

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:

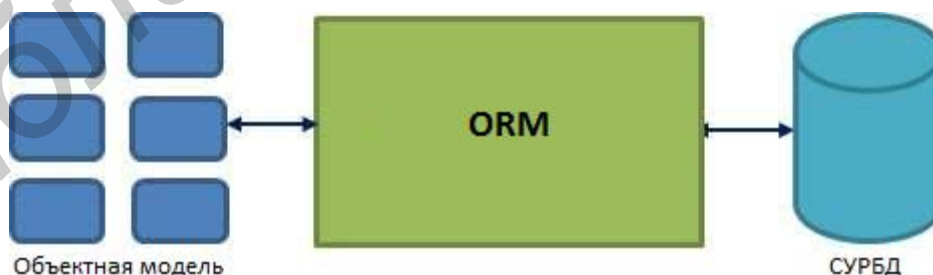


Рис. 1 – Типовая схема ORM библиотеки

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;

- г) Объекты основаны на бизнес-концепциях, а не на структуре базы данных;
- д) Управление транзакциями и автоматическое создание ключей;
- е) Позволяет быстрее разрабатывать приложения.

Для платформы Java существует несколько ORM библиотек. Целью данной работы является исследование эффективности известных решений, а также разработка и улучшение текущей архитектуры.

Список использованных источников:

1. Bauer, G. King, G. Gregory, Java Persistence with Hibernate. Second Edition. // Manning Publication Co, Shelter Island, NY 2016

ДЕКОМПОЗИЦИЯ СИГНАЛА НА ВНУТРЕННИЕ МОДОВЫЕ ФУНКЦИИ С ИСПОЛЬЗОВАНИЕМ ГЕТЕРОГЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

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

Мелещеня Д.В.

Бранцевич П.Ю. – канд. техн. наук, доцент

Преобразование Гильберта-Хуанга используется для исследования нестационарных и нелинейных систем. Ключевой частью этого метода анализа данных является декомпозиция исходного сигнала на внутренние эмпирические моды. В классическом методе, предложенном Хуангом для построения мод, используется интерполяция кубическими сплайнами кривых, полученных выделением максимумов и минимумов исходного сигнала. Данная процедура является достаточно трудоемкой, поэтому в данной работе предлагается использовать неоднородную вычислительную систему для сокращения времени расчета.

Анализ данных нашел свое применение как в теоретических исследованиях, так и на практике и помогает выявить ключевые характеристики сигналов, полученных в результате измерения физических величин или численного моделирования. Однако зачастую этим данным присущи следующие недостатки: данные нестационарные и представляют нелинейный процесс. Как следствие выбор метода анализа сильно ограничен. Один из подходов к изучению нелинейных нестационарных данных – преобразование Гильберта-Хуанга.

Преобразование Гильберта-Хуанга состоит из двух этапов: эмпирическая модовая декомпозиция (EMD), преобразование Гильберта полученной декомпозиции для вычисления мгновенных фазы и частоты (HSA).

Декомпозиция на модовые функции – итеративная процедура, ставящая в соответствие исходному сигналу набор эмпирических мод (IMF). В классическом определении данным Хуангом IMF собой представляет функцию, удовлетворяющую следующим двум условиям: на всем наборе данных количество пересечений оси абсцисс и экстремумов должно быть равно или отличаться на единицу; в любой точке среднее значение огибающих определяемых максимумами и минимумами должно быть равным нулю. Этим обеспечивается корректность, с физической точки зрения, результатов преобразования Гильберта.

Фактически алгоритм модовой декомпозиции, или просеивания, сводится к следующему. Строятся верхняя и нижняя огибающие, для нахождения которых выделяются локальные максимумы (для верхней) и минимумы (для нижней) на которых строятся интерполяционные сплайны, представляющие собой искомую функцию. Следующим этапом является нахождение разности между средним значением огибающих и исходным сигналом. Далее, если остаток удовлетворяет критерию остановки, он считается очередной модовой функцией, в противном случае разность принимают за исходный сигнал и алгоритм повторяется. После нахождения IMF, ее вычитают из исходного сигнала и, если разность не является монотонной функцией, то алгоритм просеивания повторяется.

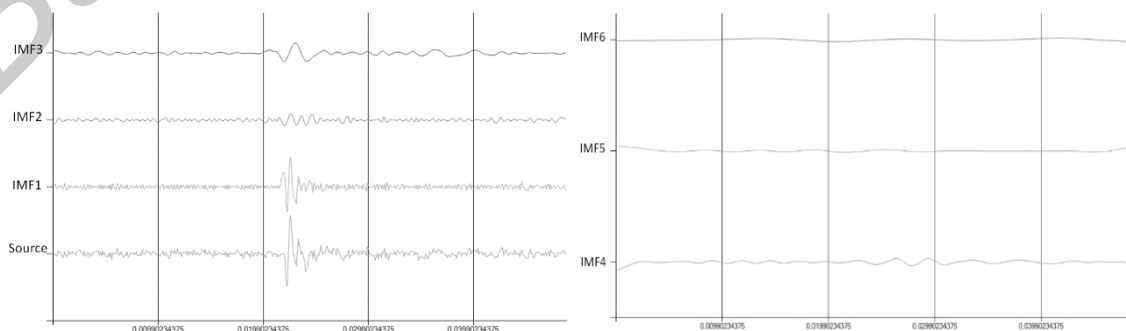


Рис. 1 – Пример разложения исходного сигнала на эмпирические модовые функции (IMF)