УДК 681.3:800.92

АКТУАЛЬНЫЕ ВОПРОСЫ РАЗРАБОТКИ ИНТЕРНЕТ-МАГАЗИНА С ПРИМЕНЕНИЕМ OST-ТЕХНОЛОГИЙ

Р.А.КУКАНОВА, Г.Ж.ЭРКИНБАЕВА
E.mail. ksucta@elcat.kg

Бул статьяда интернет-дүкөндү түзүүнүн актуалдуу суроолору JPEG жана GIF сүрөттөлүштөрүн колдонуп, Java программасынын мүмкүнчүлүктөрүнүн жардамы менен сайттын сапатын жана катышуучулугун жогорулатуу каралган.

В статье рассмотрены актуальные вопросы создания Интернет-магазина, повышения посещаемости и качества сайта посредством изображений JPEG и GIF на Java.

In this article we have considered and wrote about the pressing questions of creation of Internet shop, increase of attendance and quality of a site by means of images JPEG and GIF on Java.

Люди порой устают от утомительных поездок по магазинам, не любят стоять в очередях. В случае покупки товара через Интернет-магазин товар доставят курьеры, либо можно получить его на ближайшей почте. В результате есть возможность сэкономить время, которого в наш стремительный век всегда не хватает, и можно потратить его на себя и свою семью, можно уделить больше внимания родным и близким...

У Интернет-магазинов есть несколько важных преимуществ перед обычными магазинами. И именно эти преимущества позволяют им постоянно увеличивать долю продаж и привлекать ежедневно тысячи новых покупателей по всему миру. Так в чем преимущества Интернет-магазинов? Это доступность, анонимность, огромный ассортимент, экономия времени, свобода выбора, низкая цена, доставка, полная информация о товаре. В то же время существуют недостатки, такие как отсутствие возможности рассмотреть товар и подержать его в руках, нет возможности задать вопрос продавцу, совершив покупку в Интернет-магазине, приходится ждать доставки товара, отсутствие магазина как такового, потому что некоторые покупатели реальный магазин считают некой гарантией защиты своих прав, необходимость общения с оператором магазина, хоть это и не является недостатком для большинства, обычно, после заказа на сайте, перезванивает оператор и уточняет время и место доставки. Иногда это доставляет большое неудобство.

Таким образом, у Интернет-магазинов имеются свои достоинства и свои недостатки перед магазинами в офлайне. Поэтому в настоящее время сосуществуют обе этих формы торговли. Причем во многих торговых точках сочетаются оба метода продаж. Выплывает на поверхность вопрос о конкурентоспособности. В современном мире сохранять высокую конкурентоспособность без представительства в Интернете очень трудно. Основная задача Интернет-продвижения – это увеличить популярность сайта, количество посещений, а также повысить продажи услуг или товаров через сайт. Прежде всего, основой хорошего положения сайта в рейтинге поисковиков являются его изначальная структурная продуманность, информативность и удобство, а также первичная оптимизация – наличие в коде сайта специальных меток, по которым его могут найти и выделить поисковые роботы.

Хорошо продуманный и удобный для посетителя сайт уже сам по себе является гарантией, что его найдут и клиенты им заинтересуются. Если стоит конкретная задача – добиться определенного количества посещений сайта, увеличить число активных клиентов

или вывести сайт в топы поисковых рейтингов – этого можно добиться с помощью средств Интернет-продвижения.

Интернет-магазин требует особенного подхода в продвижении. Как правило, такие сайты отличаются огромным объемом страниц, так как их количество будет соответствовать продающимся в нем товарам. Это накладывает определенные требования на компанию, занимающуюся продвижением Интернет-магазинов, ведь такое продвижение требует совместной работы большого количества разнопрофильных специалистов – оптимизаторов, дизайнеров, программистов. Уже при создании дизайна и программной части специалисты компании задумываются об удобстве дальнейшего продвижения Интернет-магазина, так как если о его существовании никто не узнает, то и покупателей на сайте не будет. Еще одна отличительная особенность продвижения Интернет-магазина – то, что результат необходим как можно скорее, для этого следует применять комплексные методы Интернет-маркетинга, в некоторых случаях это совмещение продвижения с контекстной рекламой или размещением баннеров работы, или использовать графическую информацию при помощи выгрузки базы товаров в Яндекс.Маркет и другие эффективные методы привлечения в Интернет-магазин целевых посетителей уже в первые недели его работы.

Продолжая развивать тему о развитии Интернет-магазина, мы хотим посвятить данную статью некоторым особенностям внутренней оптимизации. Дело в том, что, чем больше углубляешься в изучение вопроса, тем больше обнаруживаешь нюансов, которыми возникает желание поделиться.

Внутренняя оптимизация, особенно Интернет-магазина, это огромная работа. Поэтому, даже делая внутреннюю оптимизацию поэтапно, можно многие детали упустить.

Раздельная карта сайта. Данный подход к использованию и настройке карты сайта подходит ко всем магазинам. Однако упомянуть этот подход будет нелишним. Дело в том, что, помимо определенного количества товаров, сайт Интернет-магазина имеет определенное количество контентной составляющей в виде статей, помимо контента в виде описания товаров. Добавляют seo оптимизированные статьи под определенные запросы и оформляются такие статьи, как правило, в виде отдельного раздела на сайте или блога. Порой, со временем, таких статей на сайте Интернет-магазина собирается очень много. В итоге получается, что на сайте в сумме собирается много контента. Может быть несколько тысяч товаров и несколько сотен статей на сайте. Если добавить это все в одну карту сайта, то страницы будут сканироваться долго и медленно.

На наш субъективный взгляд, хорошим вариантом решения данной проблемы будет раздельная карта сайта. А точнее, несколько карт сайта – для товаров Интернет-магазина Joomla VirtueMart и для контента в виде статей. Наверняка будет проще использовать именно такой. Необходимо просто создать несколько карт сайта и добавить их в панелях веб-мастеров поисковых систем.

Оптимизация страниц товаров. При создании семантического ядра наверняка можно обратить внимание на тот факт, что доля запросов может составлять до 80 % от общей массы. Таким образом, запросы становятся конвертабельными. Однако относительно внутренней оптимизации Интернет-магазина приятно то, что практически каждый конкретный товар (его описание) можно оптимизировать под конкретные тематические запросы из семантического ядра сайта.

Уникальный контент каждого товара. Уникальный контент для товаров можно брать непосредственно из семантического ядра, присматриваясь к запросам. А в помощь у нас есть возможность делать комментарии. Каждый товар Интернет-магазина имеет такую возможность. Даже имея небольшие однотипные описания товаров, существует возможность дополнять и уникализировать контент каждой страницы товара комментариями, даже если их пишут не только посетители. А изображения являются неотъемлемой частью интерфейса. Соответственно, мы хотим рассмотреть работу Java с изображениями JPEG и GIF. JPEG

лучше подходит для естественных цветных изображений, таких, как фотографии, а формат GIF является наилучшими для графических эмблем, изображений кнопок и т.п.

Сначала мы загрузим изображение с помощью очень короткой программы. Затем мы будем использовать классы, которые управляют загрузкой одного или нескольких изображений. Кроме того, существует набор абстрактных классов, которые помогают создать поток изображений, и фильтры, позволяющие обращаться к отдельным элементам изображений и модифицировать их.

Загрузка в страницу одиночного изображения будет выглядеть следующим образом:

```
/* <title>SimpleImageLoad</title>
*<applet code="SimpleImageLoad" width=300 height=150>
* <param name="img" value="mupk.gif">
* </applet>
*/
import java.applet.*;
import java.awt.*;
public class SimpleImageLoad extends Applet {
Image art;
public void init() {
art = getImage(getDocumentBase(), getParameter("img"));
}
public void paint(Graphics g) {
g.drawImage(art, 0, 0, this);
} }
```

Метод paint использует drawlmage с четырьмя аргументами: это ссылка на изображение art, координаты левого верхнего угла рисунка x, y и объект типа ImageObserver. Здесь мы использовали this в качестве имени ImageObserver, поскольку он встроен в апплет. Когда этот апплет запускается, он в методе init начинает загрузку art. Процесс загрузки изображения по сети хорошо заметен, поскольку встроенный интерфейс ImageObserver вызывает процедуру paint при каждом поступлении новой порции данных из сети. Можно использовать ImageObserver для отслеживания загрузки изображения, а в это время выводить на экран другую информацию.

ImageObserver — это абстрактный интерфейс, используемый для получения сообщения о создании изображения. Метод imageUpdate из ImageObserver – это все, что вы должны реализовать в своем апплете для его использования. В то время когда мы получаем информацию о загрузке, можно показывать любую понравившуюся мультипликацию, индикатор степени завершения загрузки или любую другую заставку. Для использования ImageObserver в своем подклассе Applet мы должны добавить в него строку implement ImageObserver, как показано в этом фрагменте программы:

```
public class MyApplet extends Applet implement ImageObserver {
```

Затем придется вставить в свой класс метод imageUpdate для интерфейса ImageObserver, как показано в следующем фрагменте:

```
public boolean imageUpdate(Image img, int status,
int x, int y int width, int height) {
if((status & ALLBITS) != 1) {
System.out.println("Still processing the image");
return true;
}
else {
System.out.println("Done processing the image");
return false;
} }
```

Метод imageUpdate вызывается с изображением Image, которое находится в процессе изменения, целым параметром status, отражающим состояние изменения, и с координатами прямоугольника (x, y, width, height), которые соответствуют различным величинам в зависимости от информационных флагов, перечисленных ниже. ImageUpdate должен возвращать false по окончании загрузки изображения и true, если изображение еще обрабатывается.

Целая переменная status поразрядно проверяется на наличие одного или нескольких флагов. Возможные флаги и информация, которую они несут, перечислены ниже.

Также рассмотрим программный пример, который использует ImageObserver для показа количества обработанных строк изображения и выводит эту информацию посредством переменной progress, на консоль:

```
/* <title>ObservedImageLoad</title>
* <applet code="ObservedImageLoad" width=290 height=140>
* <param name="img" value="mupk.gif">
* </applet>
*/
import java.applet.*;
import java.awt.*;
import java.awt.image.*;
public class ObservedImageLoad extends Applet
implements Runnable, ImageObserver {
Image art;
Dimension d;
int progress;
Thread motor;
boolean loaded;
public void init() {
art = getImage(getDocumentBase(), getParameter("img"));
loaded = false;
progress = 0;
}
public void paint(Graphics g) {
d = this.getSize();
loaded = g.drawImage(art, 0, 0, this);
}
public boolean imageUpdate(Image img, int info,
int x, int y, int width, int height) {
if((info & ALLBITS) != 1) {
if(progress<d.height) {
progress = progress + height;
}
System.out.println(progress + "/" + d.height);
return true;
}
else {
return false;
} }
public void start() {
motor = new Thread(this);
motor. start();
}
```

```
public void stop() {
motor.stop();
}
public void run() {
motor.setPriority(Thread.MIN_PRIORITY);
while(!loaded) { // update progress indicator (5 fps)
repaint();
try {
motor.sleep(200);
}
catch(InterruptedException e) {}
}
} }
```

Метод imageUpdate обрабатывает статус загрузки изображения. Информация о статусе передается через переменную info, с которой сравнивается статическая переменная ALLBITS. Если еще не получено все изображение, то мы добавляем величину heihgt к общему числу обработанных строк изображения. Для проверки этой концепции мы выводим количество обработанных строк изображения на консоль. Метод run перерисовывает апплет пять раз в секунду (каждые 200 миллисекунд) до тех пор, пока изображение art не загрузится. То, как долго монитор статуса загрузки будет работать, зависит от скорости передачи данных изображения по сети.

MediaTracker – это класс, предоставляющий удобный интерфейс для контроля статуса нескольких изображений. В следующих версиях этот класс будет контролировать другие мультимедийные форматы, такие, как звуковые файлы. Для использования MediaTracker нужно создать новый объект этого класса и использовать метод addImage для контроля статуса загрузки. При загрузке группы изображений будем использовать MediaTracker. Пока все изображения, которые интересуют нас, не загружены, пользователя будет развлекать демонстрационный экран.

ImageProducer – это абстрактный интерфейс для объектов, которые готовят данные для Image. Объект, который реализует интерфейс ImageProducer, будет предоставлять массивы целых или байтовых переменных, представляющих собой данные изображений. Рассмотрим класс MemoryImageSource, реализующий ImageProducer. Мы создадим новый объект Image из данных, которые сгенерировал ImageProducer.

MemoryImageSource – класс, используемый для создания нового изображения из массива пикселей. Вот конструктор, используемый для создания объекта MemoryImageSource:

MemoryImageSource(int width, int height, int pixel[], int offset, int scanLineWidth)

Объект MemoryImageSource собирается из массива целых величин pixel[] в используемой по умолчанию модели цветов RGB для генерации данных объекта Image. В используемой по умолчанию цветовой модели пиксель – это целая величина, состоящая из Alpha, Red, Green и Blue (OxAARRGGBB). Величина Alpha обозначает степень прозрачности элемента изображения.

MemoryImageSource возвращает объект ImageProducer, который используется с createImage для получения изображения, пригодного к использованию. Приведенный ниже короткий пример создает MemoryImageSource, используя вариант простого алгоритма (побитовое исключающее ИЛИ значений x и y координат каждого элемента изображения) из книги Gerard *J. Holzmann именованной “Beyond Photography, The Digital Darkroom”*.

```
/* <title>Memory Image Generator</title>
* <applet code="MemoryImager" width=256 height=256>
* </applet>
*/
```

```
import java.applet.*;
import java.awt.*;
import java.awt.image.*;
public class MemoryImager extends Applet {
Image art;
Dimension d;
public void init() {
generateImage();
}
public void generateImage() {
int pixels[] = new int[d.width * d.height];
int i = 0;
int r, g, b;
for(int y=0; y<h; y++) {
for(int x=0; x<h; x++) {
r = (x^y)&0xff; // red is x XOR y
g = (x*2^y*2)&0xff; //green is 2x XOR 2y
b = (x*4^y*4)&0xff; // blue is 4x XOR 4y
pixels[i++] = (255 << 24) | (r << 16) | (g << 8) | b;
} }
art = createImage(new MemoryImageSource(d.width, d.height, pixels, 0, d.width));
}
public void paint(Graphics g) {
g.drawImage(art, 0, 0, this);
} }
```

Подклассы классов ImageFilter и ImageFilterSource используются совместно для создания новых изображений фильтрованием уже существующих. Подкласс CropImageFilter создает новое изображение из фрагмента существующего. Использование этого фильтра полезно тогда, когда нужно использовать несколько маленьких изображений в одном апплете. Загрузка по сети двадцати изображений по 2 Кбайта происходит намного медленнее, чем загрузка одного файла размером 40 Кбайт. Если изображения одинакового размера, есть возможность собрать их в единый блок и использовать CropImageFilter для разделения блока на отдельные изображения в Java-клиенте.

RGBImageFilter используется для получения данных о каждом пикселе изображения, которые мы можем модифицировать, и таким образом модифицировать изображение.

Существующая система обработки изображений в Java пока не полностью поддерживает потребительские стандарты из-за ограниченной переносимости в сегодняшнем многообразии компьютерных платформ. Но в Java нет никаких “врожденных” ограничений на разработку мультимедийных приложений. Мы надеемся стать свидетелями больших успехов в развитии и совершенствовании этой технологии в течение ближайших лет для различных платформ в том числе.

Этот перечисленный комплекс мероприятий поможет приблизить Интернет-магазин ближе к ТОПу по многим продающим запросам. И что самое главное – можно это сделать бесплатно и самостоятельно, хотя это большой труд, требующий много времени.

Список литературы

1. http://kleparj.com/seo/19-osobennosti-vnutrennej-optimizatsii-internet-magazina-na-joomla-virtuemart.
2. http://pro-spo.ru/sites/3813-funkczionirovanie-internet-magazina--osnovnye-osobennosti.

3. Самое главное о... Интернет: А. Крупник. СПб.: Питер, 2004. – 128 с.
4. Баженова И.Ю. Язык программирования Java. – М.: ДИАЛОГ-МИФИ, 1997.