Федеральное государственное бюджетное образовательное учреждение высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**Реферат**

по дисциплине «Программная инженерия» на тему:

«Интеграционные шаблоны программной инженерии»

Подготовил

Студент группы ЗБ-ПИ20-1

Ерошкин К.В.

Преподаватель:

Будаев Е. С.

Москва 2024

Для начала стоит понять, что же такое интеграционные шаблоны в программной инженерии и какие подходы они собой представляют. Интеграционные шаблоны в программной инженерии представляют собой стандартизированные подходы и практики для объединения различных компонентов программной инженерии, которые будут объединены в единое целое.

Эти шаблоны облегчают решение общие проблемы интеграции, обеспечивая совместимость, надежность и масштабируемость систем.

Основные типы интеграционных шаблонов:

1. Шаблоны интеграции данных;
2. Шаблоны интеграции процессов;
3. Шаблоны интеграции интерфейсов;

Давайте рассмотрим более детально каждый из описанных выше шаблонов.

**Шаблоны интеграции данных**

1. ETL
   1. Extract(извлечение)

Этап на котором происходит извлечение данных, позволяет получать данные из различных источников, как пример: базы данных, файловые системы, получение данных через Api и т.д.

Задачи которые могут быть решены этим шаблоном:

* Подключение к источникам данных
* Получение данных в том формате, которых они предоставлены(исходные)
* Обработка исключений

Примеры источников данных:

* Базы данных(SQL сервер, MySQL и другие)
* Файлы(XML, CSV и другие)
* Веб-ресурсы и API
  1. Transform(преобразование)

На данном этапе мы можем преобразовывать извлеченные данные. Это может быть необходимо для анализа(машинное обучение) или для загрузки их в нашу целевую систему.

Включает в себя такие моменты:

* Очистка данных (позволит удалить дубликаты, заполнить пустые значения NaN)
* Изменение формата данных
* Обьединение данных из различных источников данных
  1. Load(загрузка)

На данном этапе мы перемещаем преобразованные данные в целевую систему. Загрузить такие данные мы можем, например в базу данных, аналитическую систему или в хранилище данных.

Основные цели на данном этапе:

* Поиск основных методов загрузки и целевых систем.
* Целостность данных при загрузке
* Оптимизация

1. CDC

Сейчас мы разберем метод, который позволяет отслеживать и фиксировать изменения, которые случаются в исходных данных, а так же синхронизировать их с другими системами в режиме реального времени.

* 1. Изменения

Метод, который мы рассматриваем позволяет отслеживать такие изменения как вставки, удаление и обновление данных в базе данных. Для реализации могут быть использованы различные способы, такие как:

* Триггеры
* Использование логов транзакции
* Снимки данных
  1. Репликация

Когда мы зафиксировали изменения, их нужно передать в систему. Все это может быть реализовано в режиме реального времени. Методы, с помощью которых все это реализуется:

* Потоковая передача данных (streaming)
* Пакетная передача данных (batch processing)
  1. Изменения

Для того чтобы данных в целевой системе соответствовали данных в исходных данных, нам необходимо применять изменения, реализовать это можно такими способами:

* Вставка новых записей
* Обновление уже существующих записей
* Удаление записей

1. Data Federation

Данная технология позволяет объединять данные из различных источников в режиме реального времени без объединения данных в одном хранилище физически.

* 1. Виртуализация данных

Для того чтобы получать данные из различных источников без их физического переноса в единое хранилище, данный способ позволяет создать единый интерфейс, который будет позволять иметь доступ к данным из различных источников. Способы для виртуализации данных:

* Создание виртуальных представлений или таблиц
* Определение схем для различных источников данных
  1. Объединение данных

Система, которую мы рассматриваем, позволяет объединять данные, обеспечивая их целостность, достигается это такими этапами как:

* Объединяющие запросы JOIN
* Управление различиями в схемах данных
  1. Доступ к данным

Для получения доступа к объединённым данным, мы можем использовать единый интерфейс, как пример sql запросы или использование api. Это обеспечивает:

* Единый источник данных
* Нет необходимости дублировать данные
* Повышение скорости доступа к данным

Каждая из этих технологий важна в управлении данными и позволяет различными способами работать с большими объёмами данных.

**Шаблоны интеграции процессов**

1. Business Process Management

Данная методология и набор технология позволяет управлять и оптимизировать бизнес-процессы. Данная методология охватывает весь цикл процесса.

* 1. Моделирование

Моделирование бизнес-процессов включает в себя создание визуальных представлений процессов, в свою очередь это позволяет понять текущие процессы и составить план по их улучшению. Задачи, которые предстоят на этом этапе:

* Определение шагов процесса
* Назначение ролей и ответственных лиц
* Определение последовательности действий

Инструменты моделирования:

* BPMN(Business Process Model and Notation) – это стандарт для графического представления процессов.
* ARIS, Microsoft Visio и т.д.
  1. Исполнение

После того как мы смоделировали процессы, они внедряются в работу при помощи BRM систем, в свою очередь эти системы автоматизируют выполнение задач и управляют их исполнением. Задачи, которые предстоят на этом этапе:

* Настройка бизнес-правил
* Настройка условий перехода между задачами
* Интеграция с другими системами
* Управление исполнением задач
  1. Мониторинг

В данном этапе мы отслеживаем выполнение задач, сбор данных, а также выявляем узкие места. Задачи, которые предстоят на этом этапе:

* Сбор метрик и показатели KPI
* Визуализация данных
* Уведомления о проблемах

Инструменты, которые позволяют мониторить:

* Дашборды
* Отчеты
* Интеграция с инструментами бизнес-аналитики

1. Workflow Automation

Данный процесс, включает в себя автоматизацию рабочих процессов. Основная цель – это улучшить эффективность и снизить вероятность появления ошибок.

* 1. Проектирование рабочих процессов

Данный этап включает в себя определением последовательности выполнения задач. Задачи, которые предстоят на этом этапе:

* Определение шагов рабочего процесса
* Определение условий для перехода между задачами
* Назначение участников и выдача им ролей
  1. Автоматизация

Данный процесс, включает в себя процессы по внедрению систем автоматизации, которые выполняют задачи и передачи их между участниками. Задачи, которые предстоят на этом этапе:

* Настройка автоматизация
* Настройка уведомлений
* Управление заданиями и контроль над их выполнением
* Интеграция других систем для передачи данных
  1. Мониторинг и управление

Данный процесс, включает в себя отслеживание и управление автоматизированными процессами, а также обработку исключений. Задачи, которые предстоят на этом этапе:

* Отслеживание состояния задач и процессов
* Обработка исключений
* Сбор данных для анализа
  1. Оптимизация

Данный процесс, включает в себя анализ данных, а также внесение улучшений для повышения эффективности. Задачи, которые предстоят на этом этапе:

* Анализ метрик производительности
* Внесение изменений в рабочие процессы
* Улучшение процессов на постоянной основе

Оба предоставленных инструмента позволяют повышать эффективность и оптимизацию бизнес-процессов. Они также помогают улучшить организациям упростить и улучшить свои бизнес-процессы.

**Шаблоны интеграции интерфейсов**

1. API Gateway

Сейчас мы рассмотрим варианты использования шлюза API, в разных случаях он может представлять собой либо сервер, либо службу, которая будет управлять доступом к микросервисам, но при этом обеспечивая единую точку входа для клиентов.

* 1. Маршрутизация

На данном этапе API Gateway принимает входящие запросы и направляет их к соответствующим микросервисам, что в свою очередь позволяет абстаригровать клиентов от сложной внутренней системы. Задачи, которые предстоят на данном этапе:

* Определение маршрутов
* Настройка правил маршрутов
* Балансировка нагрузки
  1. Агрегация

API Gateway позволяет агрегировать данные из различных микросервисов в одном ответе, что позволяет уменьшить количество запросов. Преимущества данной агрегации заключается в снижении латентности, улучшение производительности.

* 1. Преобразование

Данная система может изменять запросы и ответы, выполнять задачи аутентификации, авторизации, логирования, а также преобразовывать форматы данных. Преимущества заключается в централизованном управлении политикой безопасности, а также стандартизации данных.

* 1. Безопасность и мониторинг

Также данная система обеспечивает безопасность путем аутентификации и авторизации запросов, что в свою очередь предоставляет инструменты для мониторинга и логирования. Задачи, которые предстоят на этом этапе:

* Проверка подлинности пользователя
* Ограничение скорости запросов
* Сбор метрик и логов

1. Backend for Frontend

Сейчас мы поговорим о паттерне разработки, в котором создаются отдельные backend для различных пользовательских интерфейсов(пример, веб-приложение и мобильное приложение). Данный способ позволяет оптимизировать доступ к данным и функциональности из-за специфики каждого из интерфейсов.

* 1. Упрощение взаимодействий

BFF позволяет упростить взаимодействие с микросервисами, не показывая всю сложность внутренней архитекруты. Преимущества такого способа заключаются в снижении нагрузки на клиентское прилоежние, а также улучшении производительности и повышении уровня безопасности.

* 1. Разгрузка клиентской логики

Обработка и фильтрация данных, кэширование, а также управление сессими, все это позволяет перенести на серверную часть, тем самым разгрузив клиентскую.

* 1. Управление изменениями и версиями

BFF позволяет лучше управлять изменениями и версиями, так как при необходимости улучшить мобильное приложение, мы не будем затрагивать web версию. Преимущества такого способа заключается в снижении уровня ристка возникновения регрессии, повышении гибкости при обновлениях.

API Gateway и Backend for Frontend одни из ключевых шаблонов интеграции интерфейсов, позволяющие более гибко, безопасно взаимодействовать между клиентами и серверной частью.

**Заключение**

Интеграционные шаблоны программной инженерии, открывают огромные возможности для создания устойчивых, масштабируемых систем, а также их гибкость. При правильном подходе и использовании позволяют существенно улучшить качество и эффективность разработки на всех ее этапах. Правильных подбор шаблонов является ключевым фактором успешной разработки современных информационных систем.