

понимается, что не существует такого алгоритма, обратного алгоритму сжатия, позволяющего точно восстановить исходное изображение.

Описанный алгоритм преобразования является общим для RAW формата и используется в каждом интерпретаторе. Однако алгоритмы, используемые на каждом из этапов могут отличаться, поэтому JPEG изображения полученные из одного RAW файла, но с использованием разных интерпретаторов может значительно отличаться друг от друга.

Список использованных источников:

6. Cambridgeincolour [Электронный ресурс]. – Режим доступа: <http://www.cambridgeincolour.com/ru/tutorials-ru/raw-file-format.htm>. Дата доступа: 30.03.2017.
7. <https://photo-monster.ru> [Электронный ресурс]. – Режим доступа: <https://photo-monster.ru/books/read/format-raw-iznutri.html>. Дата доступа: 01.04.2017.

ПЕРСОНАЛИЗАЦИЯ СОДЕРЖИМОГО ВЕБ СТРАНИЦ САЙТА ПОД УПРАВЛЕНИЕМ WCMSSITECORE

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Макаров Р.С.

Мельникова Е.В. – ассистент каф. ПОИТ

В настоящее время веб сайты крупных продуктовых компаний содержат большое количество информации, и эта цифра постоянно растет. С целью своевременного обновления информации на веб сайте были разработаны системы управления содержимым (contentmanagementsystems). Так как количество информации на сайте постоянно увеличивается, возникает проблема невозможности отображения всех существующих продуктов и проблема перегрузки страницы данными. Перегруженные данными страницы влекут за собой снижение посещаемости ресурса и, как следствие, снижение продаж. Для повышения вероятности покупок и вовлеченности пользователя, необходимо формировать список отображаемых товаров и услуг в соответствии с потребностями конкретных покупателей.

Для решения таких проблем следует задуматься о персонализации содержимого сайта под потребности пользователя. Существует два типа персонализации: явная и неявная. Неявной персонализацией является создание профиля потребителя на основе совокупности собранных данных из веб сайта и внешних систем, например: Youtube, Facebook. Под явной персонализацией понимают прямой опрос потребителя с целью выявления его предпочтений для формирования товаров и услуг, соответствующих его ожиданиям.

Для реализации неявной персонализации необходимо отслеживать действия пользователя на сайте с целью последующего анализа и применения математического моделирования для определения предпочтений пользователя. Далее необходимо отнести пользователя к одной из логических групп, тем самым сузив объем информации для отображения. Следующим шагом необходимо расширить и детализировать представление на основе истории покупок и интересов, добавив механизмы программ лояльности и скидок для конкретного индивида.

Графическое объяснение персонализации представлено на рисунке 1.

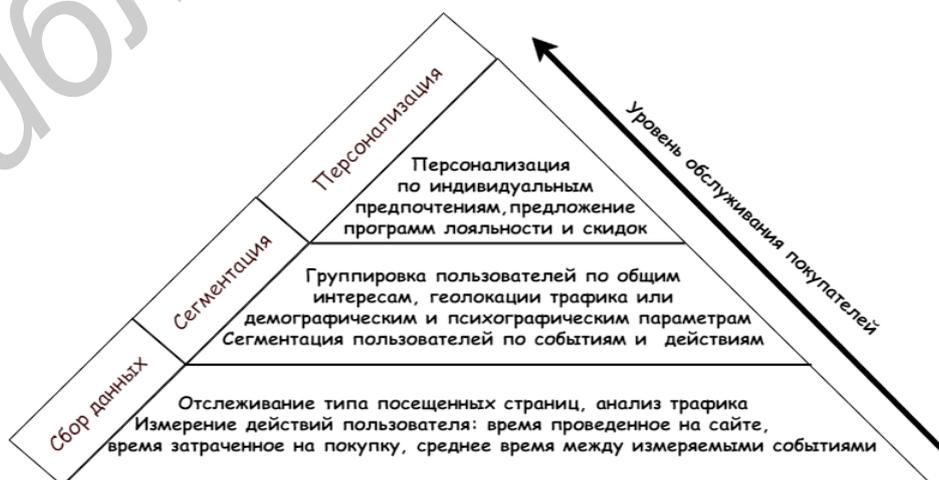


Рис.1 - Персонализация

Для повышения качества персонализации информации следует анализировать информацию о действиях пользователя во внешних системах по договоренности. Более глубокий анализ и больший набор данных поможет повысить качество и точность персонализации контента сайта под потребности конкретного пользователя, увеличив его вовлеченность и обеспечив положительноый опыт взаимодействия, и тем самым, повышая вероятность покупок.

Важно отметить, что сбор информации о пользователе должен быть обезличен, и эта информация не должна передаваться третьим лицам в соответствии с соглашением о неразглашении.

Список использованных источников:

1. Yogesh Patel, Sitecore Cookbook for Developers, Packt Publishing - ebooks Account ,April 29, 2016, 338p.
2. Dan Rousseau, Book of Wisdom: 5 Essentials for delivering personalized experiences.

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ И РАЗРАБОТКА АРХИТЕКТУРЫ ORM БИБЛИОТЕКИ ДЛЯ ПЛАТФОРМЫ JAVA

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Маркевич С.А.

Смолякова О.Г. – канд. техн. наук, доцент

Когда мы работаем с объектно-ориентированными системами, возникает несоответствие между объектной моделью и реляционной базой данных. СУРБД (система управления реляционными базами данных) представляет данные в табличном формате, тогда как объектно-ориентированные языки, такие как Java или C#, представляют их как взаимосвязанный граф объектов. При объединении данных моделей возникают проблемы, которые призваны решить ORM (объектно-реляционное представление) библиотеки.

ORM обозначает Object-RelationalMapping (объектно-реляционное представление) - это метод программирования для преобразования данных между реляционными базами данных и объектами объектно-ориентированного программирования, который создан для решения нескольких проблем. Первая проблема: простое изменение кода, при изменении структуры базы данных, в готовом приложении. Во-вторых, загрузка и хранение объектов в реляционной базе данных приводит к следующим проблемам несоответствия:

- 4) Гранулярность - иногда объектная модель имеет больше классов, чем число соответствующих таблиц в реляционной базе данных.
- 5) Наследование - СУРБД не определяют ничего похожего на наследование, которое является естественной парадигмой в объектно-ориентированных языках программирования.
- 6) Идентичность - СУРБД определяет ровно одно понятие «идентичности» - первичный ключ. Java, однако, определяет, как равенство объекта ($a == b$), так и идентичность объекта ($a.equals(b)$).
- 7) Ассоциации - объектно-ориентированные языки представляют собой ассоциации, использующие ссылки на объекты, тогда как СУРБД представляет связь как столбец внешнего ключа.
- 8) Навигация - способы доступа к объектам на Java и в СУБД принципиально различаются.

На рисунке 1 изображена упрощенная типовая схема архитектуры ORM библиотек:

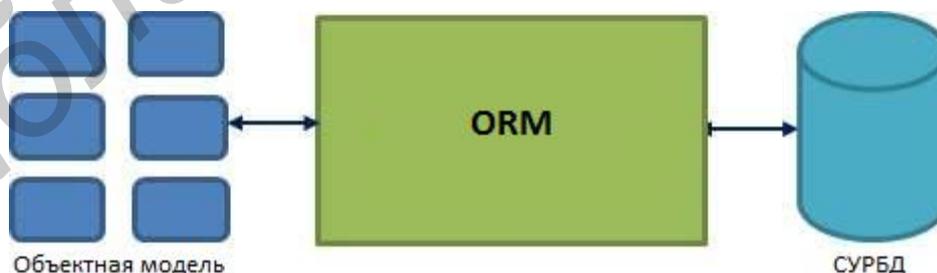


Рис. 1 – Типовая схема ORM библиотеки

На схеме видно, что ORM библиотека занимает своё место между объектной моделью и СУРБД, Работа с объектно-ориентированным программным обеспечением и реляционными базами данных может быть громоздкой и трудоемкой. Затраты на разработку значительно выше из-за несовпадения парадигмы между представлением данных в объектах и реляционными базами данных. Именно преимущества ORM библиотек позволяют решать обозначенные проблемы.

Система ORM имеет следующие преимущества перед ручной записью SQL команд:

- а) Позволяет работать с объектами при манипуляции с данными, а не с таблицы СУРБД;
- б) Скрывает детали SQL-запросов из программного кода;
- в) Нет необходимости иметь дело с реализацией базы данных;