Das iJUG Magazin

Java aktuell

Sommer 2011

Java aktuell

Magazin der Java-Community

- Neu: GlassFish 3.1
- Single Sourcing in Java
- Nützliche Java-Tools
- Entwickeln mit Scala

Kommt es zum Bruch?

Community

ORACLE

Sonderdruck

Erfahrungen, Ideen und Lösungen für Java-En

iJUG Verbund

www.ijug.eu

D: 4,90 EUR A: 5,60 EUR CH: 9,80 CHF Benelux: 5,80 EUR

Inhalt

3 Editorial

5 „Die langfristige Java-Strategie von Oracle ist mir nicht transparent genug ...“ Interview mit Dr. Michael Paus, Vorsitzender der Java User Group Stuttgart

7 Das Java-Tagebuch
Andreas Badelt

10 Source-Talk-Tage

11 Kommt es zum Bruch?
iJUG e.V.

12 „Any place – any time“ Interview mit Rolf Scheuch, Chief Strategy Officer (CSO) der OPITZ CONSULTING GmbH

14 Java-Anwender wünschen sich neue Impulse von der JavaOne für die Community

15 GlassFish 3.1
Markus Eisele

18 Das Cargo-Kult-IT-Problem
Oliver Szymanski

19 PDF-Dokumente mit iText
Gunther Petzsch

21 Java Forum Stuttgart

22 Google macht Eclipse ein Geschenk – WindowBuilder Pro
Jürgen Thierack

25 Scala – Polyglott-Programmierung mit Java
Dario Giancola, Dr. Christine Müller und Dr. Normen Müller

29 Single Sourcing in Java: Desktop-Anwendung und Web-Applikation aus einer Quelle
Björn Christoph Fischer & Oliver Zandner

32 Orientierungsfragen rund um Oracle
Till Brügelmann

33 Concurrency – Durchsatz steigern
Dr. Stefan Koch

36 Des Entwicklers kleine nützliche Helfer ...
Daniel van Ross

39 Schematron: XML mit Präzision
Tobias Israel

42 Initiierung und Motivation von JSRs durch Open-Source-Systeme
Bernd Müller

44 Mehr Rapid Java mit XDEV 3
Gerald Kammerer

51 Quo vadis Java?
Wolfgang Weigend

54 Camel in Action
gelesen von Kai Wähner

55 Die Integrationskraft von Java in der Praxis
Hans Niedermeier

60 Die iJUG-Mitglieder stellen sich vor

62 Testen mit JUnit
Gerald Kammerer

28 Vorschau

41 Unsere Inserenten

50 Impressum

„Any place – any time“: Java spielt bei deren Entwicklung eine tragende Rolle, Seite 12

GlassFish 3.1: Oracle präsentiert den ersten hochverfügbaren Java-EE-6-Server, Seite 15

Dies ist ein Sonderdruck aus der Java aktuell. Er enthält einen ausgewählten Artikel aus der Ausgabe 03/2011. Das Veröffentlichen des PDFs bzw. die Verteilung eines Ausdrucks davon ist lizenzfrei erlaubt.
Weitere Informationen unter www.ijug.eu

Schematron: XML mit Präzision

Tobias Israel, buschmais GbR

Die Realisierung technischer Schnittstellen auf Basis von XML hat sich einen festen Platz erobert. Mithilfe aktueller Werkzeuge und Frameworks sind hervorragende Möglichkeiten geschaffen, Geschäftspartnern die Anbindung an die eigene IT-Infrastruktur zu ermöglichen.

Die automatisierte Abwicklung von Geschäftsprozessen erfolgt nicht selten auf der Basis vertraglicher Vereinbarungen. Die Messlatte für die Qualität der betroffenen Schnittstellen liegt damit entsprechend hoch. Im Folgenden soll der Frage nachgegangen werden, wie Integritätsbedingungen einer XML-Schnittstelle zielführend erfasst und überprüft werden können.

Im allgemein akzeptierten Kontext der Webservice-Technologien bilden die Web Services Description Language (WSDL), das Simple Object Access Protocol (SOAP) sowie das XML-Schema die Eckpfeiler einer jeden XML-Schnittstelle. Bei WSDL und SOAP weisen ein entsprechendes Verständnis der Spezifikationen und die Kenntnis angrenzender Standards des „WS-*"-Universums den Weg hin zum sauber gearbeiteten Korsett der Schnittstelle. Der eigentliche Knackpunkt liegt jedoch im Entwurf der auszutauschenden XML-Nachrichten. Mit ihnen wird die fachlich motivierte Anbindung an die eigene Prozesslandschaft realisiert.

Ein häufig auftretender Stolperstein ist dabei oft schon der Wiederstreit zwischen Flexibilität und strukturell verankerter Präzision. Die Frage, ob wegen jedes neu über die Lieferkette abgewickelten Geschäftsfalles auch die Schnittstelle angepasst werden soll, lässt sich eben nicht so einfach beantworten. Egal wie man es dreht und wendet, die Beschreibung der Datenstrukturen in XML-Schema wird in jedem Fall eine Restmenge an gewollten oder ungewollten Grauzonen zurücklassen.

Eine häufig ins Feld geführte Waffe hiergegen ist die Produktion einiger Kilogramm beschriebenen Papiers. Mit ihm soll den externen Schnittstellen-Partnern die Integritätsbedingungen der Schnittstelle nahegebracht werden. Die unterschiedlichen Standpunkte, Sprach- und Prozesswelten sorgen jedoch nicht selten für ein breit gefächertes Spektrum an Missverständnissen. Abhilfe kann hier die Flankierung der grammatikbasierten XML-Schema-Beschreibungen mit einer kontextabhängigen Validierung schaffen. Möglichst nah an der technischen Schnittstellen-Beschreibung kann so das umfangreiche Regelwerk der Integritätsbedingungen erfasst und später auch überprüft werden.

Gestatten: Schematron

Die Sprache Schematron bietet die Möglichkeit, XML-Dokumente regelbasiert zu validieren. Basis eines Schematrons ist – anders als bei XML-Schema – keine formale Grammatik. Die Beschreibung beruht vielmehr auf der Definition von Regeln, in denen Bezug auf einen bestimmten Kontext innerhalb des XML-Baums genommen wird.

Die Notation einer Schematron-Beschreibung beruht – wie auch bei XML-Schema – auf der XML (siehe Listing 1). Weiterhin ist eine Abfragesprache zur Ermittlung der Kontexte und zur Formu-

```
<s:schema xmlns:s="http://purl.oclc.org/dsdl/schematron">
  <s:title>Issue management service interface V1.00.0
        integrity constraints</s:title>
  <s:ns prefix="t"
      uri="http://example.com/IssueMngtService/1.0"/>
  <s:pattern>
    <s:rule context= "/t:CreateIssueRequest/t:ServiceLevel">
      <s:assert test="(t:ServiceLevel='EXPRESS') and
                (count(t:Issue) = 1)">
        Only one issue per request is permitted for
        service level EXPRESS.
      </s:assert>
    </s:rule>
    <s:rule context=
            "/t:CreateIssueRequest/t:Issue/t:Detail">
      <s:report test="contains(t:Key, 'SysERR05') and
                not(contains(t:Key, 'SysERR05-1'))">
        To avoid further enquiries about your issue,
        please provide sub code 'SysERR05-1' for main error
        code 'SysERR05'.
      </s:report>
    </s:rule>
    </s:pattern>
</s:schema>
```

Listing 1: IssueManagementServiceConstraints.sch

lierung der Zusicherungen und Tests erforderlich. In der Regel kommt an dieser Stelle XPath zum Einsatz. In Abhängigkeit von der eingesetzten Schematron-Implementierung kann jedoch auch auf die Mächtigkeit von XPath 2.0 oder XQuery zurückgegriffen werden.

Jede Schematron-Beschreibung wird mit dem Wurzelelement <schema> eingeleitet. Durch Angabe von <title> kann man optional eine verbale Beschreibung ergänzen. Die in den XPath-Ausdrücken verwendeten Namespaces und deren Präfixe sind zunächst mithilfe von <ns>-Elementen zu definieren. Nun folgen die innerhalb von <pattern>-Elementen gruppierten Regeln <rule>. Das Context-Attribut jedes Rule-Elements definiert den Bezugspunkt für die eingebetteten Zusicherungen. Diese zu formulieren ist Aufgabe der beiden Elemente <assert> und <report>.

Ein Assert beschreibt eine Zusicherung, die innerhalb des zuvor bestimmten Kontextes durch ein XML-Dokument erfüllt sein muss. Validiert der XPath-Ausdruck zu „false“, wird die angegebene Fehlermeldung ausgegeben. Bei Report-Elementen ist hingegen der Erfolg der gemachten Zusicherung für die Ausgabe des Textes entscheidend. Die Spezifikation schreibt ihnen daher auch einen informellen Charakter zu.

Asserts und Reports ermöglichen über die Prüfung einer technisch korrekten Umsetzung der XML-Datenformate hinaus auch einen Bezug zu den dahinter liegenden, fachlich motivierten Aspekten. Einzige Voraussetzung: Die Fehlermeldungen und Hinweise sind fachlich korrekt formuliert. Ganz nebenbei sorgt dieser Umstand dafür, dass komplexe Details auch schon in der Analysephase in verständlicher Form vorliegen.

Schematron-Prüfung

Das Anwenden eines Schematrons auf konkrete XML-Dokumente erfordert eine entsprechende Implementierung dieses Standards. Gleich welchen technologischen Kontext das eigene Projekt in diesem Punkt vorgibt, die Einstiegshürde ist gering. Die Verarbeitung wird durch die meisten Frameworks als eine Reihe von Transformationsschritten durchgeführt. Die Umsetzung mit einer neutralen Technologie wie XSLT liegt nicht nur auf der Hand, sondern wurde mit der Referenzimplementierung auch genauso umgesetzt.

```
<?xml version="1.0" encoding="UTF-8"?>
<t:CreateIssueRequest
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:t='http://example.com/IssueMngtService/1.0'
  xsi:schemaLocation='http://example.com/IssueMngtService/
                      1.0 IssueManagementService.xsd'>
  <t:CustomerID>20000000300340 1</t:CustomerID>
  <t:ServiceLevel>EXPRESS</t:ServiceLevel>
  <t:Issue>
    <t:Detail>
      <t:Key>SysERR01</t:Key>
      <t:Value>12AXU3B</t:Value>
    </t:Detail>
    </t:Issue>
    <t:Issue>
       <t:Detail>
          <t:Key>SysERR02</t:Key>
          <t:Value>34A45H8E</t:Value>
       </t:Detail>
    </t:Issue>
</t:CreateIssueRequest>
```

Listing 2: IssueManagementService.xml

Abbildung 1 stellt den Ablauf der Prüfung dar. Die erstellte Schematron-Beschreibung der abzusichernden Integritätsbedingungen wird in der ersten Phase mithilfe generischer XSLT-Skripte der Schematron-Implementierung transformiert. Das Ergebnis dieser Transformation ist ein weiteres XSLT-Skript. Dieses wird zur Überprüfung und Erstellung eines Testreports in Phase 2 auf die zu validierenden XML-Daten angewendet. Das Ergebnis der Prüfung ist eine XML-Datei im SVRL-Format. Sie bildet die Ausgangsbasis für eine mögliche dritte Phase, in der sie entsprechend bestehender Reporting-Anforderungen in ein besser lesbares Format überführt wird. SVRL ist als Standard-Ausgabeformat für ISO-Schematron fester Bestandteil der Spezifikation.

Abhängig vom Funktionsumfang der Implementierung ist jedoch die Möglich-

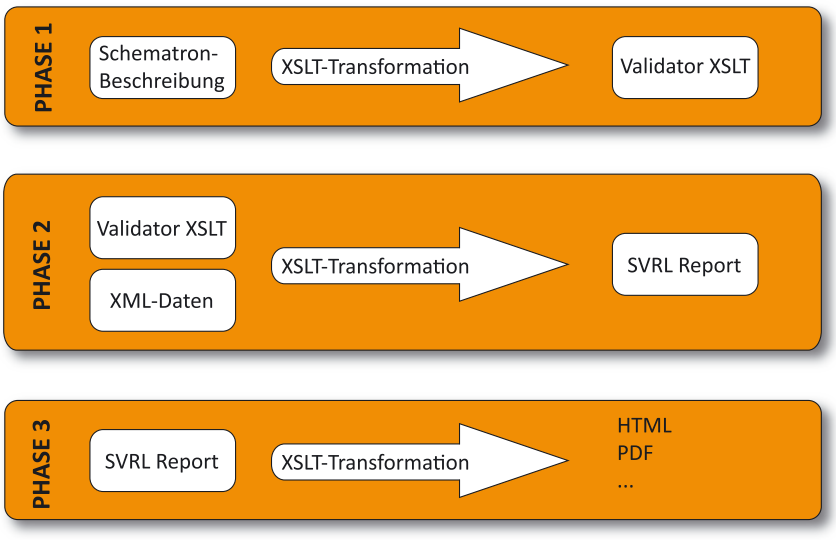

Abbildung 1: Ablauf der Schematron-Prüfung

Unsere Inserenten

aformatik Training und Consulting GmbH & Co. KG
www.aformatik.de
Seite 43

BASIS Europe Distribution GmbH
www.basis-europe.eu
U 4

CaptainCasa Gmbh
www.CaptainCasa.com
Seite 23

DOAG e.V.
www.doag2011.org
U 3

esentri consulting GmbH
www.esentri.de
Seite 35, U 2

Sun User Group Deutschland e.V.
www.sourcetalk.de
Seite 3

keit gegeben, in Phase 2 das Prüfergebnis bereits in einem anderen Format ausgeben zu lassen. Wendet man das Schematron aus Listing 1 auf die XML-Datei aus Listing 2 an, entspricht das in SVRL formulierte Ergebnis dem aus Listing 3. Neben den bei der Prüfung angewandten Regeln finden wir auch eine fehlgeschlagene Zusicherung und die dazugehörende verbale Beschreibung.

Fazit

Obwohl dieser Artikel nur einen Teil der Mächtigkeit von Schematron beleuchtet, kann die vorgestellte Teilmenge schon in vielen Situationen ein hilfreicher Begleiter bei der Erstellung und Umsetzung XML-basierter Schnittstellen sein. Der zunächst entstehende Mehraufwand bei Erstellung und Pflege des zusätzlichen Dokuments ist sicher nicht zu leugnen. Er wird sich in den allermeisten Projekten jedoch sehr schnell bezahlt machen – gerade wenn es darum geht, eigene Systeme für eine Vielzahl unterschiedlicher Partner zu öffnen.

Quellen

http://xml.ascc.net/schematron/
http://www.schematron.com/
http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Kontakt:
Tobias Israel
tobias.israel@buschmais.com


```
<?xml version="1.0" encoding="utf-8" ?>
<fileset date="2010/11/08">
   <file name="IssueManagementService.xml">
      <svrl:schematron-output
            xmlns:iso="http://purl.oclc.org/dsdl/schematron"
            xmlns:sch="http://www.ascc.net/xml/schematron"
            xmlns:svrl="http://purl.oclc.org/dsdl/svrl"
            xmlns:schold="http://www.ascc.net/xml/schematron"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:t="http://example.com/IssueMngtService/1.0"
            title="Issue management service interface V1.00.0
                   integrity constraints"
            schemaVersion="">
         <svrl:ns-prefix-in-attribute-values
               uri="http://example.com/IssueMngtService/1.0"
               prefix="t"/>
         <svrl:active-pattern/>
         <svrl:fired-rule
            context="/t:CreateIssueRequest/t:ServiceLevel"/>
         <svrl:failed-assert
      test="(t:ServiceLevel='EXPRESS') and
          (count(t:Issue) = 1)"
      location="/*[local-name()='CreateIssueRequest'
        and namespace-uri()='http://example.com/
        IssueMngtService/1.0']/*[local-name()=
        'ServiceLevel' and namespace-uri()=
        'http://example.com/IssueMngtService/1.0']">
      <svrl:text>
        Only one issue per request is permitted for
        service level EXPRESS.
      </svrl:text>
    </svrl:failed-assert>
    <svrl:fired-rule context="/t:CreateIssueRequest/
                  t:Issue/t:Detail"/>
    <svrl:fired-rule context="/t:CreateIssueRequest/
                  t:Issue/t:Detail"/>
   </svrl:schematron-output>
  </file>
</fileset>
```

Listing 3: result.xml

Schematron 1.0	Erste Version entwickelt am Academia Sinica Computing Centre, Taiwan
Schematron 1.3 Schematron 1.5 Schematron 1.6	Weiterentwicklungen der Spezifikation; Umsetzung in zahlreichen Implementierungen für unterschiedliche Plattformen
ISO Schematron	Aufbauend auf Version 1.6 offizieller ISO/IEC-Standard (19757-3); enthält das standardisierte SVRL-Ausgabenformat

Tabelle 1: Versionsübersicht Schematron

Tobias Israel arbeitet als IT-Berater auf dem Gebiet der Java-Enterprise-Technologien. Ein Schwerpunkt seiner Arbeit liegt auf der Erstellung und Umsetzung von Integrationsstrategien für verteilte Unternehmensanwendungen. Er ist Mitinhaber der buschmais GbR – einem Beratungshaus mit Sitz in Dresden.

Bestellen eines kostenlosen Exemplares der Zeitschrift Java aktuell

Anschrift:

Name, Vorname

Firma

Abteilung

Straße, Hausnummer

PLZ, Ort

ggf. Rechnungsanschrift

E-Mail

Telefonnummer

Die allgemeinen Geschäftsbedingungen* erkenne ich an, Datum, Unterschrift

Jetzt Abonnement sichern:

- O Abonnement Newsletter: Java aktuell – der iJUG-Newsletter, kostenfrei
- O Java aktuell – das iJUG-Magazin Abo: vier Ausgaben zu 18 Euro im Jahr

Für Oracle-Anwender und Interessierte gibt es das Java aktuell Abonnement auch mit zusätzlich sechs Ausgaben im Jahr der Fachzeitschrift DOAG *News* und zwei Ausgaben im Jahr Business News. Weitere Informationen unter www.doag.org/shop/

Senden Sie das ausgefüllte Formular an:

Interessenverbund der Java User Groups e.V.
Tempelhofer Weg 64
12347 Berlin

oder faxen Sie es an:
0700 11 36 24 39

oder bestellen Sie online:
go.ijug.eu/go/abo

*Allgemeine Geschäftsbedingungen:
Zum Preis von 18 Euro (inkl. MwSt.) pro Kalenderjahr erhalten Sie vier Ausgaben der Zeitschrift "Java aktuell - das iJUG-Magazin" direkt nach Erscheinen per Post zugeschickt. Die Abonnementgebühr wird jeweils im Januar für ein Jahr fällig. Sie erhalten eine entsprechende Rechung. Abonnementverträge, die während eines Jahres beginnen, werden mit 4,90 Euro (inkl. MwSt.) je volles Quartal berechnet. Das Abonnement verlängert sich automatisch um ein weiteres Jahr, wenn es nicht bis zum 31. Oktober eines Jahres schriftlich gekündigt wird. Die Wiederrufsfrist beträgt 14 Tage ab Vertragserklärung in Textform ohne Angabe von Gründen.

Impressum

Herausgeber:
Interessenverbund der Java User Groups e.V. (iJUG)
Tempelhofer Weg 64, 12347 Berlin
Tel.: 0700 11 36 24 38
www.ijug.eu

Verlag:
DOAG Dienstleistungen GmbH
Fried Saacke, Geschäftsführer
info@doag-dienstleistungen.de

Chefredakteur (VisdP):
Wolfgang Taschner,
redaktion@ijug.eu

Chefin von Dienst (CvD):
Carmen Al-Youssef,
office@ijug.eu

Titel, Gestaltung und Satz:
Claudia Wagner,
DOAG Dienstleistungen GmbH

Anzeigen:
CrossMarkeTeam, Ralf Rutkat,
Doris Budwill
redaktion@ijug.eu

Mediadaten und Preise:
http://www.ijug.eu/images/vorlagen/2011-ijug-mediadaten_java_aktuell.pdf

Druck:
adame Advertising and Media GmbH Berlin
www.adame.de

Java aktuell – das Abo
4 Ausgaben für 18 Euro