclusions and future work.....	7
Conclusions.....	7
Future Work.....	7
Thank you!.....	8

# Schematron for Information Architects

---

## Introduction

---

### Many XML Technologies

Schema languages

- XML Schema
- DTD
- Relax NG
- NVDL
- Schematron

XML processing languages

- XSLT
- XQuery
- XProc

### But not so much time

We need to make it easy to start working with a technology

This is an attempt to lower the entry barrier for **Schematron**

## Business Rules

---

### Definition

Business Rules are constraints that are not generic enough to be part of a standard but they are relevant for a project or an organization

### Examples

- Do not scale images dynamically in the XML source by using for example a `@scale` attribute.
- Make sure list items do not end with a ";" character.
- Make sure the number of words in a short description stays within some specific limits.

### Enforcing business rules

As part of validation or application processing

As a separate business rules layer:

- between validation and application processing
- usually implemented with **Schematron**

## Schematron

---

### About

- One of the 5 standards that specify schema languages for XML

- A different kind of language - focus on checking patterns rather than defining the structure
- Invented by Rick Jelliffe in 1999
- ISO Standard since 2006 (Part 3 - ISO/IEC 19757 – DSDL – Rule-based validation)
- Not a very complex language - 21 elements in total

## Implementation

Reference implementation - Skeleton

Mainly XSLT based implementations

- Convert the Schematron schema using XSLT to an XSLT script
- Apply the generated XSLT script on the XML document to get the validation results

## Use cases

- Provide error messages that writers/authors can easily understand
- Check rules that cannot be specified in the DTD or XML schema
- Check for common mistakes
- Controlled values - check against an external data source
- Integrity checks across multiple files
- Style guide integration
- Checks for targeting a specific deliverable, for example if you want to publish for mobile devices

## Information Architects

---

### Role

Identify and specify the business rules that the information should follow

- usually these are documented in prose, as part of a style guide

### Challenge

Can we enable them to specify the business rules so they can be also enforced?

While they know the domain, we cannot expect them to know Schematron

## Simple use of Schematron

---

### Abstract patterns

Allows to reuse patterns by making them generic

Provides placeholder parameters that will specify an actual pattern

### Example abstract pattern

```
<pattern id="restrictWords" abstract="true">
  <rule context="$parentElement">
    <let name="words"
        value="count(tokenize(normalize-space(.), ' '))"/>
    <assert test="$words <= $maxWords" role="warn"> It is recommended
      to not exceed <value-of select="'$maxWords'"/> words! You
      have <value-of select="$words"/> <value-of
        select="if ($words=1) then ' word' else ' words'"/>.
  </rule>
</pattern>
```

```

</assert>
<assert test="$words >= $minWords" role="warn"> It is recommended
  to have at least <value-of select="'$minWords'"/> words! You
  have <value-of select="$words"/> <value-of
    select="if ($words=1) then ' word' else ' words'"/>.
</assert>
</rule>
</pattern>

```

## Pattern instantiation

Refer to an abstract pattern and specify values for its parameters

Reduced complexity

Anyone can learn the syntax for this in a few minutes

## Example pattern instantiations

```

<pattern is-a="restrictWords" id="restrictDescription">
  <param name="parentElement" value="shortdesc"/>
  <param name="minWords" value="3"/>
  <param name="maxWords" value="50"/>
</pattern>

```

```

<pattern is-a="restrictWords" id="restrictAbstract">
  <param name="parentElement" value="abstract"/>
  <param name="minWords" value="10"/>
  <param name="maxWords" value="100 "/>
</pattern>

```

## Recommended best practice

Use a library to define a set of abstract patterns (generic rules)

To create a schema just instantiate an abstract pattern from the library

## Roles

- **Schematron developer**
  - defines the library of abstract patterns
  - needs to know Schematron
- **Information Architect**
  - instantiates abstract patterns to define business rules
  - needs to know only what rules are available and their parameters
- **Document writer/author**
  - will be notified as he/she changes the document, as soon as an check fails

## Integrate Schematron within a style guide

---

### Why do you need a style guide?

Document possible problems so you do not run into them again

## Encoding rules within the style guide

Information to be encoded:

- pattern name
- list of parameters
  - parameter name
  - parameter value

## Encoding rules in DITA

In Schematron:

```
<pattern is-a="restrictWords"
  see="http://example.com/styleguide/webhelp/
c_WritingShortDescriptions.html">
  <param name="parentElement" value="shortdesc"/>
  <param name="minWords" value="1"/>
  <param name="maxWords" value="50"/>
</pattern>
```

In DITA:

```
<section audience="rules">
...
<dl>
  <dlhead>
    <dthd>Rule</dthd>
    <ddhd>restrictWords</ddhd>
  </dlhead>
  <dlentry>
    <dt>parentElement</dt>
    <dd>shortdesc</dd>
  </dlentry>
  <dlentry>
    <dt>minWords</dt>
    <dd>1</dd>
  </dlentry>
  <dlentry>
    <dt>maxWords</dt>
    <dd>50 </dd>
  </dlentry>
</dl>
...
</section>
```

## Link the rule to the style guide topic

```
<pattern is-a="restrictWords"
  see="http://example.com/styleguide/webhelp/
c_WritingShortDescriptions.html">
```

## The DIM Project

---

### About

DIM = Dynamic Information Model

Initiated by George Bina / Syncro Soft and Frank Miller / Comtech Services

Contributions from Alex Jitianu / Syncro Soft, Dawn Stevens / Comtech Services, Nico Kutscherauer / data2type and Octavian Nadolu / Syncro Soft

A DITA-based style guide that embeds business rules and other information

#### License

Apache 2.0

#### Availability

GitHub - <https://github.com/oxygenxml/dim>

## Deliverables

- Style guide output
- Extracted Schematron rules file
- oXygen configuration file to integrate the style guide

## Demo

Project layout

Available abstract patterns library

Defining a new rule

Generate the Schematron rules file

Using the Schematron rules

Quick Fixes in action

Link an element to a style guide topic

Use the linking information while editing

## Conclusions and future work

---

### Conclusions

By defining a library of abstract patterns we enable a simpler use of Schematron - just instantiate the abstract patterns, thus lowering the entry barrier for using Schematron.

Having the pattern instantiations defined inside a style guide provides even more benefits and using custom actions make it even easier to instantiate the available generic rules.

### Future Work

- Implement a custom URL handler to see the DITA-based style guide as a Schematron schema

**or**

Implement a pre-processing step similar to extracting embedded Schematron rules from XML Schema or Relax NG

- Define more generic rules (abstract patterns) in the Schematron library
- Integrate Schematron Quick Fixes (see Nico Kutscherauer work at <http://www.schematron-quickfix.com/index.html>)

## Thank you!

---

**Question?**

**Contact information**

george@oxygenxml.com

@georgebina

**Links**

Here you can find some useful resources

**The DIM project**

<http://www.github.com/oxygenxml/dim>

**ISO Schematron**

[c040833\\_ISO\\_IEC\\_19757-3\\_2006\(E\).zip](#)

**Schematron Quick Fixes**

<http://www.schematron-quickfix.com/index.html>