

Entwicklung von dynamischen Datenzugriffstechniken für Scalable Vector Graphics

Projekt zur Programmierung von XML-basierten Client-Server-Anwendungen

von Dr. rer. nat. Thomas Meinike / Studiengang Kommunikation und Technische Dokumentation (KTD)
Geusaer Straße, 06217 Merseburg, Tel.: 03461/463055, E-Mail: thomas.meinike@et.fh-merseburg.de

Projektziel:

Im September 2001 wurde vom W3C die Spezifikation für Scalable Vector Graphics (SVG) in der Version 1.0 offiziell verabschiedet, Version 1.1 folgte im Januar 2003 [1]. Typische Anwendungsgebiete sind gegenwärtig vor allem Kartografie, Mess- und Prozessdatenverarbeitung, CAD und Technische Illustration.

Besonders interessant ist die Kopplung von SVG-Dokumenten mit online verfügbaren Datenquellen. Die hier dargestellte Anwendung soll den Zugriff auf einen XML-Webservice aus einem SVG-Dokument heraus demonstrieren und die verwendeten Programmierstechniken aufzeigen.

Überblick:

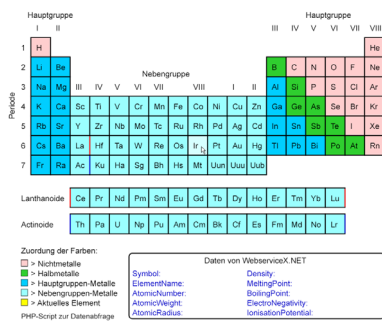
SVG-Dokumente sind XML-Instanzen und können über das Document Object Model (DOM) ereignisgesteuert dynamisch verändert werden. Dazu lassen sich u. a. in JavaScript programmierte Routinen verwenden. Elemente und Attribute der Grafiken sind mittels DOM-Methoden beeinflussbar. So kann man z. B. auf einen Mausklick reagieren und der Grafik neue Elemente hinzufügen oder bestehende verändern bzw. aus dem Dokumentenbaum entfernen.

Außerdem können externe Datenquellen über eine URL-Referenz an ein SVG-Dokument gebunden werden. Die benötigten Techniken werden in [2] ausführlich beschrieben.

XML Web Services erweisen sich als eine moderne Form der Datenkommunikation. Im World Wide Web existieren bereits Dienste, die Börseninformationen oder aktuelle Wetterdaten zur Verfügung stellen. Für die Beispielanwendung wurde der Web Service „Periodic Table“ ausgewählt [3]. Darüber lassen sich physikalische Daten der chemischen Elemente abrufen.

Praktische Umsetzung:

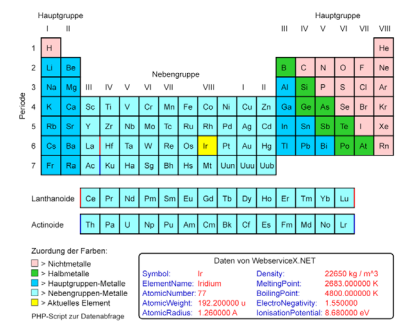
Zunächst wurde eine SVG-Repräsentation des Periodensystems der chemischen Elemente realisiert. Beim Anklicken der den Elementen zugeordneten quadratischen Boxen wird jeweils eine JavaScript-Funktion aufgerufen, die mit dem genannten Web Service Kontakt aufnimmt. Dazu dient ein auf dem Webserver befindliches PHP-Skript, welches die XML-Daten abholt und an das SVG-Dokument zurück gibt. Die erhaltenen Informationen werden über das Objektmodell als Textknoteninhalte ausgegeben.



SVG-Dokument zur Darstellung des Periodensystems

```
<?xml version="1.0" encoding="utf-8"?>
<NewDataSet>
  <Table>
    <AtomicNumber>77</AtomicNumber>
    <ElementName>Iridium</ElementName>
    <Symbol>Ir</Symbol>
    <AtomicWeight>192.2207</AtomicWeight>
    <BoilingPoint>4880</BoilingPoint>
    <IonisationPotential>8.66</IonisationPotential>
    <ElectroNegativity>1.55</ElectroNegativity>
    <AtomicRadius>1.26</AtomicRadius>
    <MeltingPoint>2683</MeltingPoint>
    <Density>22650</Density>
  </Table>
</NewDataSet>
```

vom Web Service gelieferter XML-Code für Iridium



Ausgabe der Daten für Iridium im SVG-Dokument

Fazit:

Scalable Vector Graphics bieten vielfältige Einsatzmöglichkeiten. Durch dynamische Datenanbindung werden diese noch wesentlich erweitert. Auf der Basis der verwendeten Techniken lassen sich neue Potenziale für interaktive XML-Anwendungen erschließen.

Die Beispielanwendung steht online zur Verfügung [4]. Weitere Details werden in [5] behandelt.

Referenzen:

- [1] W3C: <http://www.w3.org/Graphics/SVG/>
- [2] T. Meinike: SVG ruft Datenbank, XML & Web Services Magazin 3.03, S. 66-69.
- [3] Web Service „Periodic Table“: <http://www.webservice.net/WS/WSDetails.aspx?WSID=19&CATID=7>
- [4] Beispielanwendung online: <http://www.et.fh-merseburg.de/person/meinike/ptablesvg/>
- [5] T. Meinike: Chemiestunde online, XML & Web Services Magazin (geplant für Ausgabe 1.04).

