

# Graphen in Apex

**Thomas Hernando Gotthardt**  
syntegris information solutions GmbH  
Neu-Isenburg

## Schlüsselworte

Apex, Oracle, PL/SQL, GraphViz, JavaScript, d3.js, jQuery, Graphen, Bäume, DWH

## Einleitung

In vielen Situationen werden Graphen als Daten- oder Darstellungsstruktur benötigt: der klassische Stammbaum, das Navigationssystem oder Datenflüsse in Datawarehouse Umgebungen stellen nur eine kleine Auswahl an Anwendungsmöglichkeiten dar.

Da es in Oracle Application Express derzeit keine Möglichkeit gibt, Graphen per Standard zu erstellen, müssen andere Technologien herangezogen werden.

Im Vortrag sollen aus den zwei grundlegenden Architekturen (Serverseitige oder Clientseitige Generierung) Möglichkeiten aufgezeigt werden, wie sich Graphen in APEX für den Benutzer darstellen lassen. Es werden die Vor- und Nachteile der Nutzung von GraphViz und d3.js aufgezeigt und beide Methoden während des Vortrags live demonstriert.

## Graphen in Apex

In Apex gibt es derzeit lediglich die Möglichkeit Bäume als spezielle Graphen in Form von Navigationselementen darzustellen. Diese sind in ihren Standard Möglichkeiten begrenzt:

- Bilder als Knoten
- Links an Knoten
- Tooltip's für Knoten

Diese Bäume eignen sich gut für die Nutzung als Navigation, allerdings ist die grafische Aufbereitung in Form eines Baums oder gar Graphen, wie in Abbildung 1, nicht möglich. In Abb. 1 sehen Sie einen Graphen, der nicht durch Standardmitteln in Apex erstellt werden kann.

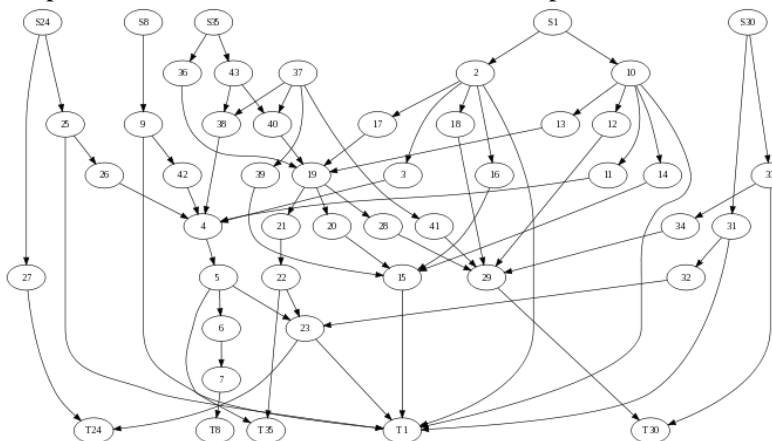


Abbildung 1 - Graph erstellt mit GraphViz (Quelle: [www.graphviz.org](http://www.graphviz.org))

In der Einleitung sind bereits einige Anwendungsbeispiele genannt worden. Die Anwendungsbreite ist allerdings noch wesentlich höher und erstreckt sich über vielfältige Bereiche: IT, Physik, Chemie, BWL usw. In der IT werden bspw. in Netzwerküberwachungstools Graphen dargestellt, die das (gesamte) Netzwerk zeigen. Die grundlegenden Daten dazu liegen meist in einer Datenbank und werden mit extra Tools selektiert und in Graphen dargestellt. Dies ist auch in Apex möglich. Hier gibt es zwei grundlegenden Möglichkeiten der Graph Erstellung: **Serverseitige und Clientseitige Erstellung**

### Serverseitige Erstellung

Bei der Serverseitigen Erstellung, wird ein Code (z.B. XML, JSON, dot usw.) an ein Server-Programm gesendet, welches als Ergebnis einen Graphen zurückliefert (siehe Abbildung 2).

Bei dieser Methode ist darauf zu achten, dass es notwendig ist ein zusätzliches Tool auf einem Server zu installieren. Unsere Wahl fiel auf das von AT&T entwickelte "GraphViz". Das Tool wurde auf der Datenbankmaschine installiert und kann per Kommandozeile verwendet werden.

GraphViz bietet neben der manuellen Erstellung von Graphen auch die Möglichkeit über die Kommandozeile das Tool zu bedienen. Außerdem kann es verschiedene Outputs generieren, angefangen bei png, jpg bis hin zu pdf und vml unterstützt GraphViz einige Ausgabeformate. Das Programm selbst unterstützt zusätzlich verschiedene Layout Engines, die es erlauben unterschiedliche Arten von Graphen zu erstellen. Im Vortrag soll derweil die dot-Engine mit dem gleichnamigen dot-Code gezeigt werden. Mithilfe des dot-Codes kann ein (gerichteter) Graph beschrieben werden. In Abbildung 2 ist der Code und der daraus generierte Output gezeigt.

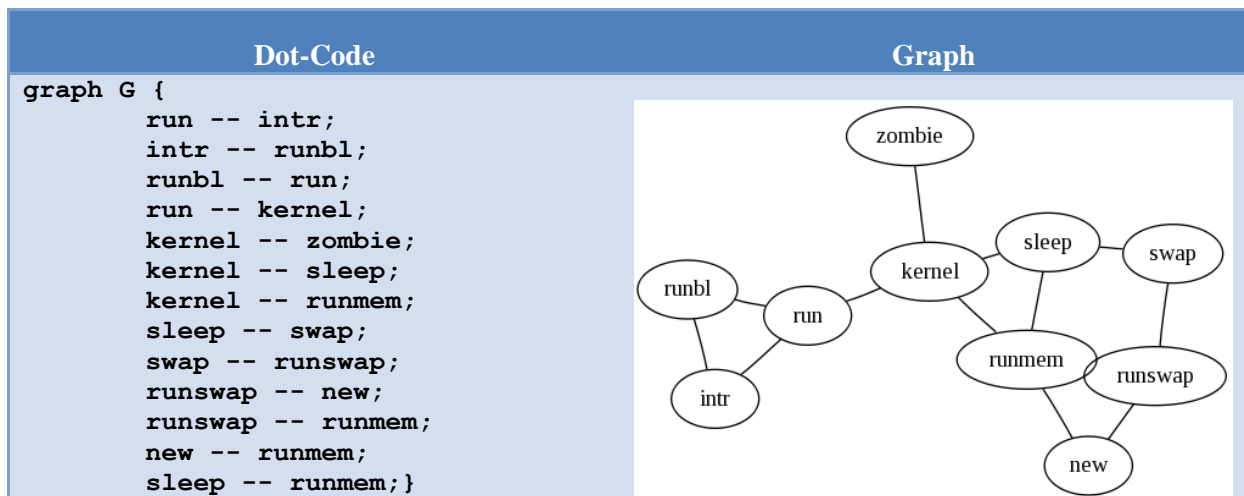


Abbildung 2 - Graph und Code (Quelle: [www.graphviz.org](http://www.graphviz.org))

Innerhalb von Apex wird nun zur Generierung des dot-Codes ein PL/SQL Package/Funktion erstellt, welche(s) den dot-Code aus vorhandenen hierarchischen Daten erstellt. Je nach Größe der selektierten Daten muss vorher geprüft werden ob ein VARCHAR oder ein CLOB benötigt wird. Der generierte Code wird nun als \*.dot Datei auf das Dateisystem der DB-Maschine geschrieben. Um Dateien aufs Dateisystem zu schreiben, können Sie das UTL\_FILE Package benutzen. Als Speicherort für die erzeugten Dateien (Graphen und Ausgangsdaten) bietet sich das Image Verzeichnis der Apex-Installation an, sofern es auf der Datenbank-Maschine liegt. In der Oracle Datenbank wird zunächst ein Directory benötigt, welches auf das neu erzeugte Verzeichnis im Apex-Image-Verzeichnis zeigt.

Die Datei wird in das Verzeichnis geschrieben. Durch eine External Table und der Preprocessor Option kann per SQL ein Systemcall abgesetzt werden, der die dot-Engine startet. Dazu muss auf der Unix Maschine ein Shell-Skript erzeugt werden, welches folgendes Kommando an die dot-Engine schickt:

```
>dot -Tpng demo.dot -o demo.png
```

Das Attribut -T erhält das Ausgabeformat gefolgt von der Code-Datei und im Anschluss das Attribut -o erwartet die Ausgabedatei. Die Engine generiert aus dem empfangenen dot-Code ein Bild und schreibt dieses in das Verzeichnis, auf welches die External Table zeigt und in dem auch der zuvor generierte dot-Code abgelegt wurde. Um das erzeugte Bild anzuzeigen, genügt es per #IMAGE\_PREFIX# ein Display Image Item auf der Apex-Seite mit dem Bild zu befüllen.

### Clientseitige Erstellung

Bei der clientseitigen Erstellung eines Graphen, wird auf dem Server lediglich der Export in Form von XML, JSON oder einer anderen Dateistruktur durchgeführt. Dieser Code wird an den Client gesendet. Der Client muss anschließend den Code nur noch interpretieren.

Die geläufigste Variante um im Browser grafische Arbeiten zu erledigen ist JavaScript. Nun gibt es verschiedene Frameworks im JavaScript Bereich die dem Entwickler die Arbeit erleichtern. In Apex wird seit geraumer Zeit jQuery genutzt, aber auch Frameworks wie das TwitterBootstrap oder AngularJS können zum Einsatz kommen.

Um grafische Arbeiten sehr zu vereinfachen gibt es unter anderem das Framework d3.js (d3 = Data Driven Development). Mithilfe dieses Frameworks können bspw. folgende visuellen Aufbereitungen erstellt werden:

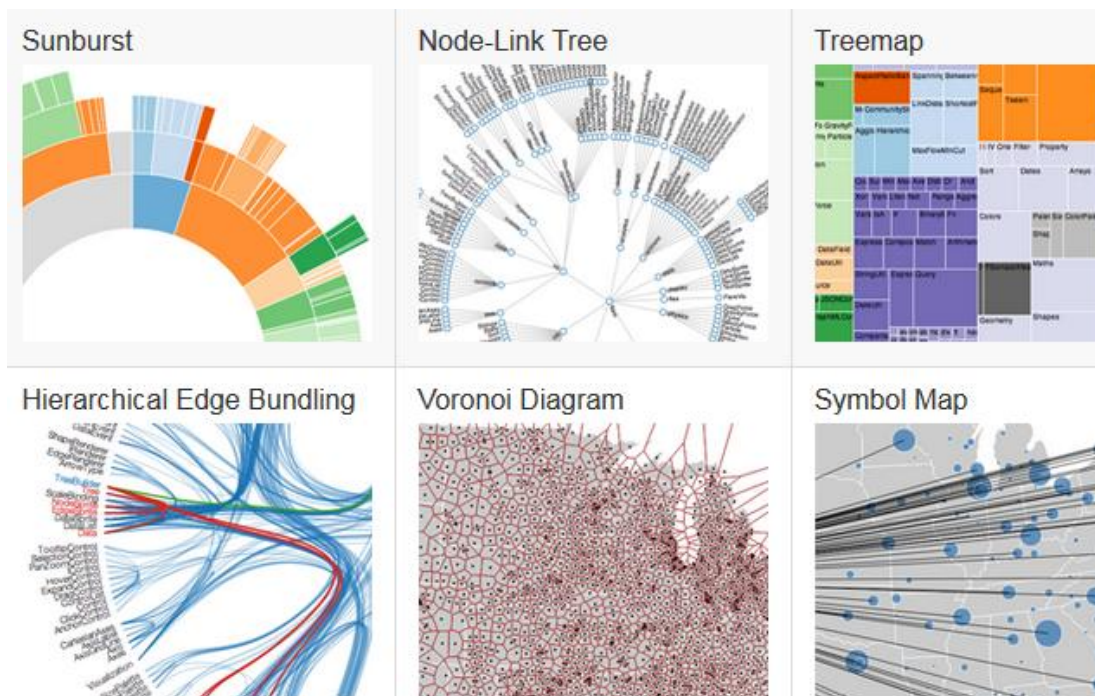


Abbildung 3 Graphen generiert von d3.js (Quelle: [www.d3js.org](http://www.d3js.org))

Um erfolgreich einen Graphen in D3.js zu erstellen, wird zunächst eine DB-Funktion benötigt, die JSON, XML oder eine CSV-Datei bereitstellt. Im Vortrag wird unter anderem mit JSON Daten gearbeitet, da dies in der Verarbeitung mit JavaScript sehr einfach von der Hand geht. Das Besondere an D3.js ist allerdings die Möglichkeit, Daten aus unterschiedlichen Quellen und mit unterschiedlicher Struktur, mit dem gleichen Programmcode zu verarbeiten. Um den in Abbildung 4 gezeigten Graphen zu erzeugen, kann bspw. JSON Code verwendet werden.

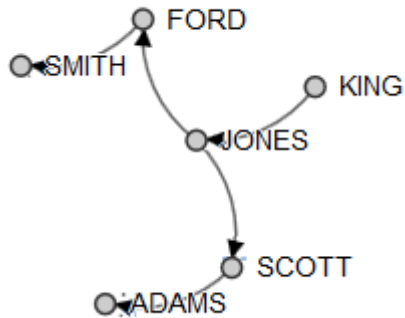


Abbildung 4 Graph generiert durch d3.js in Apex Anwendung

Die Möglichkeit verschiedene Datenstrukturen mit dem gleichen Programmcode zu verarbeiten, wird als data-driven Development bezeichnet.

Um Daten an den JavaScript-Code zu übergeben bieten sich Application Processes an. Diese können per Ajax-Call aufgerufen werden und liefern direkt die Daten an die d3.js-Engine. Dort können die Daten ganz individuell weiterverarbeitet werden. Durch einfache Schnittstellen ist es möglich in kurzer Zeit komplexe grafische Anforderungen umzusetzen. Das Framework kann neben normalem HTML, SVG-Grafiken aufbauen oder direkt in ein HTML5-Canvas Element zeichnen. Je nach Anwendungsgebiet ist ein anderes Ausgabeformat zu wählen. Im Vortrag werden SVG-Grafiken erstellt, da diese als normaler HTML-Inhalt später auf der Seite vorhanden ist und sich dadurch anbietet weitere Aktionen mit den generierten Elementen durchzuführen. Im Vortrag wird hauptsächlich mit SVG-Grafiken gearbeitet.

Zum Ende des Vortrags wird der Referent ein Fazit über die beiden vorgestellten Möglichkeiten geben und die Vor- sowie Nachteile präsentieren.

**Kontaktadresse:**

Thomas Hernando Gotthardt  
syntegris information solutions GmbH  
Hermannstr. 54-56  
63263 Neu-Isenburg

Telefon: +49 (0) 61 02 - 29 86 68  
Fax: +49 (0) 61 02 - 55 88 06  
E-Mail [thomas.hernando@syntegris.de](mailto:thomas.hernando@syntegris.de)  
Internet: [www.syntegris.de](http://www.syntegris.de)