Объектная модель XML. DOM

XML — это текстовый формат, удобный для хранения и передачи структурированных данных. Но работать с ним напрямую сложно в любом языке программирования. Чтобы можно было легко считывать из XML-документа данные или модифицировать его, нужно вначале его преобразовать в более подходящую форму. На данный момент для решения этой задачи наиболее широко используется программная модель DOM, основанная на том, что любой корректный документ может быть представлен в виде дерева объектов. Применяется DOM и Flash-плеером. DOM (Document Object Model — объектная модель документа) была разработана комитетом W3C и реализована в большинстве языков и программных сред. Ее основной конкурент — основанная на событиях модель SAX — распространена куда меньше. Причиной столь значительной популярности DOM является ее простота и интуитивность. В ActionScript реализован упрощенный вариант DOM, имеющий много отклонений от стандарта. Его рассмотрению мы посвятим данный раздел.

Как уже говорилось, основной идеей DOM является то, что для того, чтобы описать XML- документ, нужно построить дерево объектов. Каждый объект изображает один узел (проще говоря, тег). У каждого объекта есть массив ссылок на объекты, соответствующие дочерним узлам описываемого им узла дерева XML. Они расположены в том же порядке, в котором прописаны теги, на основании которых они были заданы. У каждого объекта узла также есть свойство, ссылающееся на объект родительского узла. В особом объекте, связанном с объектом узла, в форме свойств хранятся его атрибуты.

В ActionScript есть специальный класс, объекты которого выступают в роли узлов дерева DOM. Это класс XMLNode. Его свойства и методы дают возможность осуществлять навигацию по дереву, считывать данные и производить разнообразные модификации. Помимо XMLNode, в ActionScript есть еще один класс, связанный с работой с XML. Этот класс называется XML, и его объекты олицетворяют документ в целом. Этот класс необходим, чтобы можно было проводить операции, относящиеся ко всему документу. К таким операциям относятся, например, считывание и инициализация XML-объявления и DDT-объявления, загрузка содержимого и отправка документа на сервер, преобразование XML-текста в дерево объектов DOM и так далее.

Через объект класса XML осуществляется доступ к объекту дерева DOM, соответствующего корневому тегу. Для _____большей интуитивности, эта операция осуществляется при помощи тех же инструментов, с использованием которых

можно перейти от объекта родительского узла к узлу дочернему. Поэтому описывающему документ объекту класса XML должны быть доступны те же свойства и методы, что и объектам узлов дерева DOM. Чтобы это было возможно, класс XML был сделан подклассом XMLNode. В прототипе XMLNode хранятся элементы, предназначенные для работы с деревом документа. В прототипе же XML записаны методы и свойства, предназначенные для проведения более общих операций.

На практике объектом класса XML используются лишь свойства класса XMLNode, предназначенные для доступа из объекта родительского узла к объекту узла дочернего. Эта операция нужна, чтобы получить доступ к объекту класса XMLNode, описывающему корневой узел документа. То есть, формально объект класса XML является родительским узлом для объекта корневого узла описывающего XML-документ дерева DOM.

Нужно помнить о различиях между классами XML и XMLNode. Попытка задействовать метод или свойство, присущее только классу XML, через объект узла, приведет к сбою. Также нужно понимать, что для того, чтобы сделать некий метод или свойство доступным объектам всех узлов, его нужно записать в прототип класса XMLNode. Если сохранить его в прототипе класса XML, то объекты, образующие дерево DOM, использовать его не смогут.

Довольно странной на первый взгляд особенностью DOM, реализуемой Flash-плеером, является то, что вложенный в тег текст не помещается в свойство описывающего узел объекта, а для его хранения формируется новый объект узла. Этот узел считается дочерним узлом узла, к которому относится текст. Узел, сформированный на основании тега, и узел, заданный исходя из текста, формально относятся к разным типам узлов — ELEMENT и TEXT. Реально же описывающие их объекты предельно схожи, и различия между ними можно обнаружить только посредством свойства nodeType.

В “настоящем” DOM от W3C на основании любого элемента XML создается объект узла. Различаются эти объекты по типу. Обычному _____узлу, сформированному на основании некорневого тега, соответствует тип ELEMENT. Узел, описывающий текст, вложенный в тег, относится к типу TEXT. Узел, хранящий атрибут, принадлежит к типу Attr — и так далее. Всего типов узлов в DOM 12. И лишь два из них — ELEMENT и TEXT — есть в DOM, реализуемой Flash-плеером. Это связано как с тем, что анализатор XML плеера поддерживает не все возможности данного языка, так и с тем, что DOM, применяемая в ActionScript, имеет ряд отклонений от стандарта (например, для каждого атрибута не создается отдельного объекта узла).

Если текст узла разрывается тегом, то каждая из его частей рассматривается анализатором как самостоятельная (поэтому узел типа TEXT не может иметь дочерних узлов). К примеру, следующий документ даст дерево DOM, у которого будет 4 узла второго уровня: два типа ELEMENT и два типа TEXT:

<text>

<tab/> Предком XML является <link addr="http://w3c.org"/> SGML

</text>

Если бы использовалась DOM от W3C, то у дерева рассматриваемого документа было бы три уровня (узел типа Attr был бы создан для атрибута addr тега link).

DOM — это далеко не идеальная система. Главный ее недостаток заключается в том, что документ должен быть полностью разобран, прежде чем с ним можно будет начать работу. А это может потребовать значительного времени. Также в случае больших XML-документов дерево DOM может занять большой объем памяти. Альтернативой DOM, используемой для анализа XML- данных в режиме реального времени, является SAX (Simple API for XML). Данная технология основана на событиях: когда SAX разбирает очередной тег, она посылает соответствующее сообщение. Увы, но пока в ActionScript нет возможности работать с моделью SAX.