



Е.В. Симонова, Д.А. Новиков

СИНХРОНИЗАЦИЯ WEB-ПРИЛОЖЕНИЯ И БАЗЫ ДАННЫХ В МУЛЬТИАГЕНТНОЙ СИСТЕМЕ ЦЕЛЕВОГО ПЛАНИРОВАНИЯ КОСМИЧЕСКИХ АППАРАТОВ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ ЗЕМЛИ

(Самарский государственный аэрокосмический университет)

Введение

В связи с ростом числа использования веб-приложений для отображения данных и постепенным отказом от старых способов отображения информации, например, форм в C# и Swing в Java, проблема синхронизации современных веб-приложений и информации, хранящейся в базах данных (БД), становится все более актуальной.

Постановка задачи синхронизации web-приложения и БД в акторной системе

Имеется база данных, браузер клиента, клиент использует некоторое web-приложение. Необходимо показать, каким образом в окно web-приложения, открытое во вкладке браузера клиента, передаются сообщения об удалении, добавлении, изменении информации в БД. Взаимодействие между web-приложением и БД организовано через web-сервер. Архитектура системы представлена на рисунке 1.

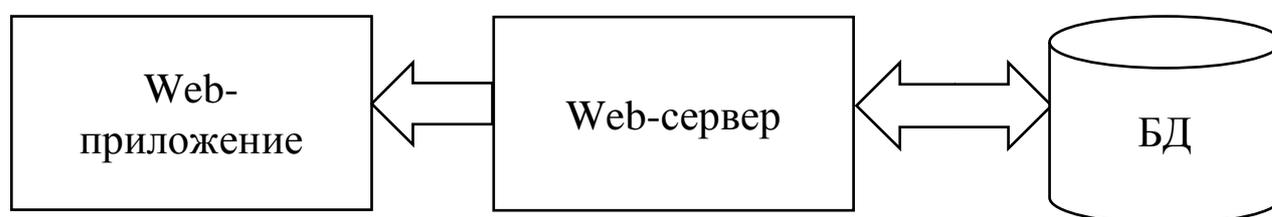


Рисунок 9 – Архитектура системы

Связь между web-приложением и web-сервером осуществляется посредством технологии WebSocket [1], которая идеально подходит для пересылки небольших сообщений. Web-сервер написан на языке Java и основан на фреймворке Akka [2], в основе которого лежит модель акторной системы.

Синхронизацию web-приложения и базы данных должен осуществлять web-сервер. Следовательно, все изменения в акторной системе, т.е. добавление или удаление объектов (акторов) и изменение внутренних состояний (свойств) акторов, должны сохраняться в БД и одновременно отсылааться web-приложению. Для базы данных определены следующие типы акторов: спутники, объекты наблюдения, пункты приема информации. Типы акторов описываются с помощью таблиц, строки которых соответствуют акторам, состояния акторов определяются значениями строк таблиц.



Описание акторов в акторной системе web-приложения

В процессе инициализации web-сервера загружаются следующие данные из БД:

- 1) список спутников;
- 2) список объектов наблюдения;
- 3) список пунктов приема информации.

Для каждого объекта из каждого списка создается уникальный актор с параметрами каждого объекта.

Так как акторная система web-приложения в целом достаточно сложна и объемна, в рамках решаемой задачи имеет смысл рассмотреть только следующие акторы:

- 1) DBActor – актор взаимодействия с БД, осуществляющий операции запроса, добавления, изменения и удаления данных непосредственно из БД;
- 2) WebActor – актор взаимодействия с web-приложением, осуществляющий отправку событий на web-сокеты (WebSocket).

Все акторы имеют доступ к единой шине событий (EventBus) и могут осуществлять ее прослушивание путем создания слушателя шины событий (EventBusListener), который регистрируется у шины событий посредством вызова метода Subscribe и передачи себя в качестве параметра. Акторы могут также генерировать произвольные события (Event) и отправлять их по шине событий.

Принцип взаимодействия акторов на стороне web-сервера

При создании нового объекта (актора) генерируется событие создания (CreateEvent) и отправляется по шине событий. Другие акторы, подписавшиеся на прослушивание шины событий, получают это событие. Акторы, получившие событие, должны определить, отвечают ли они за его обработку, путем сравнения типа события с типом, явно прописанным в них программистом, и при необходимости произвести действия, соответствующие каждому событию.

При изменении внутренних свойств актора происходит генерация события изменения (ChangeEvent) и отправка его на шину событий. Заинтересованные в этом событии акторы произведут соответствующие действия.

Принцип взаимодействия окна web-приложения и web-сервера

Каждое окно web-приложения, открытое в отдельной вкладке web-браузера, полностью независимо от других окон, открытых в соседних вкладках. Синхронизация осуществляется не между web-приложением в целом и web-сервером, а между окном web-приложения и web-сервером.

Каждое открытое окно создает своего слушателя и регистрирует его на прослушивание web-сокета, и подобно акторам в акторной системе, окна получают события и производят соответствующие им действия.

Решение задачи синхронизации

Например, был создан новый объект Спутник (SatteliteActor). При создании он сгенерировал событие (CreateSatteliteEvent) и отправил его на шину событий.



На прослушивание шины событий были подписаны DBActor и WebActor. Соответственно, каждому из них пришло событие добавления нового объекта. Актор DBActor по типу события произвел добавление новой строки в таблицу Спутники. В тот же момент, асинхронно с ним, актер WebActor сгенерировал новое событие и отправил его на web-сокеты. Окна web-приложения, которые подписались на прослушивание web-сокета, получили данное событие и произвели соответствующие действия.

В случае изменения свойств Спутника будет сгенерировано событие (ChangeSatteliteEvent), агенты DBActor и WebActor выполняют соответствующие действия – DBActor изменит записи в таблицах, а WebActor уведомит окна, подписавшиеся на это событие, об изменении свойств Спутника.

Заключение

Была решена проблема синхронизации данных между web-приложением и базой данных для мультиагентной системы целевого планирования космических аппаратов дистанционного зондирования Земли с помощью взаимодействия акторов.

Литература

1. WebSocket. – Режим доступа: <https://ru.wikipedia.org/wiki/WebSocket>
2. Akka. – Режим доступа: <http://akka.io/>

Г.А. Черезов¹, А.Н. Попов²

ПОСТРОЕНИЕ ПРИЗНАКОВОГО ПРОСТРАНСТВА ИЗ КОЭФФИЦИЕНТОВ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ

(¹Самарский государственный университет путей сообщения, ²Уральский государственный университет путей сообщения)

Для повышения достоверности диагностирования приемо-передающей аппаратуры тональных рельсовых цепей (ТРЦ), используемых в настоящее время на сети железных дорог России в качестве непрерывных датчиков контроля состояния участков пути, целесообразно использование вейвлет-преобразования [1, 2].

В общем случае вейвлет-преобразование имеет вид [3]

$$C_x(a, b) = \langle x(t), \psi_{a,b}(t) \rangle = |a|^{-1/2} \int_{-\infty}^{+\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt$$

или

$$x(t_i) = \sum_{k=1}^{2^{N-M}} a_{m,k} \varphi_{m,k}(t_i) + \sum_{m=1}^M \sum_{k=1}^{2^{N-M}} d_{m,k} \psi_{m,k}(t_i),$$

где M – число уровней разложения; $a_{m,k}$ и $d_{m,k}$ – коэффициенты разложения; $\varphi_{m,k}$ и $\psi_{m,k}$ – масштабирующие и вейвлет функции, определяемые