**SVG: Замена Flash-у**

Eric Vitiello Jr.

На протяжении последних 10 лет компания Macromedia является доминирующей силой, продвигающей векторную графику и анимацию в Web. Времена меняются и на горизонте появляются новые методы. В векторной графике появился новичок - Scalable Vector Graphics (SVG) (масштабируемая векторная графика), язык, основанный на XML, разрабатываемый консорциумом W3C.

Большинство из вас, читатели, помнят, что в начале 90-х годов Macromedia разработала продукт под названием Shockwave. Shockwave позволял разработчикам мультимедийных программ создавать анимации и даже программировать их, что открывало невиданные до селе возможности в интерактивной анимации.

Броузеры тех времен стали поддерживать Shockwave с помощью plug-in-ов, и вскоре Macromedia осознала, что необходимо создать облегченную версию Shockwave, которая будет предназначена исключительно для броузеров. В 1996 году Macromedia приобрела компанию FutureWave software, чей продукт - FutureSplash - стал тем, что мы знаем как Flash 1.0.

Примерно в середине 90-ых на небосклоне стала восходить звезда языка XML, благодаря его простому методу организации данных. На языке XML было создано множество схем данных (data schemas). Схемы данных описывали, например, как хранить и выводить математическую информацию (MathML), как описывать ресурсы общего характера (RDF), и даже как хранить и выводить химическую информацию (CML). Этот метод организации данных был положен в основу SVG.

Наступил 1999 год. Этот год - год рождения новой XML схемы данных - SVG. SVG схема определяет, как в XML записывается структура данных, описывающих векторную или растровую картинку. В спецификации языка SVG также описано, как создавать SVG-анимации, SVG-программы и документы. Все это делает SVG сильным конкурентом Flash-у.

**Подробнее об SVG**

С первого взгляда ясно, что спецификация языка SVG предлагает альтернативу многим функциям, имеющимся в наличии у Flash, а также добавляет много других. Ядром SVG является метод создания векторных графических элементов точно так же, как Flash строит всю анимацию на основных геометрических фигурах. Также как и Flash, SVG позволяет создавать анимацию для каждого элемента, позволяет управлять элементами с помощью скриптов через DOM, JavaScript, ECMAScript или с помощью любого другого скриптового языка, который поддерживается SVG-программой. В SVG разработчику доступны многие основные элементы, включая окружности, прямоугольники, эллипсы, многоугольники, кривые и текст. Так же как и в HTML, отображением элементов можно управлять с помощью стилей (CSS2), либо напрямую с помощью атрибутов.

В настоящее время существует множество SVG-броузеров и редакторов, позволяющих создавать и просматривать SVG-документы. Компания Adobe создала plug-in, который позволяет просматривать SVG-файлы в броузере, а также включил поддержку работы с SVG-Файлами в Illustrator 10. Другие компании, такие как JASC, Corel, Sun и IBM также включили в свои продукты определенную поддержку SVG. Так как схема SVG достаточно проста, вы можете без труда создать SVG-файл даже в простом текстовом редакторе.

Простой SVG-файл, рисующий черный круг, будет выглядеть следующим образом:

1: <?xml version="1.0" encoding="UTF-8" standalone="no"?>

2: <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"

3: "http://www.w3.org/TR/SVG/DTD/svg10.dtd">

4: <svg xmlns="http://www.w3.org/2000/svg">

5: <title>A Circle</title>

6: <circle cx="100" cy="100" r="50" stroke="black" stroke-width="2px" fill="none"/>

7: </svg>

Обратите внимание: первая строка - стандартный заголовок XML-документа, во второй и третьей строке расположен заголовок SVG DOCTYPE. В 4-ой строке располагается корневой элемент SVG-документа с указанием пространства имен (name space) SVG. Внутри этого элемента располагается элемент <title> и элемент для рисования круга <circle>. В элементе <circle> мы задаем координаты центра круга (атрибуты cx и cy), радиус (атрибут r). Остальные атрибуты определяют, какой инструмент используется для рисования края круга, какой у края будет цвет, и каким цветом будет залит сам круг (этот атрибут имеет значение "none" - т.е. у круга заливки не будет) .

В SVG документ можно вставлять и другие элементы. Например, можно вставить PNG, GIF или JPG картинку с помощью элемента <image>. С помощью SVG-схемы можно определить свои собственные шрифты, и даже написать текст вдоль кривой. С помощью скритов можно менять все атрибуты всех этих элементов, а также создавать анимации, о чем речь пойдет ниже.

**Работаем с SVG**

Создать SVG файл крайне просто, и в отличие от Flash, для этого вам не понадобится пользоваться каким-либо графическим редактором или приобретать какое-либо ПО - plug-in от Adobe распространяется бесплатно. Эта простота дает SVG еще одно преимущество перед Flash: благодаря текстовой природе XML, SVG-файлы могут индексироваться поисковыми серверами. Таким образом вам не нужно создавать дополнительные метафайлы для поисковых серверов, как это делается в случае с Flash. SVG-документ можно с легкостью вставить в HTML-документ.

Так как SVG-формат базируется на XML, разработчики получают возможность строить SVG-изображения основываясь на данных, которые также могут храниться в XML-формате. Преобразуя XML-данные в SVG с помощью простого XSL, можно легко получить графическое представление любых данных. Более того, можно например в SVG-графике отобразить, как располагаются столы в офисе, на основе данных, записанных в XML формате.

XML данные:

1: <?xml version="1.0"?>

2: <cubicles>

3: <cubicle north="10" east="15" width="10" length="10"/>

4: <cubicle north="0" east="0" width="10" length="10"/>

5: </cubicles>

XSL преобразование XML в SVG:

1: <?xml version='1.0'?>

2: <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

3: <xsl:template match="/">

4: <svg xmlns="http://www.w3.org/2000/svg">

5: <title>Our Cubicles</title>

6: <xsl:apply-templates select="cubicle"/>

7: </svg>

8: </xsl:template>

9: <xsl:template match="cubicle">

10: <rect x="{north}" y="{east}" width="{width}" height="{length}"/>

11: </xsl:template>

12: </xsl:stylesheet>

Конечный SVG документ:

1: <?xml version="1.0" standalone="no"?>

3: <svg xmlns="http://www.w3.org/2000/svg">

4: <title>A Circle</title>

5: <rect x="10" y="15" width="10" height="10"/>

5: <rect x="0" y="0" width="10" height="10"/>

8: </svg>

В результате данной трансформации создается графическое изображение. Показывающее, как располагаются в офисе столы. По этому представлению гораздо проще понять, как столы расположены относительно друг друга. Возможности представления данных графически безграничны, и с помощью SVG осуществляются очень просто.

**Анимация в SVG**

Анимация, как было уже сказано выше, осуществляется в SVG с помощью языка SMIL (Synchronized Multimedia Integration Language), который также является технологией, разработанной консорциумом W3C. В целях демонстрации возможностей SVG в сфере анимации, мы можем взять наш предыдущий пример с кругом и анимировать его, добавив элемент <animate>:

1: <?xml version="1.0" standalone="no"?>

2: <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"

3: "http://www.w3.org/TR/SVG/DTD/svg10.dtd">

4: <svg xmlns="http://www.w3.org/2000/svg">

5: <title>A Circle</title>

6: <circle cx="100" cy="100" r="50" stroke="black" stroke-width="2px" fill="none">

7: <animate attributeType="XML" attributeName="r" from="0" to="50" dur="2s"/>

8: </circle>

9: </svg>

Анимация может запускаться по определенному событию, например по событию "onmouseover", "onclick", либо с помощью скрипта, что позволяет создавать интерактивную графику. У каждого элемента ко всему прочему есть еще и свои собственные события, к которым также можно привязывать скрипты. Короче, существует множество способов сделать SVG-документ интерактивным, и делается это весьма просто. Создание сложных анимаций и сложных наборов графических изображений также не вызывает никаких трудностей в SVG. Любой объект в SVG-документе является XML-элементом, и к любому этому элементу можно получить доступ через DOM. Благодаря этому вы можете определить, как и куда каждый из элементов должен двигаться, как и на какие события реагировать.

С помощью скриптов можно поменять любое значение любого элемента в документе, а также поменять взаимное расположение элементов относительно друг друга, и всю структуру документа. Изменение структуры документа может играть важную роль, так как от этого зависит то, как документ будет выводиться на экран. Например, если элемент <line> (линия) расположен в документе после элемента <circle> (круг), линия будет выведена поверх круга.

**SVG в процессе**

Разработка спецификации SVG по-прежнему продолжается. В настоящее время спецификация SVG 1.1 приобрела статус Last Call Working Draft (Окончательный рабочий черновик проекта). Также в разработке находятся спецификации SVG Basic и SVG Tiny, предназначенные для мобильных устройств, что позволит в будущем мобильным телефонам и наладонным компьютерам выводить на экран SVG-файлы. Разработка этих спецификаций приведет в конечном счете к широкому признанию и использованию SVG точно так же, как непрекращающаяся разработка HTML привела к расцвету этой технологии.

Очень скоро SVG будет использоваться повсеместно, а так как формат SVG не является собственностью какой-либо компании, его популярность будет расти стремительно. Flash будет еще некоторое время доминировать в Web из-за большого количества установленных plug-in-ов. Однако, и SVG не отстает. Благодаря распространению SVG plug-in-ов через броузеры, количество пользователей SVG вырастет очень быстро так же, как это было с Flash. Будущие версии различных броузеров будут уже заранее включать в себя SVG-поддержку, а некоторые имеют ее уже сейчас.

Возможности использования SVG - широки, а благодаря трансформациям XML в SVG, эти возможности становятся практически безграничными. Примером такой гибкости может служить например преобразование накопленных статистических данных по населению в цветную карту в зависимости от выбранных районов и временного интервала. Изменение выбранного интервала меняет цвет графика, так как меняется количество населения выбранного района. Изменение, внесенные в XML-данные, немедленно отображаются в SVG-графике.

SVG можно использовать так же как и Flash - для создания системы меню, навигации по сайту и даже для создания всего сайта целиком. Компания Adobe представила прекрасный ресурс, где продемонстрированы некоторые возможности SVG: преобразование химических данных в 3-хмерные изображения молекул, рисование графиков и схем, генерация на лету театральных билетов с интерактивного системой их заказа, и также приложение, предназначенное для создания SVG-изображений.

Установка Adobe SVG plug-in-а очень простая. Зайдите на Adobe's SVG download и загрузите plug-in, соответствующий вашей операционной системе. Перед просмотром первого SVG-документа на экране появится окно с лицензионным соглашением. Оно достаточно простое.

Попробуйте поработать с SVG сами. Может оказаться, что вам понравится то, как редактируя вручную XML-данные, вы можете преобразовывать их в графическое изображение, которым можно полностью управлять.