NEMSIS TAC Whitepaper

NEMSIS V3 Schematron Guide

# Date

Version: November 21, 2011

**Authors**

Shaoyu Su – NEMSIS Software Developer

[su.shaoyu@hsc.utah.edu](mailto:su.shaoyu@hsc.utah.edu)

This document is to address several issues in developing NEMSIS Schematron rules and releasing national rules.

1. Error Levels
   1. We propose three severity levels of business rule violation: FATAL, ERROR, WARNING
      1. Field-level products will contain two security levels (FATAL and WARNING)
      2. Products that aggregate cases can contain all three
   2. To minimize confusion, the string values for them are [FATAL], [ERROR] and [WARNING], respectively.
   3. [FATAL] marks fatal error: any NEMSIS XML file has FATAL error reported by Schematron cannot be processed further. The submission is rejected. The error must be fixed and file re-submitted.
   4. [ERROR] marks “regular” error: data receiver should provide the feedback to submitter. But it is up to receiver to decide whether to reject the submission or not.
   5. [WARNING] marks “irregularity”: data receiver should provide the feedback to submitter. The receiver should accept the submission, regardless of the action (or no action) taken by the submitter.
2. How to identify the error level of a rule?
   1. In Schematron, a PATTERN could have multiple RULEs. A RULE could have multiple ASSERTs. Unfortunately, in the current Schematron release as of writing, Schematron doesn’t provide a mechanism to identify the level of error (violation reported by ASSERT statement).
   2. For PATTER, RULE, REPORT and ASSERT elements, although each has several attributes, but not all attributes defined in .sch rule file will be included in the result report XML file.
   3. Other Schematron adopters also face similar situation of lacking an error level indicator.
   4. Attribute “role” can appear only in RULE, REPORT and ASSERT elements. According to Schematron ISO specification (<http://www.schematron.com/iso/P8.html>), it is “A name describing the function of the assertion or context node in the pattern. If the assertion has a subject attribute, then the role labels the arc between the context node and any nodes which match the path expression given by the subject attribute. An implementation is not required to make use of this attribute.” The Schematron result XML file does copy the value of attribute “role” from rule definition file. This is the most detailed description I could find about this attribute: <http://markmail.org/message/udjpvrxdngvmaoql>. (By the way, Rick Jelliffe is the main/sole maintainer of Schematron. Dave Pawson’s tutorial for Schematron is awesome.)
   5. If the community could agree to use value in attribute “role” as an error level indicator instead of the proposed “arcing” attribute, we can adopt this mechanism to determine error level:
      1. In Schematron rule file, for any NEMSIS Schematron RULE/ASSERT element, specify the attribute of “role” using one of the error level string values. The error message should start with same string value.
      2. In result XML file, for any failed assertion (rule violation), determine its “role” attribute. If not specified, check parent RULE’s “role” attribute.
      3. If the “role” attribute matches one from our list of predefined string values, use that. Otherwise ignored.
      4. If we still cannot find matching error level after previous step, check the message in failed-assert: if the message starts with one of the predefined string value, use it.
      5. If we still cannot determine error level, use “[ERROR]” as default level.

I understand that if step i is properly enforced, we don’t need step iii, iv, v at all. They are provided as a fall-back mechanism. Even step iv alone is sufficient to determine error level: but it is convenient to use a XML processor to process Schematron rule file and document any rule’s severity level.

1. The first NEMSIS TAC published national Schematron rule file will check for structural errors only. They are all marked as “FATAL”. You might note that three abstract patterns are used: first one to check elements having PN/NV/Nil attributes, second one to check elements having only NV/Nil attributes, and the last one to check state required elements. Since in our current XSD design, only state required elements have only NV/Nil attributes, so the second abstract pattern is not instantiated at all.
2. For obvious reason, national Schematron rule file will be updated when NEMSIS XSD updates.

A simple example of “solid” NEMSIS rule based on discussion above, with error level highlighted:

|  |
| --- |
| <iso:rule context="nem:dAgency.03" role="[ERROR]">  <iso:let name="selfValue" value="normalize-space(.)" />  <iso:assert role="[ERROR]" test="if ($selfValue = ‘Shaoyu Su’) then true() else false()">  [ERROR]: <iso:name/> is not valid because its value is not ‘Shaoyu Su’.  </iso:assert>  </iso:rule> |

And this is the input XML file:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  <EMSDataSet xsi:schemaLocation="http://www.nemsis.org EMSDataSet\_v3.xsd" xmlns="http://www.nemsis.org" xmlns:n1="http://www.nemsis.org" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <dAgency.03>02345</dAgency.03> <!—first record will trigger violation--  <dAgency.03>Shaoyu Su</dAgency.03> <!—second record is good -->  </EMSDataSet> |

Then this is the result report file, with comments:

|  |
| --- |
| <svrl:active-pattern document="file:/C:/ssu/Projects/NEMSIS/V3/schematron/sample6.xml"  name="Just want to make sure dAgency.03 has some value"/>  <svrl:fired-rule context="nem:dAgency.03" role="[ERROR]"/>  <svrl:failed-assert test="if ($selfValue = 'Shaoyu Su') then true() else false()" role="[ERROR]"  location="/\*[local-name()='EMSDataSet']/\*[local-name()='dAgency.03'][1]">  <svrl:text>  [ERROR]: dAgency.03 is not valid because its value is not 'Shaoyu Su'.  </svrl:text>  </svrl:failed-assert> <!—error report for first record-->  <svrl:fired-rule context="nem:dAgency.03" role="[ERROR]"/> <!—second record is good--> |