

Schematron QuickFix im Oxygen XML Editor

Vergleich zwei verschiedener Implementierungen

Donnerstag, 16. November 2017

Agenda



- Was ist Schematron?
- Was ist Schematron QuickFix?
- Was für Implementierungen gibt es?
 - Build-In-Implementierung des Oxygen
 - Escali Oxygen Plugin
- Was ist das Escali-Projekt?
- Vergleich der beiden Implementierungen anhand von Beispielen
- Nächste Ziele

Schematron - Review



- Was ist Schematron?
 - ISO Standard, initiiert von Rick Jelliffe
 - Ergänzende Validierungssprache zur Bestimmung sehr spezifischer Validierungsregeln
 - Ergänzung zu DTD / XSD
 - Geeignet für alle Konventionen, die beim Bearbeiten von XML-Daten eingehalten werden sollen.
- Beispiele:
 - Eine ID muss der Hierarchie, der Position und/oder dem Inhalt entsprechend gesetzt werden
 - Meta-Daten (z.B. `product="Produkt1"`) haben Auswirkungen auf die Struktur des Dokuments
 - Mikrotypographie: Schreibweisen von Einheiten („3 m1“) oder Abkürzungen (z. B.)

- „Quick fix“
 - eine automatische Behebung eines spezifischen Fehlers.
 - Korrekturvorschläge wie bei der Rechtschreibprüfung
- Schematron QuickFix
 - Erweiterungssprache für Schematron
 - Definiert QuickFixes für Schematron-Fehler
 - Standardisiert mit einer W3C-Note:
 - <https://www.w3.org/community/quickfix/>
- Beispiele:
 - Ersetze die fehlerhafte ID durch eine aus dem Kontext heraus kalkulierte ID
 - Konvertiere fehlerhaftes Datumsformat in `xs:date`
 - Ersetze fehlerhafte Zeichen

Implementierungen

- oXygen XML Editor seit der Version 17.0
 - Bisher die einzige Editor-basierte Implementierung
 - Basiert auf XQuery Update
 - Mit der Version 18.1 werden die wichtigsten Features von Schematron QuickFix unterstützt

Escali-Projekt



- Escali
 - Referenz-Implementierung
 - Commandline-Implementierung basierend auf XProc
 - Im Browser: escali.schematron-quickfix.com
 - **Escali Oxygen Plugin**
 - <https://github.com/schematron-quickfix/escali-package/>

Vergleich Escali / Oxygen

- Warum eine „neue“ Implementierung?
 - Oxygen ist momentan de-facto-Standard
 - Evaluation, für welche Features tatsächlich auch Bedarf existiert
 - Die folgenden Beispiele sollen aufzeigen, dass die Sprache noch mehr Potential bietet, als der Oxygen momentan unterstützt.

Mehrere QuickFixes auf einmal

- Oxygen kann nur einen QuickFix auf einmal ausführen
 - Zwei QuickFixes könnten sich in die Quere kommen.
 - Mehrere QuickFixes könnten ein Dokument invalide machen.
- Escali bietet die Möglichkeit, mehrere QFs zu selektieren und auszuführen
 - Es kann sehr praktisch sein.
 - Anwendung: 100 gleiche Fehler, 99 müssen auf die gleiche Weise gelöst werden.
 - Ein QF gibt nie die Garantie, dass das Dokument danach valide ist.

UserEntries – Datentypen

- Was ist ein UserEntry?
 - Spezieller Parameter für den QuickFix
 - Wird vom User gesetzt, beim Ausführen des QFs
 - Use Case: Ein Wert ist invalide. Ein QF will den Wert neu setzen. Dazu erhält er einen UE, mit dem der Nutzer den neuen Wert setzen kann.
 - Einem UE kann ein Typ zugewiesen werden.
- Oxygen
 - Basis-Unterstützung
 - Keine Unterstützung von Datentypen
- Escali Plugin
 - Unterstützt die Typen `xs:string`, `xs:integer`, `xs:double`, `xs:date`, `xs:time`, `xs:boolean`
 - Bietet abhängig vom Typ unterschiedliche Eingabemasken an

UserEntries – Enumerations

- Häufig reicht ein generischer Typ nicht aus
- In vielen Fällen sind nur bestimmte Werte erlaubt.
- Wünschenswert wäre eine Drop-Down-Liste mit allen erlaubten Werten („Enumeration“).
- Die Sprache enthält noch keine vollständige Struktur dafür.
- Feature des Escali Plugins
 - Der Escali bedient sich hier eines Tricks
 - UEs können einen Defaultwert haben, der per XPath bestimmt wird.
 - Eine Eingabemaske kann dann diesen Wert als Default verwenden.
 - Das Escali Plugin erzeugt eine Enumeration, wenn ein UE eine Sequenz an Werten als Defaultwert hat.

Umgang mit Entities

- Generelles Problem bei der XML-Verarbeitung
 - Erster Schritt beim XML-Parsen: Auflösen der Entities
 - Kein Zugriff auf Entities mit XPath/XSLT
- Herausforderung bei SQF:
 - Erhalt aller unbeteiligten Entities beim Ausführen der QFs
 - Oxygen und Escali unterstützen das
- Probleme gibt es dennoch:
 - Kopieren oder Verschieben von Knoten löst die Entities auf
 - Hierzu wird in der nächsten Version des Standards das `<sqf:copy-of>`-Element eingeführt mit einem `unparsed-mode`-Attribut.
 - Ist der `unparsed-mode` aktiv, sollten die Knoten kopiert werden, ohne dass Entities aufgelöst werden.
- Oxygen:
 - Hat bereits angekündigt, diesen Mode wohl vorläufig nicht zu unterstützen.
- Escali Plugin enthält einen ersten Draft für eine Unterstützung

- Schematron-Probleme in der Mikrotypographie
 - Schematron kennt nur Knoten-basierte Fehler-Locations
 - Wenn einzelne Text-Snippets (einzelne Zeichen, Abkürzungen, Mengenangaben) die Fehlerursache sind, gibt Schematron maximal pro Textknoten, eher pro Absatz eine Fehlermeldung aus
 - Das tatsächliche Text-Snippet muss manuell gefunden werden
 - Mehrere Vorkommen des gleichen Text-Snippets in einem Textknoten / Absatz werden als ein Fehler zusammengefasst.
 - SQF kann deshalb auch nur ein QF pro Fehler anbieten, der dann alle Text-Snippets auf einmal korrigieren muss.
- Escali-Erweiterung für Schematron
 - Erlaubt Textknoten per Regex aufzusplitten
 - Jeder Regex-Treffer wäre dann ein Text-Snippet, der mit Schematron-Mitteln geprüft werden kann
 - Die Fehlermeldung wird für den Text-Snippet angezeigt.
 - QFs können den betroffenen Text-Snippet einzeln behandeln

Nachteile des Escali Plugins

- Es ist ein Plugin
- Masterfiles-Validierung wird (noch) nicht unterstützt
 - Die Oxygen-API liefert dazu zu wenige Informationen
- Schema-Assoziierung
 - Im Oxygen gibt es viele Möglichkeiten, ein XML-Dokument mit einem Schematron-Schema zu verknüpfen (u.a. Framework, Validierungsszenario, `<?xml-model?>-PI`)
 - Die Oxygen-API bietet keine Informationen, welches Schematron-Schema mit dem XML-Dokument verknüpft ist
 - Das Escali-Plugin kann deshalb nur auf die `<?xml-model?>-PI` zurückgreifen
 - Zusätzlich gibt es eine Association Table, ähnlich der Association Rules eines Frameworks.
- Doppelte Validierung
 - Das Escali Plugin kann nur nachträglich die Validierung des Oxygens manipulieren
 - Kennt der Oxygen das Schematron auch, findet eine doppelte Validierung statt.

Nächste Ziele

- Vortrag bei der XUGS
 - am 29.11.2017 um 18 Uhr hier an der HdM, im Hörsaal 204
 - Ausführlichere Betrachtung von SQF / Oxygen / Escali
- Lokalisierung von SQF
 - Den W3C-Note-Standard vollenden und veröffentlichen
 - Umsetzung im Escali
- Support von Schematron 2016
- Masterfiles-Validierung
- Weitere Plugins in anderen Editoren
 - Das Eclipse-Plugin ist so aufgesetzt, dass es relativ leicht in andere Editoren portiert werden kann (keine komplette Neuentwicklung)

Weitere Links

- SQF Website
 - www.schematron-quickfix.com
 - Weitere Escali-Schematron-Erweiterungen:
http://www.schematron-quickfix.com/escali/escali-ext_en.html
- Github-Organisation
 - <https://github.com/schematron-quickfix/>
- W3C-Gruppe
 - <https://www.w3.org/community/quickfix/>
- oXygen-Dokumentation
 - <https://www.oxygenxml.com/doc/versions/19.1/ug-editor/topics/schematron-quick-fixes.html>

Fragen?