

Element	Beschreibung
schema	Das Dokument Element
title	Eine menschlich lesbare Beschreibung über den Zweck der Prüfung
pattern+	Eine Liste von verbunden Regeln, bietet den Context an
rule+	Eine Liste von Assertion zum Context
Assert or report	Deklariert Bedingungen die getestet werden müssen stellt Nachrichten bereit die zurückgegeben

XSLT2 weiterverarbeitet. Die zu prüfenden Daten liegen in der XML Datei MyData.xml vor, die nun an der Reihe ist. Als Ergebnis erhält man die Datei „result.xml“.

Sofern sich an den Businessregeln nichts mehr ändert, ist bei der Qualitätsprüfung der Datenfiles nur mehr der letzten Schritt für alle weiteren Files durchzuführen.

#### Tools

Eine Schematron Prüfung schaffen auch verschiedenen XML Editoren, dazu zählen etwa oXygen [2] – es ist nativ dazu in der Lage, oder auch XML Spy [3] mit dem XML ValidatorBuddy [4] Plugin. Von beiden Tools wird eine 30 Tage Testversionen angeboten.

#### Libraries für .net

nMatrix [5] ist eine .net Implementation unter der Sourceforge Common Public Licence 1.0.

.net unterstützt XSLT 2.0 leider noch immer nicht nativ, Saxon9 hilft dabei. Es stellt eine Command Line als auch eine Library für .net und Java bereit.

#### Fazit

Oft steht man vor der Frage, wie man komplexe XML Daten vollständig auf ihre Richtigkeit prüft. Implementiert man sein Regelwerk im Code selber oder versucht

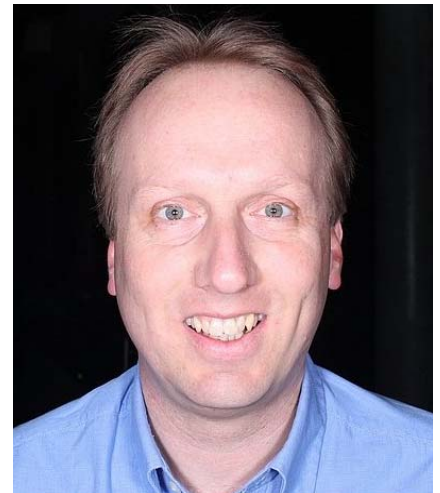
man die bekannten Regeln in einer Rule Engine abzulegen, wird es nicht weniger komplex. Man hat außerdem auf eine schwer zu wartende, eventuell proprietäre Lösung gesetzt, die zudem keinem Standard entspricht. Ändert sich die Regeln zu häufig, bleibt viel Zeit bei der Wartung liegen. Schematron bietet durch seine ISO Norm einen Standard, auf den bereits viele Bereiche der Industrie setzen. Es ist Plattform unabhängig und es stehen Open Source sowie lizenzpflichtige Tools und Libraries bereit. Man sollte Schematron als Ergänzung aber nicht als Konkurrenz zu anderen Schemasprachen einsetzen, um potentiell die Möglichkeit zu haben, komplexe Datenvalidierungen vorzunehmen.

#### Links & Quellen

- [1] <http://www.schematron.com/>
- [2] <http://www.oxygenxml.com/>
- [3] <http://www.altova.com>
- [4] <http://www.xml-tools.com/ValidatorBuddy.htm>
- [5] <http://sourceforge.net/projects/dotnetopencsrc/>
- [6] <http://relaxng.org/>
- [7] <http://saxon.sourceforge.net/>

#### Autorenbox

**Thomas Reinwart** verfügt über umfangreiche Berufserfahrung auf dem IT Sektor. In den letzten 20 Jahren war er in den Bereichen Softwareentwicklung, Softwaredesign, Architekt und als Consultant tätig. Technischer Fokus ist derzeit Microsoft .net und SQL Server, wo er alle aktuellen Microsoft Zertifizierungen hat.



```
"c:\Program Files\Saxonica\SaxonHE9.5N\bin\Transform.exe" -s:MyCheck.sch -xsl:iso-schematronfiles\iso_dsd1_include.xsl -o:stage1.sch
"c:\Program Files\Saxonica\SaxonHE9.5N\bin\Transform.exe" -s:stage1.sch -xsl:iso-schematronfiles\iso_abstract_expand.xsl -o:stage2.sch
"c:\Program Files\Saxonica\SaxonHE9.5N\bin\Transform.exe" -s:stage2.sch -xsl:iso-schematronfiles\iso_svrl_for_xslt2.xsl -o:stage3.xslt
"c:\Program Files\Saxonica\SaxonHE9.5N\bin\Transform.exe" -s:MyData.xml -xsl:stage3.xslt -o:result.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <svrl:schematron-output title="Obstkorb Check" schemaVersion="ISO19757-3" xmlns:xhtml="http://www.w3.org/1999/xhtml"
  xmlns:iso="http://purl.oclc.org/dsdl/schematron" xmlns:schold="http://www.ascc.net/xml/schematron" xmlns:saxon="http://saxon.sf.net/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:svrl="http://purl.oclc.org/dsdl/svrl">
  <!-- -->
  <svrl:active-pattern document="MyData.xml"/>
  <svrl:fired-rule context="product"/>
  - <svrl:failed-assert location="/arcticle[1]/product[1]" test="title!='Orange' and country='Austria'">
    <svrl:text> In Österreich wachsen keine Orangen. </svrl:text>
  </svrl:failed-assert>
  <svrl:fired-rule context="product"/>
  - <svrl:failed-assert location="/arcticle[1]/product[2]" test="title">
    <svrl:text> Das Produkt muss einen Titel haben. </svrl:text>
  </svrl:failed-assert>
  - <svrl:failed-assert location="/arcticle[1]/product[2]" test="title!='Orange' and country='Austria'">
    <svrl:text> In Österreich wachsen keine Orangen. </svrl:text>
  </svrl:failed-assert>
  <svrl:fired-rule context="product"/>
  - <svrl:failed-assert location="/arcticle[1]/product[3]" test="weight>'0'">
    <svrl:text> Gewichtskontrolle durchführen, Artikel ist leer. </svrl:text>
  </svrl:failed-assert>
  <svrl:fired-rule context="product"/>
  - <svrl:failed-assert location="/arcticle[1]/product[4]" test="state='fresh'">
    <svrl:text> Das Obst muss frisch sein. </svrl:text>
  </svrl:failed-assert>
  <svrl:fired-rule context="product"/>
</svrl:schematron-output>
```

Abbildung4: Ergebnis der Schematron Prüfung „result.xml“