

Schematron

State of the Union

Tony Graham, David Maus, Andrew Sales, Erik Siegel

Schematron Editorial Board

We have well attended users meetups at XML Prague since 2017.

A small group of community representatives organically grew out of the users meetups: Tony Graham, David Maus, Erik Siegel, Andrew Sales.

The four of us maintain the infrastructure, organize community efforts, and discuss possible future developments.

Meet us at this year's users meetup on Saturday or online at the Github organization <https://github.com/schematron>.

What is Schematron?

What is Schematron?

A schema language for rule based validation, part of the ISO Document Schema Definition Languages (DSDL).

It lets you check arbitrary parts of your documents against any number of arbitrary business rules.

In practice Schematron implementations use XPath as document selection and rule evaluation language.

Schematron: Assertions, Rules, and Patterns

The assert element contains a human-readable assertion and has a @test attribute with an expression that tests if the assertion holds true.

The rule element groups assertions and has a @context attribute with an expression that selects a part of the document.

The pattern element groups rules that are related in a user-defined way.

A Schematron example

1. `<schema xmlns="http://purl.oclc.org/dsdl/schematron" queryBinding="xslt3">`
2. `<ns prefix="tei" uri="http://www.tei-c.org/ns/1.0"/>`
3. `<pattern>`
4. `<rule context="tei:rs[@type = 'person']">`
5. `<assert test="@ref and doc(@ref) instance of element(tei:person)">`
6. `A string referencing a person points to the register`
7. `</assert>`
8. `</rule>`
9. `</pattern>`
10. `</schema>`

Schematron: Abstract rules & patterns, properties, ...

Schematron defines more elements to support:

- Schema reuse and composition (import, extends)
- Localization (diagnostics)
- Message templating (value-of, emph, span, dir)
- Rule templating (abstract patterns)
- Schema documentation (title, p)

Why use Schematron?

Schematron is an ISO Standard.

It lets you express functional dependencies in your documents not expressible in other schema languages.

Schematron implementations are build on top of well matured technologies (XSLT and XPath).

It has been heavily in use for more than 15 years.

Why use Schematron?

“Schematron is a feather duster to reach the corners that other schema languages cannot reach.”

Rick Jelliffe

Community

Timeline

1999

Schematron implementation by Rick Jelliffe

2006

ISO Schematron, 1st edition

2014

Schematron Quickfix

2016

ISO Schematron, 2nd edition

2017

Schematron Users Meetup @ XML Prague

2018

Schematron Users Meetup @ XML Prague

2019

Schematron Users Meetup @ XML Prague

2019

New FLOSS implementation in XSLT

2020

Schematron Users Meetup @ XML Prague

2020

ISO Schematron, 3rd edition

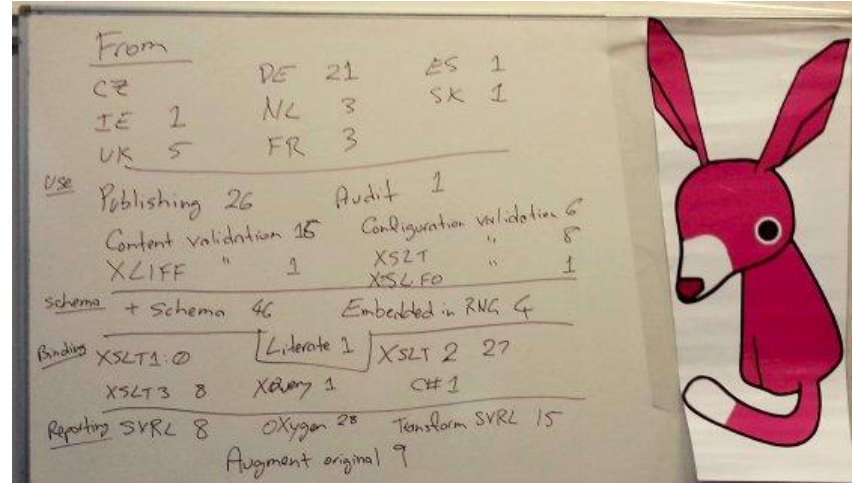
Schematron users

You find users in almost every industry using Schematron for most different tasks

- Finance, healthcare, publishing, digital humanities, libraries, ...
- Content validation, configuration validation, auditing, reporting

Some users regularly meeting at the Schematron Users Meetup XML Prague.

- Sat June 11th 09:30-13:00



A handwritten table on a notepad, with a pink rabbit illustration on the right side. The table is organized into several sections with horizontal lines separating them. The sections are: 'From' (listing countries and counts), 'Use' (listing tasks and counts), 'Schema' (listing schema types and counts), 'Bindings' (listing XSLT versions and counts), and 'Reporting' (listing reporting methods and counts).

From	
CZ	DE 21 ES 1
IE 1	NL 3 SK 1
UK 5	FR 3

Use	
Publishing 26	Audit 1
Content validation 15	Configuration validation 6
XLIFF " 1	XSLT " 8
	XSL-FO " 1

Schema	
+ Schema 46	Embedded in RNG 4

Bindings	
XSLT1:0	Literate 1 XSLT 2 27
XSLT3 8	XQuery 1 CH 1

Reporting	
SVRL 8	Oxygen 28 Transform SVRL 15
	Augment original 9

Schematron resources

Mailing list

<https://schematronist.org>

Rick Jelliffe's Schematron page

<https://schematron.com>

Github Organisation

<https://github.com/schematron>

Awesome Schematron

<https://github.com/schematron/awesome-schematron>

Recent publications

Hillman, Tomos, and Vincent Lizzi. 2020. “Self-Generating Quality Control: A Case Study.”

Holmes, Martin, Kaitlyn Fralick, Kailey Fukushima, and Sarah Karlson. 2019. “How We Tripled Our Encoding Speed in the Digital Victorian Periodical Poetry Project.”

Maus, David. 2019. “Ex-Post Rule Match Selection: A Novel Approach to XSLT-Based Schematron Validation.”

Nadolu, Octavian. 2019. “Taking Schematron QuickFix To The Next Level.”

Siegel, Erik. 2022. “Schematron“, forthcoming

New features, functions, and ideas

Ali, Amer. 2014. “Schematron Based Semantic Constraints Specification Framework & Validation Rules Engine for JSON.”

Dziurlaj, John. 2021. “Streaming Extensions for Schematron.”

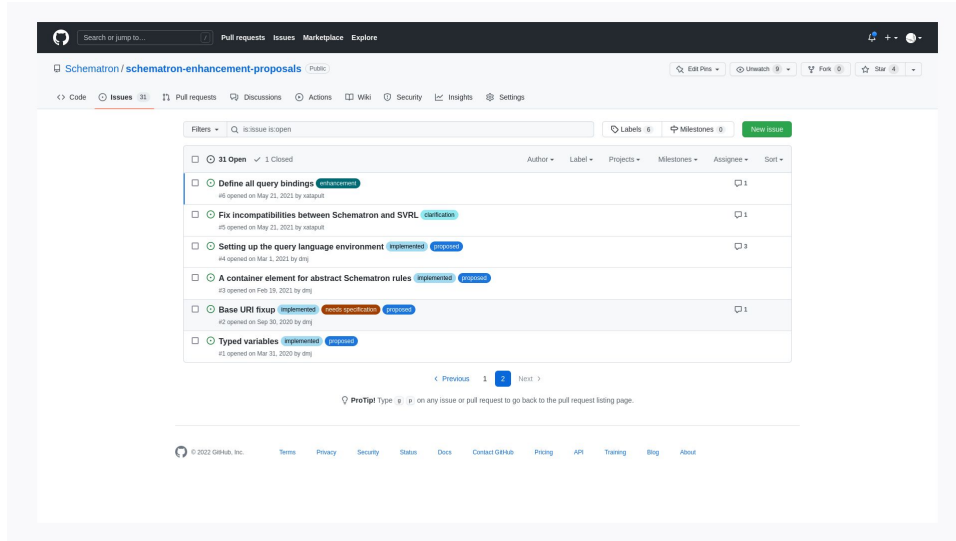
Maus, David. 2021 “A container element for abstract Schematron rules.”

Schematron Enhancement Proposals

A Github repository with a collection of proposed corrections, clarifications and enhancements to the ISO Schematron specification:

[https://github.com/schematron/schematron-enhancement-proposals.](https://github.com/schematron/schematron-enhancement-proposals)

SEP Github Repository



The screenshot shows the GitHub repository page for Schematron/schematron-enhancement-proposals. The page is titled "Issues" and displays a list of 31 open issues. The issues are sorted by "New Issue" and are filtered to show only open issues. The issues are listed in a table with columns for "Author", "Label", "Projects", "Milestones", "Assignee", and "Sort".

Issue Title	Label	Author	Sort
Define all query bindings	enhancement	kataputi	1
Fix incompatibilities between Schematron and SVRL	clarification	kataputi	1
Setting up the query language environment	implemented, discussion	dmg	3
A container element for abstract Schematron rules	implemented, proposal	dmg	1
Base URI fixup	implemented, needs specification, discussion	dmg	1
Typed variables	implemented, discussion	dmg	1

- Issues labeled with erratum, clarification, and enhancement
- Wiki pages with elaborate description and proposed solutions

Example SEP: Correction

Error in ISO 2020, section 3.13 (pattern)

<https://github.com/Schematron/schematron-enhancement-proposals/issues/18>

Section 3.13 of the 2020 specification reads:

pattern unordered collection of rules (3.18) with an optional identifier and ancillary information

This is add odds with section 6.5 clause 4 and 5. The order of rules in a pattern does matter.

Example SEP: Clarification

Define the production rules for SVRL

<https://github.com/Schematron/schematron-enhancement-proposals/issues/7>

The Schematron standard defines SVRL by providing a (RelaxNG) schema only. There is no description on how to get from executing a Schematron-schema to the resulting SVRL. Both the older "skeleton" implementation and the newer "SchXslt" implementation create something that seems obvious. But this still leaves the exact production rules undefined. It's not a good idea to leave important things in standards to being obvious.

Example SEPs: Enhancement

Allow typed variables

<https://github.com/Schematron/schematron-enhancement-proposals/issues/1>

The current specification of Schematron does not provide means to declare the required type of a variable. Users of Schematron are working around this shortcoming by defining variables using the `xsl:variable` instruction and relying on the underlying processor to copy these variable declarations to the validation stylesheet.

Example SEPs: Enhancement

Define parameters for abstract patterns

<https://github.com/Schematron/schematron-enhancement-proposals/issues/8>

There is no way you can define which parameters are there for an abstract pattern. Therefore the processor cannot check that an abstract pattern invocation is correct.

Moving forward



Schematron and ISO

Schematron was incorporated into the ISO DSDL family of specifications in 2006.

According to its then chief designer, Rick Jelliffe, he made a draft revision in 2010 which was adopted six years later in 2016.

Forward another four years and three things happened

1. A third revision of ISO Schematron was published, adding XSLT 3.0 and XPath 3.0;
2. The DSDL working group of the responsible ISO subcommittee disbanded;
3. ISO made this revision of the standard non-public.

With the working
group disbanded,
corrections and
enhancements go
nowhere.

Moving forward

Our goal is to initiate work on a new ISO Schematron edition that incorporates corrections, clarifications, and enhancements from the Schematron Enhancement Proposals.

A new work item for the ISO SC 34 Plenary in September 2022 will be proposed this month.

If everything goes according to plan, a new edition of ISO Schematron will be ready in 24 to 36 months.

Call to Action

Join the discussion of the Schematron Enhancement Proposals. Input for the new work item proposal is due this month.

If you want to get involved with the ISO process, you can do so through your national body.

Meet us at this year's Schematron Users Meetup tomorrow at 09:30.