

обработки, что улучшает характеристики сжатия. При этом весьма значительно уменьшается суммарное время преобразования, поскольку оно максимально в точке $k = n/2$ и быстро уменьшается при удалении от нее.

SUMMARY

The paper expounds compression of binary codes, in which use methods of binary binomial count. Algorithms of transformation are differ by simplicity and capability to adaptation. They are illustrated by examples and tables.

СПИСОК ЛИТЕРАТУРЫ

1. Борисенко А.А., Чередниченко В.В. Нумерация равновесных кодов на основе биномиальных чисел // Право і безпека. – 2004. - № 3 – 4. - С.194 -197.
2. Борисенко А.А. Биномиальный счет. Теория и практика: Монография. – Сумы: ИТД «Университетская книга», 2004. – 170 с.
3. В. Б. Чередниченко. Оценка быстродействия нумерации биномиальных чисел методом последовательного счета //АСУ и устройства автоматизи. – 2005. -№ 131. - С.25- 30.
4. Чередниченко В.В. Оценка времени преобразования равновесных кодов в биномиальные // Вісник СумДУ. – 2004 - № 12(71). - С. 113- 117.

Поступила в редакцию 12 мая 2006г.

УДК 519.17:519.816:004.715

РЕАЛИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ FLASH И JAVA APPLETS

Б.О. Кузиков, студ.; С.П. Шаповалов, доц.
Сумский государственный университет

Интеграция различных технологий является достаточно сложной проблемой. В статье исследованы возможности интеграции Flash и Java Applets, описаны механизмы возможного взаимодействия, рассмотрена их практическая реализация.

На рынке веб-приложений давно закрепились такие технологии, как Macromedia Flash и Java Applets. К безусловным преимуществам первого относится простота использования в сфере дизайна. Java же предоставляет богатые функциональные возможности, в частности в области вычислений и работы с различными серверами. В частности, в рамках лаборатории дистанционного обучения СумГУ, при подготовке учебного материала, Flash применяется для создания виртуальных лабораторных столов по ряду предметов, а на базе Java разрабатываются интерактивные тренажеры, типичными операциями для которых являются отображение и проверка формул, поддержка нескольких языков интерфейса, сохранение промежуточных результатов, связь с преподавателем.

Иногда возникают ситуации, когда нужно объединить наглядность представления данных со сложными математическими расчетами и гибким взаимодействием с сервером. Рассмотрим варианты возможного взаимодействия указанных технологий.

ПОСТАНОВКА ЗАДАЧИ

Следует понимать, что Java Applets, как и Flash, в разных системах имеет разную реализацию контейнеров-интерпретаторов, а их выполнение в контексте браузера налагает определенные ограничения в плане безопасности. Значит, прямое взаимодействие объектов на разных

платформах может различаться. Требуется найти некоторый универсальный способ их взаимодействия. Поэтому отдельно рассмотрим способы взаимодействия с внешними объектами каждой из этих технологий через контекст браузера.

РЕАЛИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ В FLASH

Начиная с Flash 6, разработчики применяли следующие способы для реализации взаимодействия с внешними объектами [1,2] .

1) Использование методов SetVariable, TCallLabel и fscommand. К положительным аспектам такого подхода можно было отнести скорость взаимодействия, надежность, возможность передачи больших объемов информации. Явным недостатком – невозможность реализации этого подхода для Mac OS. Наиболее популярный для этой системы браузер Safari не поддерживает такую возможность коммуникации, а Firefox требует Flash Player 8.

2) Использование 2-раздельных Flash-объектов, один из которых мы будем снова и снова создавать, используя тег PARAM для задания информации. Для взаимодействия между Flash-объектами используется Flash LocalConnection, для вызова javascript-методов используется метод getURL с параметрами вида "javascript:myMethod()". К преимуществам подхода можно отнести наиболее широкую кросс-браузерную поддержку данного метода коммуникации. Недостатки: ресурсоемкость, большие временные задержки, возможность передачи только малого объема данных.

Из вышеприведенного видно, что реализация двустороннего взаимодействия является достаточно сложной проблемой в Flash. Коренным образом ситуацию изменил выход Macromedia Flash 8. Новая версия предоставляет разработчикам пакет flash.external, в котором реализованы следующие методы [1] :

- static addCallback(methodName: String, instance :Object, method: Function):Boolean – средство декларации внешних интерфейсов, где:
 - methodName – имя, под которым будет зарегистрирован интерфейс;
 - instance – объект, содержащий метод;
 - method – метод, связываемый с внешним интерфейсом.
- static call(methodName: String, [parameter1:Object]):Object – средство вызова функций внешних объектов, будь то функции JavaScript или методы ActiveX-контейнеров. Заметим, что этот метод позволяет получать возвращаемые методами значения, реализуя тем самым синхронное взаимодействие между объектами, чего трудно было добиться ранее.

Проиллюстрируем использование вышеописанных методов.

```
import flash.external.*;

// Объявление внешнего интерфейса
ExternalInterface.addCallback("flashMethod", this,
flashMethod);
// Метод связываемый с внешним интерфейсом
function flashMethod(str:String):Void {
    in_ti.text =str;
}

function doClick(eventObj:Object):Void {
// Вызов метода getFromFlash объекта appletName
    ExternalInterface.call("appletName.getFromFlash",
out_ti.text);
}
```

В случае локального использования требуется установить в соответствующих тегах `object` и `embed` гипертекстового документа параметр `allowScriptAccess` с `"sameDomain"` на `"always"`.

Следует также указать существующие недостатки [2]:

- требует Flash Player 8;
- обладает невысокой производительностью (по некоторым оценкам порядка $O(n^2)$);
- блокирует браузер на время пересылки данных;
- не работает в Internet Explorer в случае, если Flash-объект был добавлен на страницу с помощью `document.writeln`, DOM-методов или `innerHTML`.

РЕАЛИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ В JAVA

Технология Java Applet предоставляет более простой доступ к внутренним объектам. Так, любые методы, объявленные как `public` доступны извне. Однако стандартными средствами не предусмотрен доступ к внешним объектам – не апплетам. Для реализации взаимодействия необходимо импортировать в проект пакет `netscape.javascript`, в котором на базе класса `JSObject` реализован доступ к объекту `window`-браузера, а также к его коллекциям [3,4]. Укажем основные методы данного класса:

- `static JSObject getWindow(Applet applet)` – возвращает JavaScript-объект `window`, соответствующий окну, содержащему апплет;
- `Object call(String methodName, Object [] args)` – выполняет указанный JavaScript-метод;
- `Object eval(String JavaScriptCode)` – выполняет указанный JavaScript-код;
- `Object getMember(String propertyName)` – возвращает значение указанного свойства.
- `Object getMember(String propertyName)` – возвращает значение указанного свойства;
- `void setMember(String propertyName, Object value)` – устанавливает значение указанного свойства;
- `Object getSlot(int arrayIndex)` – возвращает значение указанного элемента массива;
- `void setSlot(int arrayIndex, Object value)` – задает значение указанного элемента массива;

Проиллюстрируем использование указанного объекта.

```
import netscape.javascript.JSObject;

public class J2Flash extends JApplet implements
ActionListener {

    JTextField jTF_toF, jTF_fromF;
    JButton Send;
    ...
    // Метод, используемый для доступа из Flash
    public void getFromFlash(String S){
        jTF_fromF.setText(S);
    }
    // Метод, реализующий вызов внешнего метода
    private void setToFlash(String S){
    // Реализация крос-браузерной совместимости
        JSObject window = JSObject.getWindow(this);
        JSObject nav = (JSObject>window.getMember("Navigator");
```

```

    JSONObject document = (JSONObject)window.getMember("document");
    int is_ie =(nav.getMember("appName").toString().indexOf(
"Microsoft");
    JSONObject move = (JSONObject)((is_ie!= -1)?
        window.getMember("flashName"):
        document.getMember("flashName"));
//Вызов метода Flash
move.call(flashMetod,new String [] {S});
}
}

```

СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ

Выше были рассмотрены варианты реализации Flash и Java Applets с внешними объектами в целом. Рассмотрим пример их взаимодействия.

```

<html>
<body>
<table>
<tr><td>
<!-- flash -->
<object classid="..." codebase="..."
    width="330" height="245"
    id="flashName">
    <param name="allowScriptAccess" value="always" />
    <param name="movie" value="ExtInt.swf" />

    <embed src="ExtInt.swf"
        width="330" height="245"
        name="flashName"
        allowScriptAccess="always"
        swLiveConnect="true"
        type="application/x-shockwave-flash"
        pluginspage="..." />
    </object>
</td><td>
<!--applet -->
<applet width="330" height="245"
    code="J2Flash.class"
    name="appletName"
    MAYSCRIPT></applet>
</td></tr>
</table>
</html>

```

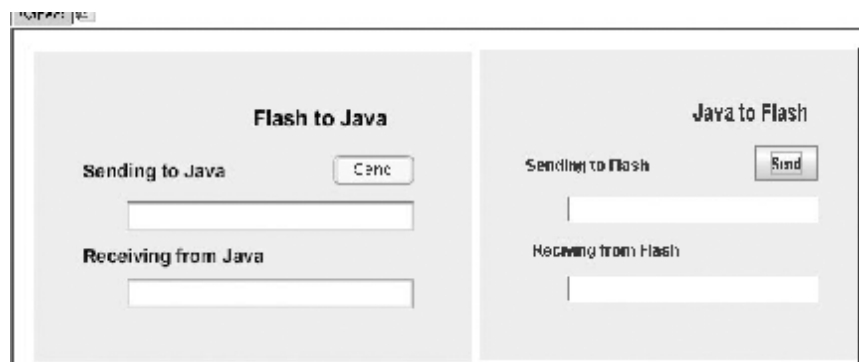


Рисунок 1

На рис. 1 представлен внешний вид объектов в браузере.

В данном примере подразумевается, что в документе, содержащем объекты, тег `applet` содержит атрибут `name="appletName"`, тег `object` и его субтег `embed`-атрибуты `id="flashName"` и `name="flashName"` соответственно. Атрибуты `swLiveConnect` и `MAVSCRIPT` включают поддержку технологии `LiveConnection`, необходимую для реализации взаимодействия в браузерах на основе `Netscape Navigator`. Атрибут `allowScriptAccess="always"` определяет политику безопасности в случае локального использования [1,4].

Принципиальная схема взаимодействия объектов представлена на рисунке 2.

В случае размещения `Flash`-ролика и апплета в разных фреймах или окнах браузера для организации взаимодействия целесообразно явно реализовать `JavaScript`-прослойку.

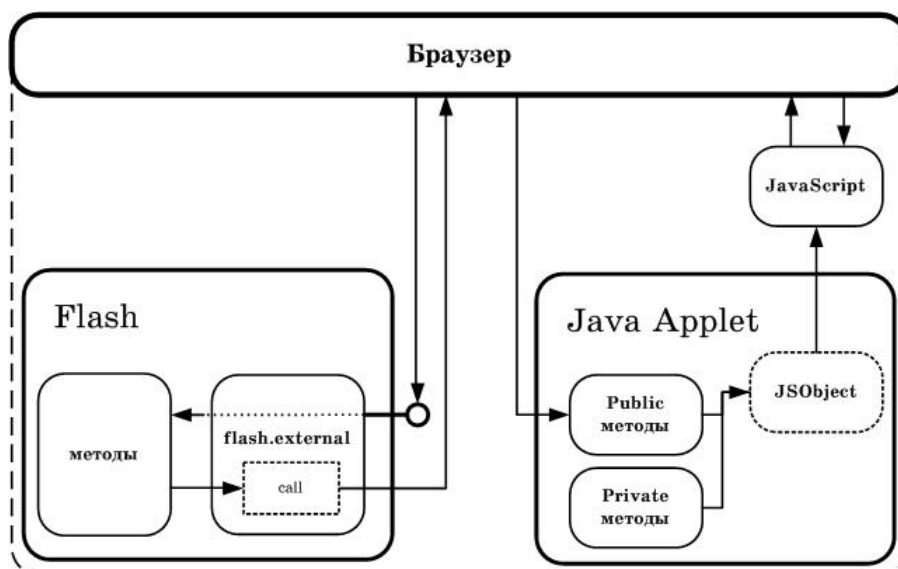


Рисунок 2

ВЫВОДЫ

В статье представлены способы установления взаимодействия между технологиями `Flash` и `Java applets` в контексте браузера. Указаны возможности этих технологий в зависимости от версии и платформы.

Был предложен вариант взаимодействия с использованием пакетов `flash.external` и `natscape.javascript`, как наиболее эффективный на основании анализа существующих в рамках данных технологий возможностей.

SUMMARY

This paper content describe the possibility way of communication between Java Applet, Flash and it's container. Been described JSObject, that's realise JavaScript objects in Java and can be used as the bridge between Java and any object included in browser content. Else been viewed using of package flash.external for creating interface that enables straightforward communication between Flash and external objects.

СПИСОК ЛИТЕРАТУРЫ

1. <http://livedocs.macromedia.com/flash/8/main/00002200.html> - Macromedia Falsh 8 Live Documentation.
2. <http://dir.gmane.org/> Обсуждение пакета dojo.flash

3. Холл, Марти, Браун, Лэрри. Программирование для Web. Библиотека профессионала /Пер. с англ. – М.: Вильямс, 2002. -1264с.: ил.
4. Client-Side JavaScript Guide.// Ch 15. Live Connect Overview.

Поступила в редакцию 15 мая 2006 г.

УДК 621.391.251

ДИНАМИЧЕСКОЕ АДРЕСНО-ВЕКТОРНОЕ СЖАТИЕ ДВОИЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

И.А.Кулик, доц.; С.Н. Харченко, магистр
Сумский государственный университет

С целью повышения коэффициента адресно-векторного сжатия более перспективным является применение динамических алгоритмов, которые позволяют кодеру быстрее, чем при статических алгоритмах, адаптироваться к изменениям в текущем потоке данных.

ВВЕДЕНИЕ

Для повышения эффективности использования коммуникационных и информационно-вычислительных ресурсов применяются различные методы и средства. Среди них важную роль играет сжатие, которое стало неотъемлемой частью систем хранения и передачи данных, обеспечивая необходимую скорость их обработки путем выявления скрытых резервов в производительности отдельных элементов и системы в целом, неучтенных на этапе проектирования. Это эквивалентно повышению пропускных способностей каналов передачи или увеличению емкости запоминающих устройств [1, 2].

Современный этап развития информационных систем характеризуется существенным ростом объемов передаваемых и обрабатываемых данных, которые различаются между собой назначением и форматом представления. В структуре информационных потоков доминирующее положение занимают мультимедийные данные, данные смешанного типа – текст, графика, видео и т.д. Следовательно, источники информации различного вида имеют нестационарный характер и могут изменять свои параметры в течение передачи не только информационного пакета, но и блока данных. Множество методов и алгоритмов сжатия настраиваются на фиксированные значения или, в лучшем случае, узкие области изменения параметров информационного источника (например, вероятностей появления двоичных слов). В случае же изменения характеристик источника или выхода их значений за predeterminedные граничные значения такие методы сжатия теряют свою эффективность [2, 3].

ПОСТАНОВКА ЗАДАЧИ

Достойное место среди методов сжимающего кодирования занимает адресно-векторный метод, отличающийся высокой скоростью преобразования данных и простотой реализации [4, 5, 6]. Адресно-векторное кодирование заключается в переходе от векторного к адресному методу кодирования в зависимости от числа k двоичных единиц в n -разрядном слове. Такой переход осуществляется согласно системе неравенств: