

# ЯЗЫК ОПИСАНИЯ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ MOZILLA XUL

М.В. Трофимов, IV курс, Институт информатики, инноваций и бизнес-систем ВГУЭС

*Владивостокский государственный университет экономики и сервиса, Владивосток*

Язык Mozilla XUL, XML User-interface Language, это язык описания графического интерфейса пользователя, диалект языка XML (eXtensible Markup Language, расширяемый язык разметки). Mozilla Firefox, Mozilla Thunderbird, Mozilla Sunbird, ActiveState Komodo, NVU и KompoZer – примеры программ, использующих данную технологию.

Так сложилось, что аббревиатура XUL происходит от имени персонажа Зул из фильма «Охотники за приведениями», в силу этого XUL, произносится как «зул». Кроме того, Venkman, отладчик в Mozilla Navigator, тоже оттуда. Ключевой оказалась фраза из фильма «Нет больше Даны, есть только Зул» трансформировавшаяся в слоган «Нет больше данных, есть только XUL». Это основная концепция XUL, заключающаяся в отделении данных от логики приложения. Достигается это посредством следующих абстрактных слоев:

- содержание (content): объявление окон и элементов пользовательского интерфейса, ассоциированных с ними (достигается посредством XUL);
- оформление (skin): включение CSS и изображений, определение вида приложения;
- локализация (locale): текст, отображаемый в пределах приложения, распределен по специальным локальным файлам (в основном DTD и HTML).

Язык XUL является частью XPFE, Cross Platform Front End, кроссплатформенной среды, предназначенной для того, чтобы разрешить проблемы создания графических интерфейсов, которые будут работать под любой операционной системой. На данный момент поддерживаются операционные Windows, Unix и Mac. Технология XPFE очень похожа по структуре на Dynamic HTML в том, что также используется JavaScript для развития функциональности и CSS (Cascading Style Sheets, каскадные таблицы стилей) для дизайна. Отличие лишь в том, что для DHTML используется язык разметки HTML, а для XPFE – язык описания графического интерфейса XUL.

Язык XUL это полнофункциональный язык разметки, соответствующий стандарту W3C XML 1.0. Приложения, написанные на XUL, также могут использовать HTML, CSS, DOM (Document Object Model, объектная модель документа) и JavaScript. Технология позволяет создавать пользовательские интерфейсы любой сложности, содержащие все пользовательские элементы, которые можно увидеть в современных десктоп-программах («программы рабочего стола» с которыми пользователь имеет дело всякий раз, когда пользуется компьютером). Кроме того, имеется возможность разрабатывать собственные компоненты для применения в рамках XUL. Также имеется возможность добавлять свои собственные тэги в код XUL-документа, что исключает нехватку элементов управления и дает эффект повторного использования кода.

**Пример XUL-документа, состоящего из пользовательского окна с надписью “hello, world” .**

```
<?xml version="1.0"?>
<window
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <description>hello, world</description>
```

</window>

В примере XUL-документа первая строка - заголовок XML-документа, спецификация 1.0. Вторая строка - корневой тэг <window>. В соответствии с этим тэгом, система визуализации "отрисует" пользовательское окно. Третья строка - пространство имен, характерное для каждого XUL-документа. По адресу этого пространства имен в Интернет есть страница, которая ни для чего не используется, но если по нему перейти, то можно увидеть на странице фразу «THERE IS NO DATA. THERE IS ONLY XUL.», что с английского значит «НЕТ БОЛЬШЕ ДАННЫХ, ЕСТЬ ТОЛЬКО XUL». Четвертая строка содержит тэг <description>, который предназначен для вывода текста. В XUL, как и в XML, все тэги закрывающиеся. Файл с кодом нужно сохранить с расширением ".xul" и открыть в браузере Netscape Navigator или Mozilla, или в браузере, основанном на их коде (Firefox, K-Meleon, Camino и др.).

Сложные графические интерфейсы помогает строить коллекция виджетов (widgets), которыми являются пользовательские элементы, такие как кнопка, надпись, выпадающий список, текстовое поле и более сложные – прогресс-бар, мастер. Ниже представлены некоторые из них (код).

#### **Использование виджетов «Кнопка и надпись».**

```
<?xml version="1.0" encoding="windows-1251"?>
<window
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <description>Привет</description>
  <button label="Я Кнопка"/>
</window>
```

В примере атрибут encoding со значением "windows-1251" нужен для указания того, что для корректного отображения текста необходимо, браузером, использовать русскую кодировку для Windows. Далее в примерах заголовок XML документа, корневой тэг <window> и идентификатор пространства имен XML будут опущены.

#### **Поле ввода и кнопка:**

```
<textbox value="что искать?"/>
<button label="поехали!"/>
```

#### **Многострочное поле ввода:**

```
<textbox multiline="true"/>
```

Поле ввода становится многострочным с установкой атрибута multiline="true".

#### **Список:**

```
<listbox width="100" height="100">
  <listitem id="a1" label="Ты"/>
  <listitem id="a2" selected="true" label="Я"/>
  <listitem id="a3" label="Он" />
  <listitem id="a4" label="Она" selected="true"/>
</listbox>
```

Атрибуты width и height задают ширину и высоту элемента соответственно (в пикселях). Атрибут id – это уникальный идентификатор. Хорошим тоном является установка этого идентификатора к любому элементу. Служит неким уникальным

именем, по которому можно будет обращаться к нему из других частей программы. Атрибут `selected` указывает на выбранный пункт.

### **Ниспадающий список:**

```
<menulist>
  <menupopup>
    <menuitem label="Первый" selected="true"/>
    <menuitem label="Второй"/>
    <menuitem label="Третий" />
    <menuitem label="Четвертый и последний"/>
  </menupopup>
</menulist>
```

Использование JavaScript и CSS в XUL почти ничем не отличается от использования их в HTML. Для подгрузки файла с JavaScript-кодом используется строка `<script type="application/x-javascript" src="code.js"/>` и для подгрузки файла с CSS-кодом директива `<?xml-stylesheet href="rules.css" type="text/css"?>`.

### **Ниже приведен пример использования JavaScript и CSS в XUL-документе.**

Файл `func.js` с JavaScript кодом:

```
function Hello(text){
  alert(text);
}
function Bye(){
  alert('Прощай из JS...');
  close();
}
```

Код XUL-документа:

```
<?xml version="1.0" encoding="windows-1251"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <script src="func.js" type="application/x-javascript"/>
  <label control="idHi" value="Жми кнопку..."/>
  <radiogroup>
    <radio id="a1" selected="true" label="Выбери меня!"/>
    <radio id="a2" label="Нет, меня!"/>
    <radio id="a3" label="Ммм, а меня?"/>
    <radio id="a4" label="Я тоже хочу..."/>
  </radiogroup>
  <button id="idHi" label="Привет!" onclick='Hello("Привет от JS!");'/>
  <button id="idBye" accesskey="q" label="Пока?" onclick='Bye();'/>
</window>
```

Строка `<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>` указывает браузеру, что он должен использовать глобальные стилевые таблицы. Тэг `<radio>` представляет кнопку выбора (radio button), а тэг `<radiogroup>` служит для объединения нескольких тегов `<radio>` в одну группу (объединение нескольких кнопок выбора в одну группу, из которой пользователь может выбрать только одну). Атрибут `onclick` служит для привязки события нажатия кнопки к соответствующей функции. Атрибут `accesskey` служит для назначения "горячей клавиши". Клавиша, присвоенная атрибуту

и нажатая вместе с клавишей CTRL или клавишей ALT (может быть вместе с клавишей SHIFT, в зависимости от платформы), приводит к срабатыванию кнопки.

Язык XUL позволяет создавать достаточно богатые по функциональности пользовательские интерфейсы, которые можно запускать как стандартные приложения (десктоп-программы), так и загружать их из сети Интернет. При этом приложения на XUL можно легко настраивать, менять в них текст или графические объекты, переводить на различные языки и т.д. Для программирования на XUL не требуется специальных навыков. Любой web-разработчик, знакомый с Dynamic HTML, сможет быстро выучить XUL и начать создавать свои приложения. Отличительные особенности XUL:

- мощный язык разметки с поддержкой пользовательских элементов (widgets). В отличие от DHTML, с помощью которого можно создавать web-страницы, с помощью XUL можно создавать переносимые приложения, содержащие окна, кнопки и другие элементы управления;

- XUL основан на существующих стандартах. XUL - диалект XML, основанный на стандарте W3C XML 1.0. Приложения, написанные на XUL, используют дополнительные стандарты W3C как HTML 4.0, CSS (версии 1, 2 и частично CSS 3), DOM (уровня 1, 2 и частично DOM 3), JavaScript 1.5, включая ECMA-262 Edition 3 (ECMAScript), XML 1.0;

- межплатформенная переносимость (характеризуется поддержкой Windows, Mac и множеством Unix-подобных операционных систем);

- легкость локализации, модификации или настройки.

Язык XUL является легким в освоении, гибким и переносимым языком. Но существует проблема - пользователь вынужден ставить браузер Mozilla (или другой продукт на его базе) для запуска XUL-приложений. Это затрудняет распространение приложений на базе XUL (в силу большого размера сопутствующего ПО - компоненты, библиотеки и другие файлы). XULRunner и XRE (XUL Runtime Environment) - попытка избавиться от этой зависимости. Данные средства позволяют запускать XUL приложения обычным образом: через программу-загрузчик (XULRunner), или посредством встраиваемых библиотек для рендеринга (XRE или libxul). Кроме того, применение XULRunner позволяет устанавливать, обновлять или удалять приложения таким же способом, как и обычные программы Mozilla (Firefox или Thunderbird).