УДК 004

Плотников С.Б., Апальков П.Ю.

*1. профессор, д.-.н., кафедра ВУЗ, mail@mail.ru*

*2. доцент, к.-.н., кафедра ВУЗ, mail@mail.ru*

ТЕХНОЛОГИЯ КОНФИГУРАЦИОННОГО МОДЕЛИРОВАНИЯ И ЕЕ ПРИМЕНЕНИЕ ДЛЯ МУЛЬТИМОДУЛЬНЫХ ПРОГРАММНЫХ СИСТЕМ

В данной статье рассматривается понятие мультимодульной программной системы, её преимущества над классической «монолитной» архитектуры проектирования, а также рассмотрено понятие модуля и описан способы управления его конфигурацией.

***Ключевые слова:*** модуль, проектирование программного обеспечения, конфигурационное моделирование, сервисы.

**Введение**

Начиная с 80-х годов прошлого столетия началось активное использование языков программирования для разработки промышленных приложений. Очень долгое время при их создании использовали монолитную архитектуру проектирования, согласно которой вся система являлась большим набором взаимосвязанных элементов. Но в современном мире, когда такие системы должны выполнять все больше и больше различных функций, такой подход сильно усложняет их, в следствие чего появляется все больше багов и время на разработку новых требований так же увеличивается.

С недавнего времени для решения данной проблемы многие предприятия стали использовать мультимодульный принцип проектирования программных систем. В его основу заложено то, что каждая логически обособленная часть функционала выносится в отдельный независимый модуль. В итоге вся система строится на множестве независимых модулей, что делает ее легко расширяемой и легко управляемой.

В рамках данной статьи будет рассмотрен принцип мультимодульного проектирование программных систем, а затем, на основе применения конфигурационного моделирования, будет описан способ управления настройками (конфигурациями) каждого модуля.

**1. Сравнительный анализ мультимодульной и классической архитектуры разработки ПО**

«Мультимодульная» архитектура проектирования, которую так же называют «микросервисной», является относительно новым термином в области разработки программного обеспечения. К сожалению, довольно часто люди, использующие его, не имеют конечного понимания того, что же данная архитектура из себя представляет.

Таким образом, в первую очередь определим, какие особенности имеет мультимодульная архитектура и сравним ее с классическим «монолитом», т.е. системы, работающей как единое целое.

При использовании монолитной архитектуры разработчики сталкиваются со следующими проблемами:

* необходимо ориентироваться в большом объеме кода;
* при добавлении нового функционала, его декомпозиция сильно ограничена, что приводит к большим временным, и, следовательно, финансовым затратам со стороны заказчика;
* при выходе из стоя какого либо компонента большой системы, вся система выходит из строя, что является неприемлемым для многих предприятий;
* сильная ограниченность в используемых технологиях;
* масштабирование может быть осуществлено только посредством копирования системы на отдельный сервер. Данная проблема так же обусловлена тем, что если необходимо увеличить производительность только одного элемента системы, т.к. остальные вычислительные мощности расходуются впустую.

На рисунке 1 представлена иллюстрация системы, имеющую монолитную архитектуру.

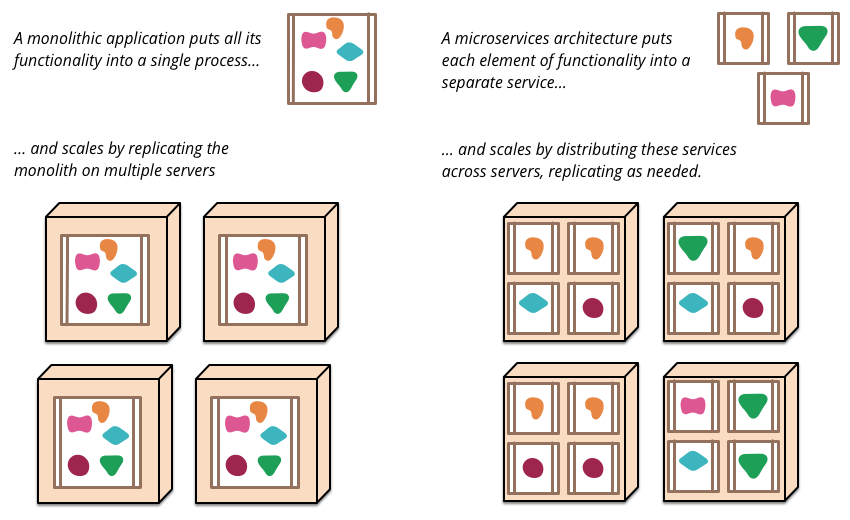


Рисунок 1 - Иллюстрация «монолитной» системы

Проектирование системы на основе независимых модулей по сути является решением вышеперечисленных проблем. Рассмотрим иллюстрацию построения системы с помощью мультимодульной архитектуры (рисунок 2).

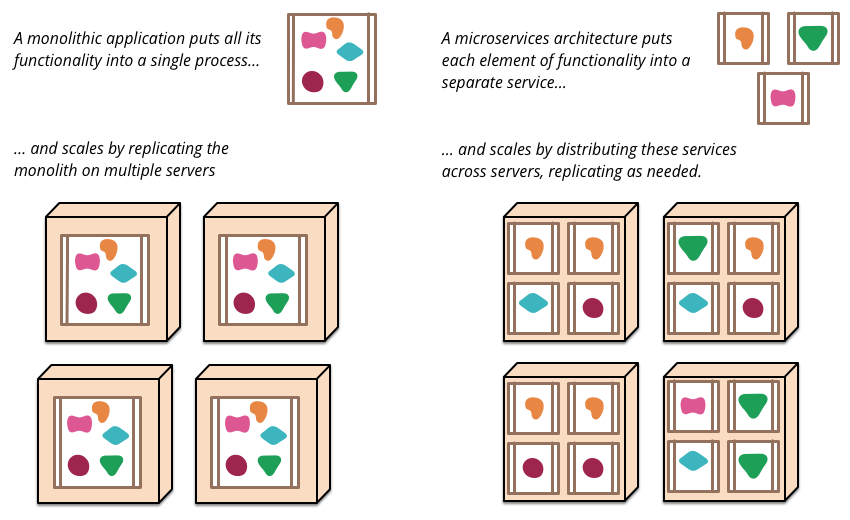


Рисунок 2 - Иллюстрация системы

на мульти модульной архитектуре

На ней видно, что каждый компонент системы обособлен и не зависит от всех остальных компонентов. Так же при необходимости увеличения производительности можно делать копию только того модуля, для которого это требуется.

Еще одним преимуществом такой архитектуры является простая декомпозиция задач при реализации новых требований, т.к. каждый модуль может разрабатываться не зависимо от готовности других модулей.

Таким образом, можно сделать вывод, что проектирование систем на основе модулей является предпочтительным решением.

**2. Описание стандартного модуля системы**

Рассмотрим более подробно свойства и состав стандартного модуля системы.

Каждый модуль должен выполнять определенную, свойственную только ему, логику, реализация которой представлена набором определенных методов. Для удобства использования данных методов, их объединяют в единый «сервис».

Сервис представляет собой совокупность методов, а так же интерфейса для работы с ними (рисунок 3). Данный интерфейс по сути является спецификацией сервиса, в которой указан идентификатор самого сервиса, а так же идентификатор каждого его метода.



Рисунок 3 - Схема классического сервиса

В процессе работы сервис может использовать параметры, которые можно отнести к так называемым настроечным параметрам, которые могут изменяться в зависимости от среды, на которой развернут модуль. Совокупность таких параметров определяет конфигурацию модуля.

Стоит отметить, что конфигурация не задаёт поведение работы модуля, а только лишь управляет значениями параметров, которыми оперирует программная логика модуля.

Конфигурацию модуля можно задать тремя способами:

* непосредственно в коде (дефолтная конфигурация);
* в ресурсном файле (файловая конфигурация)
* с помощью внешнего запроса (удаленная конфигурация)

Последний способ представляет особый интерес, так как таким образом можно менять значения параметров в режиме реального времени, соответственно не будет оказано никакого влияния на пользователей системы.

**3. Применение конфигурационного моделирования при проектировании модуля.**

Для реализации возможности удаленного управления значениями параметров модуля при его разработке необходимо использовать метод конфигурационного моделирования.

Конфигурационное моделирование предоставляет ряд методов, позволяющих смоделировать и спроектировать программный комплекс, содержащий в себе явный или же неявных механизм конфигурации. Ключевым понятием конфигурационного моделирования является понятие конфигуратора и адаптации модуля.

Конфигуратор представляет собой подсистему, способную решать задачу конфигурирования и реконфигурирование основной системы путем изменения значений некоторых ее параметров

Конфигуратор в свою очередь может быть реализован в рамках конфигурационного модуля. В этом случае саму систему можно считать автономной с точки зрения конфигурирования, либо в роли конфигуратора может выступать другая программная система, работающая в той же среде, что и основная. Во втором случае появляется задача создания канала управления, который позволял бы без потерь передавать указания конфигуратора к системе, а также сигналы от системы к конфигуратору.

Для того что бы конфигуратор мог взаимодействовать с конкретным модулем, сам модуль должен быть адаптирован, т.е. должен включать с себя определенную кодовую базу, принимающую запросы конфигуратора.

На рисунке 4 представлена упрощенная схема взаимодействия между пользователем, конфигуратором и модулем, где

* 1 - правка конфигурации пользователем;
* 2 - передача информации от конфигуратора к модулю;
* 3 - подтверждение обновления конфигурации от модуля;
* 4 – вывод результата на экран пользователя;



Рисунок 4 – Схема взаимодействия

«пользователь – конфигуратор - модуль»

**Заключение**

**ЛИТЕРАТУРА:**

1. ГОСТы
2. Электронные ресурсы

Полные – Фамилия Имя Отчество авторов

ПРАВИЛА ПОДГОТОВКИ И ОФОРМЛЕНИЯ СТАТЕЙ ДЛЯ ПУБЛИКАЦИИ В СБОРНИКЕ ТРУДОВ КОНФЕРЕНЦИИ (на англ)

Фамилия А.В., Фамилия А.В. на англ

Аннотация на англ.

***Keywords:*** те, что нужны статье на англ.