S&T06 Tutorial

Rechnen, Runden, Rekursion – das numerische Potenzial von XSLT

Dr. Thomas Meinike, Hochschule Merseburg

Motivation

Die Transformationssprache XSLT wird kurz nach der Tagung 20 Jahre alt und ist der Inbegriff für die Verarbeitung von XML-Dokumenten in unterschiedliche Ausgabeformate. Seit der Version 1.0 sind neben typischen Zugriffen auf Elementinhalte und Attributwerte auch Rechen- und Logikoperationen auf diesen Daten möglich. Was relativ zaghaft mit den Grundrechenarten und rekursiven Template-Aufrufen begann, wurde über die Versionen 2.0 und 3.0 stark erweitert. Mittlerweile entsprechen die numerischen Möglichkeiten denen anderer Sprachen. Dieses Tutorial soll die Bandbreite verfügbarer Techniken und ihre Evolution anhand von Beispielen vermitteln.

Kurze Versionsgeschichte

XSLT ist neben der Abfragesprache XPath und der Layout-Technologie XSL-FO der wesentliche Teil der XSL-Familie [1]. Im November 1999 erschien die Version 1.0 [2] mit grundständigen Techniken primär für den XML-Zugriff. In Form von einigen XPath-Funktionen [3] waren bereits Operationen für Zeichenketten und Zahlenwerte implementiert. Die numerischen Möglichkeiten waren noch auf die Grundrechenarten beschränkt, wobei sich durch die Nutzung benannter Templates mit Parametern durchaus komplexere Algorithmen umsetzen ließen. Höhere mathematische Funktionen waren noch nicht im Sprachumfang enthalten. Immerhin konnten Rundungen und Summationen über Element- bzw. Attributknoten sowie Abzählungen, etwa des Vorkommens bestimmter Objekte im XML-Baum, vorgenommen werden. Mit entsprechendem Sachverstand und Geschick war XSLT 1.0 für vielfältige Aufgaben nutzbar und ist heute noch die Version der Wahl, wenn keine Softwareunterstützung der erweiterten Standards gegeben ist.

Die im Januar 2007 erschienenen Versionen von XSLT 2.0 [4] und XPath 2.0 [5] haben jedoch eine Vielzahl an Neuerungen in den Sprachumfang eingebracht. Eine der interessantesten neuen Techniken ist xsl:function, die ihrem Namen nach benutzerdefinierte Funktionen ermöglicht, die deutlich weniger Aufwand gegenüber Ansätzen mit xsl:call-template erfordern und sich zudem wie vorgefertigte XPath-Funktionen in Abfragen verwenden lassen. Auf dieser Basis wurden vom Autor zeitnah zur Veröffentlichung der 2.0-Spezifikationen u. a. Implementierungen von Winkelfunktionen wie Sinus und Kosinus entwickelt. Was zunächst exotisch klingen mag, wird schnell plausibel, wenn Tortengrafiken mit dem XML-basierten Vektorgrafikstandard SVG realisiert werden sollen.

Ein wichtiger Schlüssel zu mehr Programmatik liegt in den neu eingeführten Sequenzen, die den praktikablen Zugriff auf Teilstrukturen von Dokumenten oder Listen von Werten ermöglichen. Schließlich haben die letzten Standardisierungen von XSLT 3.0 [6] und XPath 3.1 [7] im Jahre 2017 mit der Einführung von Arrays und Maps und einer Reihe vorgefertigter mathematischer Funktionen weiteren Entwicklungskomfort geschaffen. Auch die „Higher-Order Functions“ (HOF) sind zu nennen, deren Nutzung allerdings den kommerziellen Varianten des Saxon-Prozessors PE bzw. EE vorbehalten ist. Die freie Implementierung Saxon-HE ermöglicht jedoch die Nutzung der meisten hier einbezogenen Techniken [8].

Ab XSLT 2.0 stehen zudem aus XQuery entlehnte Abfragetechniken wie if-then-else und for-in-return zur Verfügung, die XPath-Abfragen noch mächtiger machen. Auch das Angebot von Datentypen über String und Number hinaus hilft beim Umgang mit entsprechend typisierten Inhalten. Dafür wurden die aus XML-Schema bekannten Zuordnungen wie xs:integer, xs:decimal, xs:date usw. integriert.

Themenbereiche des Tutorials

Das Tutorial liefert zu den nachfolgend genannten Teilaspekten jeweils Problemstellungen mit Lösungsansätzen anhand von ausführbaren und zur Verfügung gestellten Codefragmenten.

Template-Mechanismus

Eine kurze Einführung oder Wiederholung zeigt die Ansätze zur (rekursiven) Nutzung von Templates für die funktionale Programmierung auf. Hier sind vor allem die Techniken xsl:template / xsl:apply-templates und xsl:call-template / xsl:with-param / xsl:param zu betrachten.

Variablenkonzept

Variablen sind im Grunde Konstanten, d. h. nach Deklaration nicht mehr änderbar, jedoch zyklisch im Kontext von Templates oder xsl:for-each-Konstrukten neu erzeugbar. Variablen und Parameter sind am vorangestellten Dollarzeichen identifizierbar ($name). Auf oberster Ebene lassen sich globale Variablen und Parameter erzeugen, ansonsten sind diese auf ihre lokale Umgebung beschränkt. Ab 2.0 können Datentypen über das as-Attribut zugewiesen werden. Neben Einzelwerten sind auch Sequenzen von Werten oder XML-Teilstrukturen adressierbar.

Grundrechenarten

Addition (+), Subtraktion (–), Multiplikation (*) und Division (div, da der sonst übliche Schrägstrich bereits für den XML-Baumzugriff wie a/b/c verwendet wird) bieten die Grundlage numerischer Operationen. Der Rest einer ganzzahligen Division ist über mod zugänglich (10 mod 3 → 1).

Logiktests

In XPath-Ausdrücken können logische Verknüpfungen mittels and, or und not() hergestellt werden.

Runden

Ganzzahliges (Auf-/Ab-)Runden ist mit den Funktionen ceiling(), floor() und round() möglich. Ab 2.0 steht round-half-to-even() mit einem Parameter für die Anzahl der Nachkommastellen bereit.

Aggregatfunktionen

Neben Summierung und Abzählung mit sum() und count() können ab 2.0 avg(), min() und max() zur Bestimmung von Mittel-, Minimal- und Maximalwerten auf einer Knoten- oder Wertesequenz verwendet werden.

Zahlenformatierung

Zahlenwerte lassen sich mit dem Element xsl:number und der Funktion format-number() hinsichtlich bestimmter Muster wie Stellen vor bzw. nach einem Komma formatieren.

Umgang mit Datum und Zeit

Abfragen von Datums- und Zeitwerten sind ab 2.0 u. a. mit current-date(), current-time() und current-dateTime() möglich. Es können zusätzliche Berechnungen angestellt werden, etwa die Ermittlung eines Datums in x Tagen über Monats- oder Jahresgrenzen hinaus.

Sortieren, Gruppieren und Filtern

Sortieroperationen mit xsl:sort gibt es von Anfang an. Zum schmerzfreien Gruppieren hat XSLT 2.0 xsl:for-each-group eingeführt. Datenfilterung ist mit Techniken wie distinct-values() möglich.

Reguläre Ausdrücke

Eine verbesserte Methodik zur Verarbeitung von Zeichenketten in Mengentexten liefern reguläre Ausdrücke. Hierzu existieren ab 2.0 XSLT- und XPath-Varianten.

Erweiterte Inline-Abfragen

Die genannten if-then-else- und for-in-return-Konstrukte ermöglichen gleichermaßen komplexe wie kompakte inzeilige Abfragen.

Benutzerdefinierte Funktionen

Die genannte 2.0-Technik xsl:function lässt sich relativ schnell erschließen. Ein selbst gewählter Funktionsname wird jeweils über einen Namensraum wie beispielsweise ns:funktionsname adressiert. Mit xsl:param lassen sich die zu verwendenden Parameter deklarieren.

Mathematische Funktionen

XSLT 3.0 / XPath 3.1 stellen über den math-Namensraum eine Reihe vorgefertigter mathematischer Funktionen bereit, darunter Exponential-, Logarithmus-, Wurzel- und Winkelfunktionen.

Funktionen höherer Ordnung

Diese erweitern das Methodenspektrum zusätzlich. Über verschachtelte Anweisungen mit function() {...} sind je nach verfügbarer Prozessorvariante spezialisierte Abfragen möglich.

Array und Maps

XPath 3.1 bietet die Techniken Arrays und Maps an. Neben Funktionen aus dem Portfolio von array:* und map:* ist auch die Unterstützung von JSON-formatierten Datenstrukturen relevant.

Zufallszahlen

Die Erzeugung und weitere Verarbeitung von Zufallszahlen ermöglicht XPath 3.1 mit dem Zugriff auf random-number-generator().

Fazit und Ausblick

XSLT und XPath haben über 20 Jahre vielfältige Detailverbesserungen erfahren. Neben der klassischen XML-Verarbeitung steht bei Ausnutzung neuerer Techniken einer stärker datenorientierten

Anwendung nichts mehr im Weg. Unter [9–13] ist unterstützendes Material des Autors zu finden. Einige enthaltene und weitere Beispiele werden im Tutorial thematisiert.

Literaturangaben und Links

[1] W3C: The Extensible Stylesheet Language Family (XSL). http://www.w3.org/Style/XSL/

[2] W3C: XSL Transformations (XSLT) Version 1.0. http://www.w3.org/TR/xslt-10/

[3] W3C: XML Path Language (XPath) Version 1.0. http://www.w3.org/TR/xpath-10/

[4] W3C: XSL Transformations (XSLT) Version 2.0. http://www.w3.org/TR/xslt20/

[5] W3C: XML Path Language (XPath) 2.0 (Second Edition). http://www.w3.org/TR/xpath20/

[6] W3C: XSL Transformations (XSLT) Version 3.0. http://www.w3.org/TR/xslt-30/

[7] W3C: XML Path Language (XPath) 3.1. http://www.w3.org/TR/xpath-31/

[8] Saxonica: Why choose Saxon? http://saxonica.com/html/products/why-saxon.html

[9] Meinike, T.: XSLT 2.0 und XPath 2.0 für Praktiker – Neuerungen im Überblick. In: Gesellschaft für Technische Kommunikation – tekom Deutschland e.V., Tagungsband zur Jahrestagung 2007, S. 212–215. https://datenverdrahten.de/PDF/tekom2007_Wiesbaden_Meinike.pdf

[10] Meinike, T.: Tutorial: XSLT-Programmierung – effektiv und schmerzfrei! In: Gesellschaft für Technische Kommunikation – tekom Deutschland e.V., Tagungsband zur Jahrestagung 2011, S. 313–315. https://datenverdrahten.de/PDF/tekom2011_OTS11_Meinike.pdf

[11] Meinike, T.: 3.0-Updates von XSLT und XPath auf einen Blick. In: Gesellschaft für Technische Kommunikation – tekom Deutschland e.V., Tagungsband zur Jahrestagung 2012, S. 341–343. https://datenverdrahten.de/PDF/tekom2012_OTS3_Meinike.pdf

[12] Meinike, T.: Interaktive XML-Verarbeitung im Browser mit Saxon-JS und XSLT 3.0. In: Gesellschaft für Technische Kommunikation – tekom Deutschland e.V., Tagungsband zur Jahrestagung 2017, S. 177–180. https://datenverdrahten.de/PDF/tekom2017_IN29_Meinike.pdf

[13] Meinike, T.: Verarbeitung von JSON-Daten mit aktuellen XSLT-Techniken. In: Gesellschaft für Technische Kommunikation – tekom Deutschland e.V., Tagungsband zur Jahrestagung 2018, S. 123–126. https://datenverdrahten.de/PDF/tekom2018_IN22_Meinike.pdf

Kontakt:
thomas.meinike@hs-merseburg.de

(: Verfasst bei https://soundcloud.com/tschunkelmusik – das ist übrigens ein XPath-Kommentar. :)