

SVG 2 – ein (Er-)Wartungsbericht

Dr. Thomas Meinike, Hochschule Merseburg

Einführung

Das XML-basierte Vektorgrafikformat SVG, kurz für Scalable Vector Graphics, wurde ursprünglich Ende der 1990er Jahre von der Firma Adobe konzipiert und dem W3C zur Standardisierung und Weiterentwicklung übergeben. 2001 erschien die Version 1.0 mit den auch heute noch grundsätzlich gültigen Definitionen zur Umsetzung von (interaktiven) Grafiken. Verfeinert wurde das Konzept 2003 in Form der modular aufgebauten Version 1.1 (Profile Tiny, Basic und Full), wobei die Full-Spezifikation den Umfang der bisherigen Version 1.0 verkörperte und die zusätzlichen Profile für mobile, zu dieser Zeit noch nicht besonders leistungsfähige Endgeräte gedacht waren [1]. Die Gesamtspezifikation 1.1 wurde 2011 nochmals in fehlerbereinigter Form aufgelegt [2].

Zwischenzeitlich wurde 2008 die Spezifikation SVG 1.2 Tiny veröffentlicht, ebenfalls für den mobilen Grafikeinsatz ausgelegt [3]. Eine Überarbeitung der übergeordneten Spezifikation blieb jedoch lange aus. Erst um 2012 fand die zuständige Arbeitsgruppe [4] neu zusammen und orientierte sich auf SVG 2.0, mittlerweile als SVG 2 bezeichnet. Über diesen Zeitraum sind bis Anfang September 2014 vier Working Drafts entstanden [5], während der Editor's Draft [6] in kürzerer Folge den aktuell erreichten Stand dokumentiert. Seit August 2014 steht das Material auch auf der Plattform GitHub zur Verfügung [7].

Motivation und Methodik

Das Thema SVG wird vom Autor seit den Anfängen in Form von Artikeln, Vorträgen und Beispielcode begleitet [8, 9]. Bereits auf der Jahrestagung 2004 wurde über die Perspektiven einer Version 1.2 berichtet [10], die sich jedoch nur marginal manifestierten. Die genannten Aktivitäten der Arbeitsgruppe lassen in absehbarer Zeit konkrete Weiterentwicklungen erwarten, die auch für die Technische Kommunikation von Interesse sind. In diesem Bereich hat sich SVG u. a. für Darstellungen in Ersatzteilkatalogen, Datenvisualisierungen und Webanwendungen gut etabliert.

Die zugrunde liegenden Ideen und Entwürfe sind frei zugänglich dokumentiert, einerseits über die bereits genannten Quellen [5] und [6] sowie andererseits über weitere Kanäle wie Protokolle von Telekonferenzen oder Mitteilungen von Mitgliedern der Arbeitsgruppe in ihren eigenen Blogs oder über ihre Accounts bei Diensten wie Google+ und Twitter [11–15].

Wesentliche Änderungen und Ergänzungen

Allgemeines

- SVG 2 verzichtet auf die DOCTYPE-Deklaration, es wird keine DTD geben. Konkrete Pläne für alternative Schema-Datenmodelle sind noch nicht bekannt.
- Der Namensraum <http://www.w3.org/2000/svg> bleibt erhalten.
- Bereits existierende und neu entwickelte Spezifikationen werden integriert, u. a. CSS-Transformations, CSS-Shapes, CSS-Masking, Compositing and Blending, Filter Effects.

- Konsolidierung von DOM-Interfaces und Scripting. Das script-Element darf künftig an beliebiger Stelle auftreten, bisher nur im defs-Bereich.
- Einführung der Attribute tabindex zur Fokussierung und z-index für die Objektsteuerung.
- SVG-Attribute für Koordinaten, Längen und Radien sollen auch als CSS-Eigenschaften nutzbar sein.

Bilder

- Bisher benötigten image-Elemente stets Längenangaben in den Attributen height und width, eine automatische Bildanpassung war nicht vorgesehen. Nun kann beiden Attributen alternativ der Wert auto zugewiesen werden.

Farben und Füllungen

- Syntax für HSL und RGBA aus CSS 3 sowie LAB-Farben kommt hinzu.
- Mit dem Element solidColor aus SVG 1.2 Tiny sowie zusätzlichen Attributen und Kindelementen lassen sich eigene benannte, wiederverwendbare Farb- und Fülldefinitionen anlegen.
- Erweiterte Musterdarstellung für schraffierte Flächen mit dem hatch-Element, speziell für technische Zeichnungen geeignet.
- Füllungen lassen sich aus mehreren Komponenten zusammensetzen.

Filtereffekte

- Die vorhandenen Blend-Modi werden an die bereits via CSS verfügbaren Werte angeglichen.

Gradienten

- Radiale Gradienten (Element radialGradient) können über das neue fr-Attribut („focal radius“) genauer beschrieben werden.
- Lineare und radiale Gradienten werden durch Mesh-Gradienten (Elemente meshGradient / meshRow / meshPatch) ergänzt. Damit sollen sich „lebensechte“ fotorealistische Formen und Schattierungen besser umsetzen lassen.

Linien und Umrandungen

- Mit der Eigenschaft vector-effect lassen sich bei Bedarf Größenänderungen von Rahmenlinien bei Zoom-Operationen vermeiden, hilfreich für Explosionszeichnungen.
- Über paint-order kann die Zeichenreihenfolge von Umrandungen, Füllungen und Markern beeinflusst werden.
- Neue Eigenschaften sind stroke-dashcorner und stroke-dashadjust, während stroke-linejoin den zusätzlichen Wert arc erhält.
- Durch zusätzliche Optionen wird die Darstellung von Markern verbessert.

Medien

- Die Elemente audio und video werden gemäß der HTML5-Spezifikation einbezogen.

Pfade

- Im data-Attribut von path-Elementen werden die Kommandos B / b (für „bearing“) ergänzt, die sich zur Konstruktion von Vielecken eignen. Mit R / r lassen sich „Catmull-Rom“-Splinekurven parametrisieren.

Texte

- Ermöglichung von Textfluss über die Attribute `height` und `width` unter Verwendung zusätzlicher Eigenschaften für die Einbettung von Text in Formen.
- Steuerung von Leerraum mit der Eigenschaft `white-space` als Ersatz für `xml:space`.
- Entfernung schriftbezogener Elemente wie `altGlyph`, `altGlyphDef`, `altGlyphItem` und `glyphRef`. Im Gegenzug werden externe WOFF-Schriftarten etabliert (Web Open Font Format).

Fazit und Ausblick

SVG 2 erweitert die bestehenden Möglichkeiten praxisorientiert. Bis zur Tagung fließen vermutlich einige der aufgelisteten Techniken zumindest in die Entwicklerversionen („Nightly Builds“) der Browser Chrome, Opera und Safari bzw. WebKit ein. Da Mitglieder der Arbeitsgruppe bei den Herstellern oder im Open-Source-Umfeld beschäftigt sind, ist nach der finalen Veröffentlichung von einer zeitnahen Verfügbarkeit entsprechender Darstellung- und Erstellungswerkzeuge auszugehen.

Die genannten Schwerpunkte werden im Vortrag aufgegriffen und mit Beispielcode untersetzt.

Literaturangaben und Links

- [1] W3C: SVG Working Group; <http://www.w3.org/Graphics/SVG/>
- [2] W3C: Scalable Vector Graphics (SVG) 1.1 (Second Edition); <http://www.w3.org/TR/SVG11/>
- [3] W3C: Scalable Vector Graphics (SVG) Tiny 1.2 Specification; <http://www.w3.org/TR/SVGTiny12/>
- [4] W3C: SVG Working Group Wiki; http://www.w3.org/Graphics/SVG/WG/wiki/Main_Page
- [5] W3C: Scalable Vector Graphics (SVG) 2 (Working Draft); <http://www.w3.org/TR/SVG2/>
- [6] W3C: Scalable Vector Graphics (SVG) 2 (Editor’s Draft); <https://svgwg.org/svg2-draft/>
- [7] GitHub: SVG Working Group specifications; <https://github.com/w3c/svgwg>
- [8] Meinike, T.: Publikationen; <http://www.iks.hs-merseburg.de/~meinike/publikationen.php>
- [9] Meinike, T.: SVG – Learning by Coding; <http://svglbc.datenverdrahten.de/>
- [10] Meinike, T.: SVG auf dem Weg zum reichhaltigen Medienformat – Neuerungen in Version 1.2. In: tekomp, Gesellschaft für technische Kommunikation e. V., Tagungsband zur Jahrestagung 2004 in Wiesbaden, S. 337–340
- [11] Bah, T.: <http://tavmjong.free.fr/blog/> | <https://plus.google.com/+TavmjongBah/>
- [12] Dahlström, E.: <http://dahlstrom.net/> | <https://twitter.com/erikdahlstrom>
- [13] McCormack, C.: <http://mcc.id.au/blog/> | <https://twitter.com/heycam>

[14] Schepers, D.: <http://schepers.cc/> | <https://twitter.com/shepazu>

[15] Schulze, D.: <http://blog.dschulze.com/> | <https://twitter.com/dirkschulze>

**für Rückfragen:
thomas.meinike@hs-merseburg.de**