

ПРОЕКТИРОВАНИЕ МОДУЛЬНОЙ АРХИТЕКТУРЫ ВЕБ-ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ OSGI

В докладе рассматривается подход к проектированию модульных приложений с использованием технологии OSGi. Анализируются достоинства и недостатки технологии OSGi, делается вывод о целесообразности ее применения.

ВВЕДЕНИЕ

Все программы подвержены неизбежным изменениям, как бы хорошо они не были спроектированы и разработаны. Однако, внесение изменений для большинства приложений означает запуск очередного цикла разработки, полного цикла тестирования, выполнение развертывания приложения. Этот процесс может занимать продолжительное время. Модульная структура приложения позволяет избежать подобных временных затрат.

I. МОДУЛЬНЫЙ ПОДХОД К РАЗРАБОТКЕ СИСТЕМ

Термин «модуль» или «компонент приложения» не является строго определенным. Модуль – это программный компонент с определенным интерфейсом и зависимостями, которые могут развертываться независимо и взаимодействовать с основной системой. Обычно, модуль обладает следующими свойствами:

- является независимым и заменяемым;
- инкапсулирует определенную функциональность.

Модульный подход к проектированию информационных систем позволяет создавать расширяемые системы, более удобные в поддержке. Модульность программного кода значительно уменьшает время перекомпиляции программы и снижает время тестирования, повышает гибкость системы, позволяет ей подстраиваться под изменяющиеся требования, развиваться.

II. ТЕХНОЛОГИЯ OSGI

Платформа Java никогда не поддерживала разработку модульных приложений. Разбиение на модули поддерживается только на уровне кода, не позволяет одновременно использовать различные версии одной библиотеки и динамически загружать модули в работающую систему. Технология OSGi добавляет эту функциональность в Java. Спецификация OSGi определяет

модель компонентов и сервисов Java. Компоненты и сервисы могут быть динамически загружены, активированы, деактивированы и удалены. Спецификация OSGi концептуально разделяет фреймворк на 3 уровня: уровень управления модулями системы, связанный с компоновкой, разделением и совместным использованием кода, уровень управления жизненным циклом модулей и уровень управления сервисами. Используя уровень управления сервисами уровень можно динамически зарегистрировать интерфейсы в реестре сервисов OSGi (SOA Registry). Другие компоненты могут потом динамически находить и использовать зарегистрированные сервисы [1]. Сервис в обязательном порядке предполагает контракт в виде интерфейса, потребителю не важно как и кто обеспечивает сервис, пока он соответствует контракту. Использование сервиса зачастую предполагает этапы поиска сервиса и установки соглашений между поставщиком и потребителем.

III. ВЫВОДЫ

Применение технологии OSGi позволяет достичь таких важных характеристик проекта как простота сопровождения, слабое сопряжение, возможность повторного использования компонентов [2]. Недостатками технологии являются строгая привязка приложения к OSGi и невозможность использования традиционных подходов к построению J2EE приложений. Применение рассматриваемой технологии целесообразно в крупных проектах с длительным жизненным циклом.

Список литературы

1. Документация OSGi [Электронный ресурс] – Режим доступа: <http://www.osgi.org/Specifications/HomePage>. – Дата доступа: 05.03.2014.
2. Макконнелл, С. Совершенный код. Мастер-класс / Макконнелл С. // Издательство: Русская редакция, 2013. – 896 с.

Грученков Владимир Валерьевич, магистрант кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, vladimir.gruchencov@gmail.com.

Научный руководитель: Никольшин Борис Викторович, доцент кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники, кандидат технических наук, доцент nik@bsuir.by.