

# SVRL 2.0

A new beginning?

# Background (= problem statement)


- Schematron Validation Report Language (SVRL)
- Implementations can, but **aren't required**, to output SVRL
- **Flat** format
  - Structure doesn't reflect that of schema used (no grouping of rules etc)
- **Verbose** format
  - e.g. one `fired-rule` element per matching node
- It **isn't defined normatively** in the standard


# Both optional *and* non-normative?



- At the outset, this wasn't such(?) an issue, BUT:
- Validation report as XML seems second nature now
  - Wasn't always so: parsers of yore didn't typically do this
- Users/implementations/integrations have become reliant on SVRL
- **Isn't it high time to make SVRL a normative part of the standard?**

# Flat structure (1)

Lack of structure can be [a real problem](#):

Provide a way to express the relationship of patterns/groups, rules, assertions in SVRL #82 

 Open

 **dmj** opened on Sep 19, 2025 Member 

Currently the order of top-level elements of an SVRL report is implementation dependent such that different implementations may generate the same elements in a different order. The elements `svrl:active-pattern`, `svrl:active-group`, `svrl:active-rule`, `svrl:failed-assert`, and `svrl:successful-report` stand for their own. This makes it impossible to relate assertions, rules, patterns, and groups without knowledge of the originating Schematron schema.

# Flat structure (2)

Leads to (viable) workarounds, e.g.

```
<svrl:failed-assert patternId='foo' ruleId='bar' id='blort'>
```

But isn't nesting much simpler and more natural here? e.g.

```
<svrl:active-pattern id='foo'>  
  <svrl:fired-rule id='bar'>  
    <svrl:failed-assert id='blort'>[...]</svrl:failed-assert>  
  </svrl:fired-rule>  
</svrl:active-pattern>
```

# Verbosity

- It **can** be an issue (with node- or error-heavy documents)
- Provide summary/digest capability to lessen this, e.g.

```
<svrl:-fired-rule id='bar' count='25'>
  <!-- emit first instance of assertions only, as an example? -->
  <svrl:failed-assert id='blort'>[...]</svrl:failed-assert>
  <svrl:failed-assert id='wibble'>[...]</svrl:failed-assert>
</svrl:-fired-rule>
```

- Implementation-configurable? Markup facility still needed in SVRL
- We can/should also learn from [XVRL](#), as necessary

# In summary

- Make SVRL mandatory
- Make the SVRL Annex normative
- Add structure
- Reduce verbosity

For reference: <https://github.com/Schematron/schematron-enhancement-proposals/issues/95>

Thank you!

Any questions?