

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ

УДК 004.738.5

ПОСТРОЕНИЕ НАВИГАЦИИ ДЛЯ ИЕРАРХИЧЕСКИХ СТРУКТУР
В WEB-СИСТЕМАХ И СИСТЕМАХ УПРАВЛЕНИЯ WEB-САЙТОМ

Беляев А.В.

Постановка проблемы и анализ публикаций. Основным способом представления информации в Интернет является гипертекстовая web-среда. Документ в формате гипертекста, помимо текстовой, графической и мультимедийной информации, может содержать связи на другие документы или их фрагменты. С помощью связей можно организовать навигацию по всем созданным документам. Сформированные таким образом сложные информационные ресурсы имеют нелинейную структуру. Благодаря этим возможностям гипертекст и получил всеобщее признание [1].

Однако рост количества публикаций в гипертекстовой web-среде и отсутствие универсальных схем навигации по большому количеству документов, зачастую приводит к дезориентации пользователей. Например, знакомясь с учебным материалом, представленном в гипертекстовом формате, необходимо пройти по всем разделам ресурса, однако в неструктурированной специальным образом гипертекстовой среде, очень легко пропустить какой-нибудь важный материал.

В Интернет вся информация сгруппирована в web-ресурсы, отличительной особенностью каждого из них является собственная система навигации. Это может быть: линейная структура, иерархическая структура, нелинейная структура, смешанная структура [2]. Построение подобной системы для web-ресурсов с небольшим количеством разделов не представляет собой сложную задачу, а навигация загружается вместе с документом полностью. Однако для web-ресурсов большого объема такая схема не подходит. Поскольку каналы связи имеют ограничения по скорости передачи данных, а web-сервер имеет ограничения по продолжительности сессий, передача большого объема информации между клиентом и сервером может происходить в течение длительного времени либо вообще прерваться [3].

Цель статьи. В данной статье рассмотрен способ построения навигации для древовидной, иерархической структуры, которая не потребует предварительной загрузки всей панели навигации, загружаются только разделы верхнего уровня. Листовые узлы такой системы позволяют при «клике» мышки развернуть соответствующий список (ветку дерева). При этом формирование и появление списка должно происходить динамически, в момент совершения клика. В основу способа, рассмотренного, в данной статье, взяты возможности таких технологий как JavaScript и объекта XMLHttpRequest [4, 5], составляющие костяк технологии AJAX. А так же учтены особенности реализации этой схемы для наиболее популярных версий web-браузеров.

Основная часть

Представим себе web-систему, в которой содержимое и структура определены в реляционной базе данных, и представлены двумя таблицами «объекты» и «связи», рисунок 1. Наиболее часто структура web-сайта представляет собой древовидную иерархию, рисунок 2. Каждая ветка иерархической структуры представляет собой линейный список. Выборка ветки дерева с помощью SQL будет выглядеть следующим образом:

```
SELECT * FROM `object` , `relation` WHERE parent = id_листа and child = id, (1)
```

Определим функцию, которая используя запрос (1) будет возвращать массив связанных объектов узла ID:

function childNodes(\$ID)

(2)



Рис.1 Схема таблиц базы данных

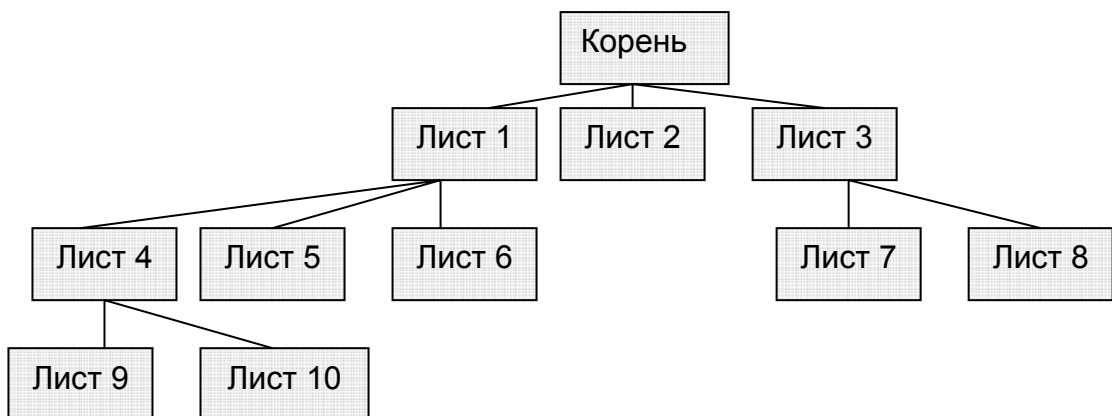


Рис. 2 Древоподобная иерархия

Исполнение функции (1), содержащее обращение к базе данных возможно только на стороне web-сервера. Поэтому следующей задачей является создание механизма обращения к web-серверу без перезагрузки страницы в момент «клика» на узел, подразделы которого мы хотим увидеть.

Такая возможность появилась не так давно. В начале Microsoft Internet Explorer создал элемент ActiveX, а затем Mozilla Firefox и Opera также создали объект XMLHttpRequest.

В Microsoft Internet Explorer он определяется:

```
var req = new ActiveXObject("Microsoft.XMLHTTP");
```

В Mozilla и Opera это делается проще (так как там это объект, встроенный в JavaScript):

```
var req = new XMLHttpRequest();
```

Для обеспечения поддержки функции во всех версиях браузеров необходимо создать две ветки. Наиболее удачный механизм вызова объекта XMLHttpRequest приведен на сайте компании Apple [5], воспользуемся им для решения наших задач. Первый фрагмент это функция, реализующая непосредственный запрос к удаленному серверу, в качестве параметра *url* необходимо будет подставить удаленный адрес программы, содержащей функцию (2):

```

var req;

function loadXMLDoc(url) {
    req = false;
    // ветка для объекта XMLHttpRequest object
    if(window.XMLHttpRequest && !(window.ActiveXObject)) {
        try {
            req = new XMLHttpRequest();
        } catch(e) {
            req = false;
        }
    }
    // ветка для Microsoft Internet Explorer /Windows ActiveX version
    } else if(window.ActiveXObject) {
        try {
            req = new ActiveXObject("Msxml2.XMLHTTP");
        } catch(e) {
            try {
                req = new ActiveXObject("Microsoft.XMLHTTP");
            } catch(e) {
                req = false;
            }
        }
    }
    if(req) {
        req.onreadystatechange = processReqChange;
        req.open("GET", url, true);
        req.send("");
    }
}

```

Рис. 3 Функция вызова удаленного документа

Здесь стоит обратить внимание на значение *onreadystatechange*, которому присваивается значение функции *processReqChange*, являющейся хендлером события, вызываемого при изменении состояния объекта *req*.

```

function processReqChange() {
    // только если объект req показывает «load»
    if (req.readyState == 4) {
        // только если "OK"
        if (req.status == 200) {
            // здесь необходимо разместить код, возвращенный по удаленному запросу и требующий показа.
            response = req.responseText;
            document.getElementById(Объект в котором должны показаться ответ сервера).innerHTML=response;
            response='';
        } else {
            alert("Проблема восстановления XML данных:\n" + req.statusText);
        }
    }
}

```

Рис. 4 Обработчик события, отвечающий за процесс загрузки удаленного документа

Таким образом, удаленный документ должен содержать вызов функции *childNodes(\$ID)* - (1), формирующей. Основной документ, отвечающий за навигацию по древовидной, иерархической структуре, должен содержать:

- Код отображающий дерево

- Вызов события, в узле каждой ветки (например onclick())
- Обработчик события, вызывающий функцию loadXMLDoc и обработчик processReqChange (рисунки 3, 4)
- Контейнер для вывода полученного от удаленной функции линейного списка связанных с узлом объектов
- Стили CSS, позволяющие управлять видимостью содержания загруженных веток

Выводы

Предложенный метод построения навигацию по древовидной, иерархической структуре, позволяет организовать навигацию для ресурсов повышенного объема, сохраняя возможность динамического перестроения структуры ресурсов, копирования и перемещения веток дерева. В качестве примера реализации данного алгоритма можно привести фрагмент “дерева” сайта газеты «День» (www.day.kiev.ua), реализованного на основе системы управления сайтом CimWebCenter [6] использующей этот метод.

In given article the way of construction of navigation for treelike, hierarchical structure with leaf like node, which gives possibility to develop the corresponding node at a mouse "clique". Thus formation and list occurrences (a tree branch) should occur dynamically, at the moment of clique fulfillment.

1. Демин И.С. Проблемы развития гипертекстовых сред. // Вестник Оренбургского государственного университета, 2004, 4.
2. Инькова Н.А., Зайцева Е.А., Кузьмина Н.В., Толстых С.Г. СОЗДАНИЕ Web-сайтов // Учебно-методическое пособие. Ч. 5. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2002. 56 с.
3. Арушанян О.Б., Богомолов Н.А., Ковалев А.Д., Сеницын М.Н. Архитектура клиентского программного обеспечения для WEB-приложений, ориентированных на представление данных // Вычислительные методы и программирование. 2004. Т.5.- С.24-37.
4. McLellan D. Very Dynamic Web Interfaces. // The XML.commune February 09, 2005. www.xml.com/pub/a/2005/02/09/xml-http-request.html
5. Apple Inc. Dynamic HTML and XML: The XMLHttpRequest // <http://developer.apple.com/internet/webcontent/xmlhttpreq.html>
6. Компания CiM. Документация CimWebCenter // <http://www.cimwebcenter.com/documentation>