Проектирование и эксплуатация информационных систем в медиаиндустрии

Выломова Екатерина Алексеевна

e-mail: evylomova@gmail.com

0. Лекция 4

Модели проектирования систем:

- Каскадная
- Спиральная

Проектирование как конструирование

Шаблоны проектирования

- Типы шаблонов
- Описание шаблонов
- Шаблоны: абстрактная фабрика, сингтон, мост, наблюдатель, адаптер

Сложность алгоритмов:

- Сложность по времени и по памяти
- Асимптотические верхние и нижние границы
- Основные примеры сложности

Повторение

0. SoC

Separation of Concerns

“Разделение ответственности”

Процесс разделения компьютерной программы на функциональные блоки, как можно меньше дублирующие функции друг друга

- Снижение системной сложности
- Повышение надеждности и гибкости программ
- Обеспечение повторного использования

0. SoC

Шаблоны проектирования:

MVC – Model-View-Controller

Процедурное программирование:

Процедуры и функции

ОО-программирование:

Объекты

Аспектно-ориенторованное:

Аспекты

Сервис-ориентированная архитектура:

Разделение ответственности между сервисами

Принципы проектирования

I. Лекция 5. Открытые системы.Сервис-Ориентированная Архитектура

- Открытые системы
- Описание архитектуры
- SOAP и веб-сервисы
- WSDL
- Пример на Python

Основные понятия

I. Открытые системы

Свойство открытости

- Переносимость ПО
- Модифицируемость
- Комплексирование с другими системами

Открытость – выделение интерфейсной части

Основа ОС – стандартизация и унификация в области ИС

Аспекты открытости достигаются через стандартизацию

- API – Application Program Interface
- Межпрограммных интерфейсов
- Сетевого взаимодействия
- Пользовательского интерфейса
- Средств защиты информации
- Комплексирование с другими системами

***Профиль* открытой системы** - совокупность стандартов и других нормативных документов, обеспечивающих выполнение системой заданных функций.

Открытые системы

II. Сервис-Ориентированная архитектура

SOA - архитектурный подход к определению, связыванию и интеграции повторно используемых бизнес-сервисов, имеющих четкие границы и самодостаточных по своей функциональности.

Сервис-Ориентированная Архитектура

II. SOA: ключевые особенности

Ключевые особенности

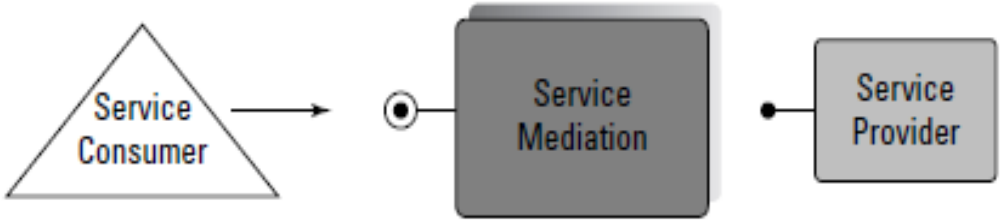
- Слабое связывание
- Повторное использование
- Расширяемость

Слабое связывание - виртуализация

II. SOA:сервис в SOA

Сервис - это функция, являющаяся четко определенной, самодостаточной и не зависящей от контекста или состояния других сервисов

Виртуализация сервиса



Виртуальный сервис является прокси-объектом для реального сервиса. Прокси-сервис представляет собой желаемый потребителем сервиса интерфейс. Потребители обращаются к прокси-сервису, передающему сообщения к действительному сервису.

Ключевые особенности

- Независимость местоположения
- Независимость передачи данных
- Независимость сообщений

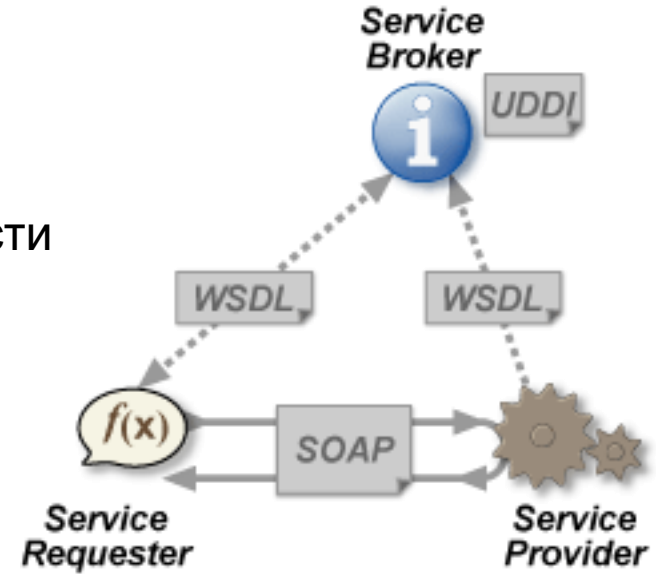
Сервис-Ориентированная Архитектура

II. SOA:сервис в SOA

Типовый функции виртуального сервиса:
Виртуальный сервис – наилучшее место реализации некоторых технических условий или обеспечения качества сервиса (QualityOfService).

- Проверка XML сообщений на корректность формата и соответствие интерфейсу сервиса.
- Аутентификация и авторизация: идентификация потребителя сервиса и проверка наличия у него прав для вызова сервиса
- Расшифровка сообщений и проверка подписи
- Балансировка нагрузки и гарантии наличия ресурсов для работы сервиса
- Маршрутизация сообщений. Передача сообщений различным реализациям сервиса в зависимости от содержимого сообщений или внешних условий.
- Мониторинг работы сервиса, производительности, а также проверка предоставления поставщикам требуемых услуг

III. Веб-служба



Web-service (веб-служба) – единица модульности при использовании сервис-ориентированной архитектуры.

Программная система, идентифицируемая строкой URI, чьи общедоступные интерфейсы определены на языке XML.

Плюсы:

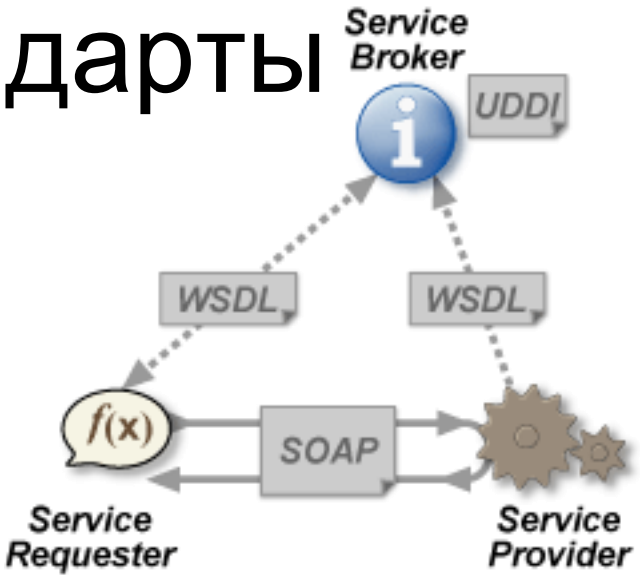
- Обеспечение взаимодействия программных систем независимо от платформы
- Использование открытых стандартов и протоколов. Простота разработки и отладки (XML)
- HTTP-взаимодействия программных систем через файрвол.

Минусы:

- Меньшая производительность и больший объем сетевого трафика

Веб-службы

III.Используемые стандарты

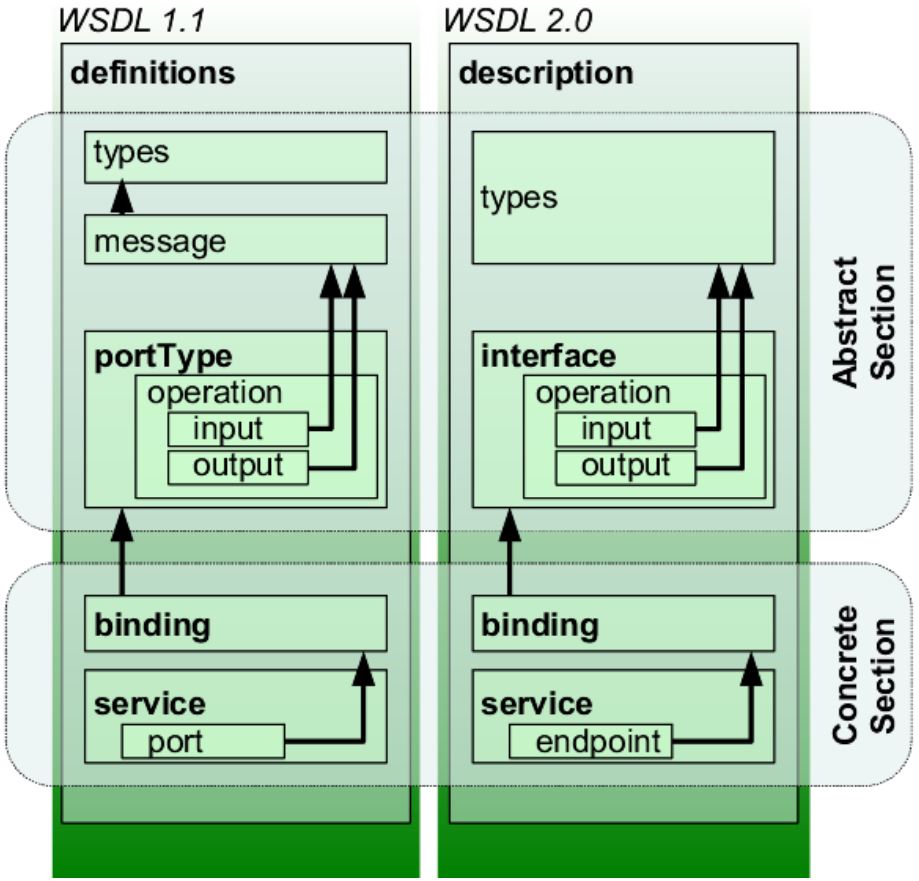


- **XML** – раширяемый язык разметки, предназначенный для хранения и передачи структурированных данных
- **WSDL** – язык описания внешних интерфйесов WS на базе XML
- **SOAP** – протокол обмена сообщениями на базе XML
- **UDDI** – универсальный интерфейс распознавания, описания и интеграции. Каталог веб-служб и сведения о компаниях, их предоставляющих

III. WSDL

Язык описания интерфейсов и доступа к ним

- types – описание типов данных (определение вида сообщений)
- message – элементы данных (сообщения, используемые WSом)
- portType – абстрактные операции (список операций, которые Могут быть выполнены над сообщениями)
- binding – связывание сервисов (способ, которым сообщение будет доставлено)



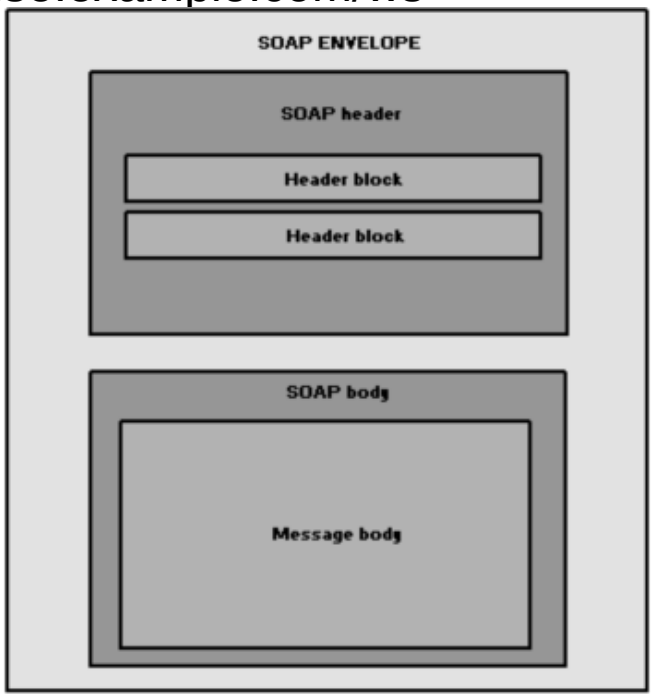
III. WSDL. Пример

```
<message name="getTermRequest">
        <part name="term" type="xs:string"/>
</message>
<message name="getTermResponse">
        <part name="value" type="xs:string"/>
</message>
<portType name="glossaryTerms">
        <operation name="getTerm">
                <input message="getTermRequest"/>
                 <output message="getTermResponse"/>
        </operation>
</portType>
```

III. SOAP

Запрос:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
        <getProductDetails xmlns="http://warehouse.example.com/ws">
                <productID>12345</productID>
        </getProductDetails>
</soap:Body>
 </soap:Envelope>
```



III. SOAP

Ответ:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
<getProductDetailsResult>
 <productID>12345</productID>
<productName>Стакан граненый</productName>
<description>Стакан граненый. 250 мл.</description>
 <price>9.95</price>
 <currency>
        <code>840</code>
        <alpha3>USD</alpha3>
        <sign>$</sign> <name>US dollar</name>
        <accuracy>2</accuracy>
 </currency>
<inStock>true</inStock>
 </getProductDetailsResult>
</getProductDetailsResponse>
</soap:Body>
</soap:Envelope>
```

IV. Пример

Создадим веб-сервис **Hello**

Серверная часть (hello):

- soaplib_handler.py
- urls.py
- views.py

См. Пример кода