

На правах рукописи

ГРИГОРЬЕВА Анастасия Викторовна

ИНСТРУМЕНТЫ АСПЕКТНО-ОРИЕНТИРОВАННОГО
ПРОГРАММИРОВАНИЯ ОБЛАЧНЫХ ВЕБ-ПРИЛОЖЕНИЙ
В MICROSOFT AZURE: РАЗРАБОТКА И РЕАЛИЗАЦИЯ
В СИСТЕМЕ ASPECT.NET

05.13.11 — Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Санкт-Петербург

2016

Работа выполнена на кафедре информатики федерального государственного бюджетного образовательного учреждения высшего образования «Санкт-Петербургский государственный университет».

Научный руководитель: ТУЛУПЬЕВ Александр Львович
доктор физико-математических наук, доцент,
федеральное государственное бюджетное
образовательное учреждение высшего образования «Санкт-Петербургский государственный университет», кафедра информатики, профессор

Официальные оппоненты: МУСАЕВ Александр Азерович
доктор технических наук, профессор,
федеральное государственное бюджетное
образовательное учреждение высшего образования
«Санкт-Петербургский государственный технологический институт» (технический университет), факультет информационных технологий и управления, декан

БУРАКОВ Вадим Витальевич
доктор технических наук, доцент,
федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский государственный университет аэрокосмического приборостроения» (ГУАП), кафедра компьютерных технологий и программной инженерии института вычислительных систем и программирования, профессор

Ведущая организация: федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики» (ИТМО)

Защита состоится **“22” сентября** 2016 г. в **17:00** часов на заседании диссертационного совета Д 212.232.51 на базе Санкт-Петербургского государственного университета по адресу: 198504. Санкт-Петербург. Старый Петергоф. Университетский пр., 28, математико-механический факультет, ауд. 405.

С диссертацией можно ознакомиться в Научной библиотеке им. М. Горького Санкт-Петербургского государственного университета по адресу: 199034, Санкт-Петербург, Университетская наб., 7-9 и на сайте <https://disser.spbu.ru/disser/soiskatelyu-uchjonoj-stepeni/dis-list/details/14/994.html>

Автореферат разослан “ ____ ” _____ 2016 г.

Ученый секретарь
диссертационного совета
Д 212.232.51, д.ф.-м.н., профессор



Демьянович Юрий Казимирович

Общая характеристика работы

Актуальность темы исследования. В работе рассматриваются два перспективных направления в области развития информационных технологий: облачные вычисления на платформе Microsoft Azure и аспектно-ориентированное программирование (АОП). Перспективность облачных приложений не вызывает сомнений: в настоящее время почти любая крупная компания предлагает свои облачные продукты от хранения личных файлов (Dropbox) и до офисных процессоров (Microsoft Office 360), от программной библиотеки распознавания образов (ABBYY Cloud OCR SDK) и до целых сред разработки (Microsoft Visual Studio Online).

В свою очередь, АОП применяется для выделения “сквозной функциональности” (распределенной по многим уровням объектно-ориентированной иерархии классов целевой системы) в один модуль — аспект. После такой операции улучшаются метрики качества исходного кода целевой системы, облегчается тестирование как аспекта, так и бизнес-логики.

Актуальность темы исследования обусловлена необходимостью повышения качества, производительности и облегчения сопровождения облачных веб-приложений. При этом актуальной является задача идентификации сквозной функциональности (авторизация, кэширование, протоколирование и пр.), которая встречается во многих веб-приложениях, а затем ее реализации: отделение от бизнес-логики в отдельные аспекты, пригодные для повторного использования. Таким образом, веб-приложение будет сфокусировано на выполнении задач предметной области, в то время как ряд нефункциональных требований переносится в сторонний проект — библиотеку аспектов. С учетом того, что для облачных веб-приложений уже существует библиотека Microsoft Enterprise Library Integration Pack, компоненты которой облегчают реализацию сквозной функциональности, также актуальной является реализация библиотеки аспектов с их помощью.

Степень разработанности темы исследования. Для аспектно-ориентированной разработки программ применяется множество инструментов, самыми популярными из которых являются AspectJ и PostSharp (по данным Google Trends). Однако анализ предметной области, проведенный в процессе написания работы, показал, что при изменении поведения программы с помощью современных АОП-инструментов для платформы Microsoft.NET отсутствует бесшовная интеграция аспектов и целевого исходного кода системы. Термин “бесшовная интеграция” можно раскрыть следующим образом: это такое расширение функциональности целевого кода, при котором не требуется вносить изменения в целевой проект, включая его код, файлы настроек и свойства проекта.

В качестве примера инструмента АОП, обеспечивающего бесшовное применение аспектов, в работе описывается и анализируется система Aspect.NET (среда аспектно-ориентированной разработки программ

для платформы Microsoft.NET), разработанная в лаборатории Java-технологии математико-механического факультета СПбГУ под научным руководством профессора В. О. Сафонова.

Впервые вопросы применения Aspect.NET к разработке веб-приложений исследовал в 2010 г. Нгуен В.Д., аспирант кафедры информатики математико-механического факультета СПбГУ. Результатом его работы стала библиотека с универсальными аспектами для безопасности, криптографии и расширения пользовательского Web-интерфейса приложений на базе ASP.NET. Проведенные им оценки производительности результирующей сборки после применения Aspect.NET показали, что накладные расходы на вызовы действий аспектов минимальны и соразмерны со вставкой их в исходный код вручную.

Объектом исследования являются инфраструктура, программные компоненты и сервисы платформы облачных вычислений Microsoft Azure, а также модели, методы и алгоритмы применения аспектов на базе инструмента Aspect.NET.

Предметом исследования является возможность улучшения качества исходного кода облачных веб-приложений с помощью АОП-рефакторинга. Кроме того, исследуется процесс разработки, реализации и применения универсальных аспектов, предназначенных для бесшовного расширения облачных веб-приложений, в том числе с использованием стороннего фреймворка Microsoft Enterprise Library Integration Pack.

Целью данной работы является повышение с помощью АОП качества и производительности, а также улучшение сопровождаемости облачных веб-приложений на платформе Microsoft Azure.

Достижение поставленной цели осуществляется решением следующих **задач**.

1. Разработать и реализовать библиотеку аспектов, расширяющую облачные веб-приложения следующими видами сквозной функциональности: перенаправление протоколирования в хранилище отладочной информации Microsoft Azure, кэширование информации, получаемой из SQL Azure, обработка исключений.

2. Предложить методику, позволяющую перехватывать обратные вызовы методов от инфраструктуры Microsoft Azure, не затрагивая при этом целевой код.

3. Найти способ публикации в облаке АОП-программ, разработанных с помощью Aspect.NET, а также обеспечить возможность их отладки в Microsoft Visual Studio.

4. Разработать метод бесшовного расширения функциональности облачных веб-приложений с помощью библиотеки Microsoft Enterprise Library и Aspect.NET.

5. Разработать методику аспектно-ориентированного рефакторинга для улучшения метрик исходного кода веб-приложения с помощью системы Aspect.NET.

Цель и задачи диссертации соответствуют области исследований паспорта специальности 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» – пункту 1 (модели, методы и алгоритмы проектирования и анализа программ и программных систем, их эквивалентных преобразований, верификации и тестирования), пункту 2 (языки программирования и системы программирования, семантика программ), пункту 3 (модели, методы, алгоритмы, языки и программные инструменты для организации взаимодействия программ и программных систем) и пункту 9 (модели, методы, алгоритмы и программная инфраструктура для организации глобально распределенной обработки данных).

Методология и методы исследования. В исследовании комбинировались два вида методологии. Первый, типичный для диссертационного исследования: обобщение, анализ и синтез теоретического и практического материала и второй, специфичный для темы исследования и задач практического программирования облачных веб-приложений на платформе Microsoft Azure:

- 1) сквозная функциональность из библиотеки Microsoft Enterprise Library (которая реализует в виде функциональных блоков практические методы решения наиболее актуальных задач в разработке программного обеспечения);
- 2) методы низкоуровневого программирования MSIL-сборок для расширения возможностей Aspect.NET;
- 3) метод программирования интерпретатора cmd.exe для описания событий пред- и пост-компиляции Microsoft Visual Studio;
- 4) методики аспектно-ориентированного программирования и рефакторинга.

Степень достоверности. Достоверность и обоснованность результатов подтверждается дополняющими друг друга, согласующимися экспериментальными данными, полученными автором при применении разработанных методик для целевых программных продуктов.

Обоснованность полученных результатов обеспечивается

- 1) корректным применением теории объектно-ориентированного и аспектно-ориентированного программирования;
- 2) результатами сравнительного анализа разработанных методов с аналогичными подходами к АОП;
- 3) результатом улучшения метрики интегрированной оценки качества исходного кода целевого приложения после проведения АОП-рефакторинга;

4) позитивным опытом внедрения и эксплуатации ПО, разработанного с применением предложенных методов.

Научная новизна работы заключается в том, что разработанная технология бесшовного расширения функциональности облачных веб-приложений на платформе Microsoft Azure с помощью подсистемы применения аспектов Aspect.NET предложена впервые. Разработанная автором библиотека аспектов AzureLibrary впервые позволила расширить функциональность облачных веб-приложений, не затрагивая их исходный код, конфигурационные файлы или настройки проекта.

Кроме того, разработанная в ходе исследования методика интеграции сквозной функциональности отличается от ранее созданных методик использования АОП своей бесшовностью применения. Это позволяет получать веб-приложения с высокой производительностью, которые легко сопровождать, и они не будут привязаны к конкретному АОП-инструменту, как это имеет место в альтернативных методиках АОП-программирования для платформы Microsoft .NET.

Таким образом, в данной работе рассмотрен принципиально новый подход к разработке облачных веб-приложений на платформе Microsoft Azure. Его преимуществами перед существующими технологиями являются: бесшовное добавление функциональности к веб-приложениям, высокая производительность, улучшение качества исходного кода и возможность использования существующих сторонних библиотек для реализации сквозной функциональности.

Теоретическая и практическая значимость работы. В рамках диссертационного исследования разработана методика для бесшовного расширения функциональности облачных веб-приложений на платформе Microsoft Azure с помощью АОП-инструмента Aspect.NET. Предложенный способ интеграции Aspect.NET в среду программирования Microsoft Visual Studio повышает удобство создания аспектов для платформы Microsoft.NET. Разработанный набор АОП-рефакторингов облачных ASP.NET веб-приложений демонстрирует повышение качества исходного кода целевого приложения за счет принципа “разделяй и властвуй”. Продемонстрированный в диссертации метод использования в аспектах стороннего решения Microsoft Enterprise Library Integration Pack дает возможность применять существующие библиотеки, компоненты и веб-сервисы для реализации сквозной функциональности, упрощая сами аспекты.

Практическая ценность заключается в реализации предложенных модификаций в Aspect.NET, создании библиотеки аспектов AzureLibrary и примеров ее применения. Также разработаны примеры, иллюстрирующие эффективность предложенного набора АОП-рефакторингов.

Так коэффициент сопровождаемости кода целевых приложений, подвергнутых АОП-рефакторингу улучшился на 3-25%. В проекте с Orchard CMS АОП-рефакторинг с помощью 6 аспектов повысил коэффициент сопро-

вождаемости целевых классов на 5-12%. Таким образом, доказан существенный технический эффект использования предложенных алгоритмов и методов при написании и сопровождении программных продуктов. Сведения о практическом использовании полученных автором диссертации научных результатов приведены в актах о внедрении программных принципов и библиотек, разработанных в рамках исследования (см. в приложении к диссертации).

Положения, выносимые на защиту

1. Разработана и реализована библиотека аспектов AzureLibrary для расширения облачных веб-приложений следующими видами функциональности: перенаправление протоколирования в хранилище отладочной информации Microsoft Azure, кэширование в распределенном облачном кэше информации, получаемой из SQL Azure, и обработки исключений. Применение AzureLibrary и Aspect.NET приводит к добавлению новой функциональности без какой-либо модификации целевого проекта.

2. Предложена методика “замещающего” аспектного наследника, который позволяет перехватывать обратные вызовы методов от инфраструктуры Microsoft Azure, поддерживающая возможность бесшовного применения аспектов в Aspect.NET. Также предложено отдельное хранение целевой и аспектной конфигурации, а для этого разработан алгоритм по их слиянию при компиляции аспектного проекта. Данные предложения реализованы в программном комплексе Aspect.NET.

3. Найден способ интеграции компоновщика аспектов Aspect.NET в Microsoft Visual Studio для обеспечения возможности публикации, а также локальной и удаленной отладки в облаке Azure АОП-программ.

4. Разработан метод бесшовной интеграции сквозной функциональности библиотеки Microsoft Enterprise Library Integration Pack for Azure в целевое приложение на примере аспектов логгирования (Logging Application Block), автоматического масштабирования облачного приложения (Autoscaling Application Block), реакции на исключительные ситуации (Transient Fault Handling). Это позволило расширить функциональность облачных приложений с помощью указанной библиотеки. Данные аспекты также реализованы автором в рамках библиотеки AzureLibrary.

5. Разработана и протестирована методика улучшения индекса сопровождаемости (Maintanability Index) целевых классов с помощью проведения их АОП-рефакторинга. Спроектирован набор из 5 аспектно-ориентированных рефакторингов на основе Aspect.NET для классов облачных веб-приложений. Проведенные эксперименты над типичными веб-приложениями (N2CMS и Orchard CMS) показали улучшение индекса сопровождаемости целевых классов на значения от 2 до 25%.

Все полученные результаты являются новыми.

Апробация результатов. Основные результаты и выводы диссертационной работы докладывались и обсуждались на следующих научных конфе-

ренциях и семинарах: 1) конференция “Технологии Microsoft в теории и практике программирования”, 2011, СПбГПУ, Санкт-Петербург; 2) вторая межвузовская научная конференция по проблемам информатики “СПИСОК-2011”, 2011, СПбГУ, Санкт-Петербург; 3) всероссийская научная конференция по проблемам информатики “СПИСОК-2012”, 2012, СПбГУ, Санкт-Петербург; 4) конференция “Технологии Microsoft в теории и практике программирования”, 2012, СПбГПУ, Санкт-Петербург; 5) межвузовский конкурс-конференция “Технологии Microsoft в теории и практике программирования”, 2013, СПбГПУ, Санкт-Петербург; 6) межвузовская научная конференция по проблемам информатики “СПИСОК-2013”, 2013, СПбГУ, Санкт-Петербург; 7) всероссийская научная конференция по проблемам информатики “СПИСОК-2014”, 2014, СПбГУ, Санкт-Петербург; 8) international conference on numerical analysis and applied mathematics 2014 (ICNAAM-2014), 2014, Rhodes, Greece; 9) всероссийская научная конференция по проблемам информатики “СПИСОК-2016”, 2016, СПбГУ, Санкт-Петербург.

Структура и объем диссертации. Диссертационная работа состоит из введения, шести глав, заключения, библиографического списка, приложения. Работа изложена на 150 страницах машинописного текста, содержит 7 рисунков, 3 таблицы, 58 листингов, 3 акта о внедрении и список литературы из 132 наименований.

Основное содержание работы

Во **введении** обоснована актуальность работы, сформулированы цели, задачи, объект и предмет диссертационного исследования, показана научная новизна и практическая ценность полученных результатов. Приводится структура диссертации.

В **первой главе** приведены основные понятия облачных технологий, описаны их виды и характеристики. Затем дается краткая история развития и предпосылки возникновения облачных вычислений, подробно рассмотрены типичные сценарии их использования и преимущества. На протяжении всей главы делается сравнительный анализ облачных технологий от разных поставщиков. Для ряда поставщиков рассмотрены предлагаемые ими облачные сервисы и линейка оборудования. При изложении особенностей Microsoft Azure делается акцент на принципы разработки облачных веб-приложений.

Во **второй главе** диссертации формулируется проблема рассредоточения сквозной функциональности и делается обзор существующих технологий ее реализации на примере паттернов объектно-ориентированного программирования, ЮС-контейнеров и АОП. Проведенный анализ показывает, что только АОП позволяет локализовать сквозную функциональность в одном

модуле (аспекте), причем аспект содержит как саму функциональность, так и правила ее внедрения в целевой код.

Далее приведена сравнительная характеристика статического (Рис. 1) и динамического методов применения аспектов. Для выбора АОП-инструмента, подходящего к целям диссертационного исследования, описаны требования к производительности и бесшовности процесса внедрения аспектов. Высокая производительность результирующих приложений необходима для уменьшения затрат при их эксплуатации в облаке, а бесшовная интеграция позволяет проводить АОП-рефакторинг и расширение функциональности без модификации целевого проекта. На основе анализа существующих АОП-инструментов для платформ Java и .NET сделан аргументированный выбор в пользу Aspect.NET, который позволяет разрабатывать аспекты для облачных веб-приложений и удовлетворяет всем озвученным критериям.

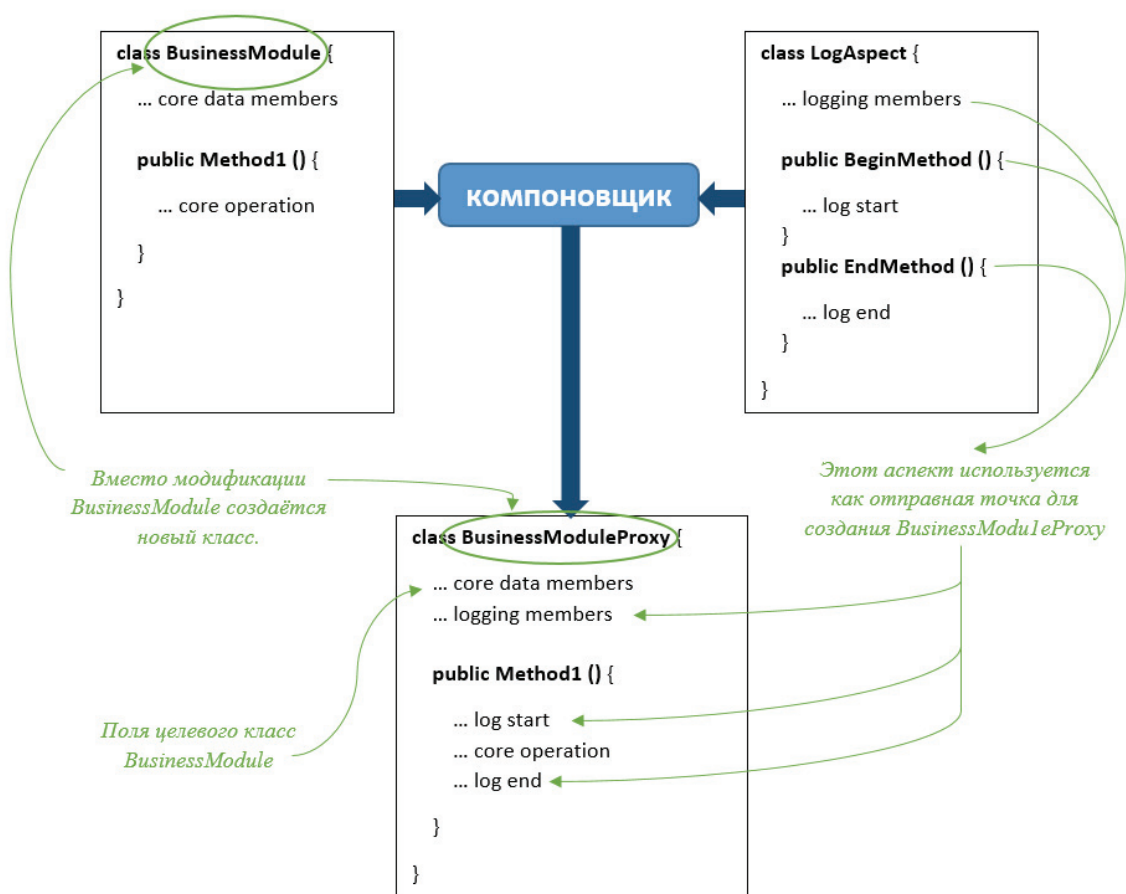


Рис. 1: Схема работы компоновщика аспектов

В третьей главе исследована возможность применения Aspect.NET для реализации облачных приложений на платформе Microsoft Azure. Показана необходимость и осуществлена интеграция Aspect.NET со средой разработки Microsoft Visual Studio через события пред- и пост-компиляции. Также была описана реализация в Aspect.NET возможности удаленной отладки результирующих приложений на облачных серверах. Далее в главе проанали-

зирована типичная сквозная функциональность облачных веб-приложений на базе ASP.NET MVC и реализовано расширение функциональных возможностей Aspect.NET для поддержки ее выделения в аспекты, а именно:

1) автоматическое слияние конфигурационных XML-файлов аспектного и целевых проектов при компиляции, что позволило сократить количество настроек в целевом проекте до минимального количества, необходимого для бизнес-логики;

2) перехват целевых методов обратного вызова, которые вызываются платформой с помощью автоматической замены целевого класса его аспектным наследником.

В последнем случае потребовалась реализация в Aspect.NET специального пользовательского атрибута [ReplaceBaseClass], предписывающего компоновщику заменить целевой класс своим аспектным наследником. Например, аспект, замещающий метод обратного вызова LogButton_Click в классе веб-страницы Default (который теперь содержит только бизнес-логику), может выглядеть как на листинге 1:

```
[ReplaceBaseClass]
public class AspectClass : Default
{
    protected void LogButton_Click(object sender, EventArgs e)
    {
        //Совершаем действия перед вызовом целевого метода...
        base.LogButton_Click(sender, e);
        //Совершаем действия после вызова целевого метода
    }
}
```

Листинг 1: Аспектный класс-наследник для обработки щелчка мышью

В четвертой главе ставилась задача создания универсальных аспектов, которые могут быть применены к различным облачным приложениям. Аспекты могут быть описаны как в программном коде, так и в файле конфигурации (обычно в web.config). Оба этих способа задействуются в аспекте, который перенаправляет отладочную информацию в облачное хранилище WAD: действия аспекта запускают диагностический монитор, а файл конфигурации аспекта регистрирует этот диагностический монитор в качестве приемника отладочной информации.

Далее был представлен аспект, который повышает надежность любого целевого метода путем определения стратегии реакции на его исключения. В качестве примера (листинг 2) взята стратегия “Предохранитель” (Circuit Breaker):

```

class SimpleCircuitBreakerAspect : Aspect {
    static DateTime ? errorTime = null;
    //Таймаут, в течение которого предохранитель будет закрыт
    static readonly int resetTimeoutInMilliseconds = 1000;

    [AspectAction("%instead %call *UnreliableService.GetContent")]
    public static string GetContentSafe() {
        //Если errorTime.HasValue, то предохранитель закрыт.
        if (errorTime.HasValue)
            //Если еще не истек таймаут с момента последнего неуспешного вызова
            if ((DateTime.Now - errorTime.Value).TotalMilliseconds <
                resetTimeoutInMilliseconds)
                return GetAnotherContent(); // Вызывается запасной метод

        try {
            //Проверяем доступность целевого метода
            string result = (TargetMemberInfo as
                MethodInfo).Invoke(TargetObject, null);
            //Исключения не было, открываем предохранитель
            errorTime = null;
            return result;
        }
        catch (Exception) {
            //Закрываем предохранитель, запомнив время сбоя.
            errorTime = DateTime.Now;
            //Вызываем запасной метод
            return GetAnotherContent();
        }
    }
    //Запасной вариант действий при сбое
    private static string GetAnotherContent() {
        return "Service is busy";
    }
}

```

Листинг 2: Аспект для реализации паттерна “Предохранитель”

Следующим был описан аспект, повышающий производительность целевого метода, через кэширование его результата средствами Windows Azure Caching.

Наконец, приведена XML-конфигурация аспекта, которая позволяет менять способ хранения веб-сессии. В рамках библиотеки AzureLibrary реализованы вышеописанные аспекты, что позволяет расширить этой же функциональностью другие веб-приложения.

В пятой главе сформулированы приемы для выделения из целевых классов веб-приложений зависимостей от служб Microsoft Azure в аспекты. На примере службы кэширования демонстрируется “очищение” целевого бизнес-класса от вторичной функции кэширования и выделения ее в %instead-действие аспекта, пригодного для повторного использования. В том случае, когда требуется подвергнуть АОП-рефакторингу метод (callback), вызываемый инфраструктурой Microsoft Azure, рекомендуется создать для него аспектный производный класс через атрибут [ReplaceBaseClass]. Однако это не поможет в случае, когда требуется переопределить целевой метод рабочей или веб-роли, так как Microsoft Azure требует для роли только прямого наследования от класса RoleEntryPoint. В этом случае целиком заме-

нить его не получится, но можно расширить его поведение, привязав `%before` или `%after`-действие аспекта к вызову методов `RoleEntryPoint`.

В ряде случаев сквозная функциональность требует отдельной панели настройки от пользователя в `.aspx`-разметке ASP.NET MVC приложения. Тогда в рамках АОП-рефакторинга она выделяется в отдельный аспект, содержащий `.ascx`-компонент представления указанной панели, связанную с моделью данных для сквозной функциональности, и класс целевого контроллера расширяется аспектным замещающим наследником для поддержки нового `.ascx`-компонента. Такой прием повышает повторную используемость сквозной функциональности, однако бывают случаи, когда в представлении есть лишь несколько разнородных полей, управляющих службой, которые не имеет смысла или затруднительно объединять в одну панель. В этом случае оптимальным будет решение выделить в классе страницы для них пустые `private`-свойства, а реализацию определить в аспекте с помощью `%instead`-действий.

Все приведенные приемы АОП-рефакторинга позволяют переложить обязанность работать со службами Microsoft Azure с целевого класса на аспект, что фокусирует его на реализации исключительно бизнес-требований. Конкретные преимущества демонстрируются повышением у целевых классов комбинированной метрики качества кода `Maintainability Index`. В заключение приведены примеры АОП-рефакторингов промышленных систем `N2CMS` и `Orchard CMS` и, также с помощью метрики `Maintainability Index`, продемонстрировано улучшение качества их исходного кода.

В **шестой главе** представлены основные принципы бесшовного расширения функциональности облачных веб-приложений с помощью `Aspect.NET` и сторонних библиотек. В результате разрывается зависимость между целевым приложением, применяемой АОП-системой и сторонней библиотекой, что позволяет заменить их в любой момент прозрачно для целевого приложения. При этом сторонняя библиотека предоставляет классы для реализации сквозной функциональности, а `Aspect.NET` средства для их интеграции в целевое приложение. В качестве сторонней библиотеки выбрана `Microsoft Enterprise Library Integration Pack`, функциональные блоки которой позволили бесшовно внедрить в веб-приложение сквозную функциональность протоколирования, автоматического масштабирования нагрузки и динамически конфигурируемой стратегии реакции на исключения. На листинге 3 представлено решение задачи вынесения в аспект протоколирования с помощью аспектного наследника:

```

//Проект с замещающим аспектным наследником
[AspectDotNet.ReplaceBaseClass]
public class AspectClass : Default {
    protected void LogButton_Click(object sender, EventArgs e) {
        Microsoft.Practices.EnterpriseLibrary.Logging.
            Logger.Write("Message from the Logging Application Block");
        base.LogButton_Click(sender, e);
    }
}
//Исходный проект, после отделения зависимости от Logging Application Block
public partial class Default : System.Web.UI.Page{
    protected void LogButton_Click(object sender, EventArgs e) {}
}

```

Листинг 3: Протоколирование, вынесеное в аспект

Для бесшовной интеграции сквозной функциональности в ряде случаев достаточно основных средств Aspect.NET, а именно замещающего аспектного наследника и %instead, %after, %before-действий. Однако, иногда замещающий аспектный наследник должен вызывать закрытый метод своего базового класса и, в этом случае, необходимо использовать рефлексияю .NET. В том случае, когда происходит несколько таких вызовов, то оптимальным будет сохранение ссылок на метаданные соответствующего закрытого метода в полях аспектного наследника и их инициализация в конструкторе.

В том случае, когда замещающий аспектный наследник должен иметь конструкторы отличные от базового класса (например, для инициализации зависимостей от Microsoft Enterprise Library), тогда необходимо %instead-действиями заменить и вызовы всех его конструкторов в целевом коде.

В результате для бесшовной интеграции Microsoft Enterprise Library в целевое веб-приложение достаточно создать новый проект с аспектом, установить Microsoft Enterprise Library в него с помощью менеджера расширений Nuget, написать код аспекта и в свойствах проекта задать события пред-компиляции для слияния XML-настроек и пост-компиляции для слияния с целевой сборкой.

Заключение

В результате диссертационного исследования были выполнены все поставленные задачи и, таким образом, достигнута цель работы. Библиотека AzureLibrary предоставляет готовые универсальные аспекты со сквозной функциональностью для облачных веб-приложений. Реализованный комплекс программ в Aspect.NET позволяет бесшовно расширять функциональность облачных веб-приложений любой сквозной функциональностью, в том числе и аспектами из AzureLibrary. Причем, аспекты не обязаны реализовывать всю сквозную функциональность и могут делегировать ее выполнение сторонним библиотекам, компонентам и веб-сервисам. Для улучшения качества исходного кода облачного веб-приложения и повышения степени его повторного использования, программист также получает возможность выделить некоторые действия в другой проект, воспользовавшись предложенным

набором АОП-рефакторингов, и дополнить его в зависимости от своего проекта.

Использовать сервисы платформы Microsoft Azure может любое веб-приложение, однако при этом возникает проблема “сквозной функциональности”, когда код обращения к соответствующим компонентам распределен по всей системе. Это приводит к размытию ответственности классов и нарушению принципов слабосвязанной архитектуры. При этом бизнес-логика перемешивается с нефункциональными требованиями, система теряет гибкость, и затрудняется ее сопровождение. Традиционно для улучшения дизайна системы программисты применяют рефакторинг, паттерны ООП и IoC-контейнеры, однако в случае сквозной функциональности эффективными оказываются только АОП-технологии.

В данной работе был произведен тщательный анализ и сравнение существующих современных облачных сервисов и платформ. Выявлено, что платформа Microsoft Azure является наиболее перспективной из них. Предложен и реализован в виде библиотек ряд идей, позволяющих сделать написание и сопровождение программного кода Microsoft Azure удобнее, качественнее, безопаснее и быстрее. Улучшение качественных характеристик подтверждено экспериментами и результатами практического применения разработанных методик к производственным программным продуктам.

Разработанные автором программные инструменты успешно протестированы, внедрены в работу предприятий и используются в настоящее время. Подтвержден существенный технический эффект использования предложенных алгоритмов и методов при написании и сопровождении программных продуктов.

Результаты, полученные в процессе выполнения проектов с использованием разработанной технологии, позволяют сделать вывод о работоспособности и эффективности использования системы Aspect.NET для бесшовной интеграции аспектов в рамках облачных вычислений на платформе Microsoft Azure.

Сформулированы **рекомендации по применению результатов** работы в индустрии и научных исследованиях. Они заключаются в необходимости учитывать следующие ограничения: в многопоточных приложениях имеет смысл маркировать все разделяемые данные аспекта атрибутом [ThreadStatic]; целевые методы должны обладать аргументами или результатом, типы которых не являются дженериками; целевые приложения должны быть разработаны в среде Microsoft Visual Studio на языке C# или Visual Basic.

Определены **перспективы дальнейшей разработки тематики**, основными из которых являются адаптация предложенных методик для таких сред разработки на платформе .NET, как MonoDevelop и Unity3D. Точная оценка эффективности АОП-рефактинга является открытой исследовательской задачей. Кроме этого, в целях повышения стабильности системы, представляется целесообразным исследовать процесс тестирования как аспектов, так и результирующей сборки в целом.

Публикации автора по теме диссертации

Результаты исследований отражены в работах 1–10. В статье 2 соискателю принадлежат рассуждения о применимости АОП к разработке облачных программ и реализация набора аспектов. Соавторам в этой статье принадлежит описание пользовательских требований и особенностей реализации геолокационной системы в облаке. В публикациях 3–10 соискателю принадлежат формулировки, рассуждения, описания и реализация в комплексе программ Aspect.NET метода бесшовной интеграции аспектов с Microsoft Enterprise Library Integration Pack, разработка и реализация набора АОП-рефакторингов, а также библиотеки аспектов AzureLibrary со сквозной функциональностью. Соавторам принадлежит общая постановка задачи, модификация компоновщика аспектов и методы проверки достоверности полученных результатов. Статьи 1, 2 и 3, содержащие основные результаты диссертационного исследования, опубликованы в журналах, входящих в перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук, сформированный Минобрнауки России.

Все выносимые на защиту результаты получены лично автором. Программные реализации выполнены автором.

В журналах из перечня рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, доктора наук

1. Григорьева, А. В. Аспектно-ориентированный рефакторинг облачных приложений MS Azure с помощью системы Aspect.NET [Текст] / А. В. Григорьева // Компьютерные инструменты в образовании. — 2012. — № 1. — С. 21–30.
2. Григорьева, А. В. Применение элементов аспектно-ориентированного подхода ASPECT.NET при разработке геолокационной системы управления ресурсами для облачной платформы Microsoft Windows Azure [Текст] / А. В. Григорьева, В. А. Васильев, В. О. Сафонов // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. — 2012. — № 4. — С. 98–102.
3. Григорьева, А. В. Бесшовная интеграция аспектов в облачные приложения на примере библиотеки Enterprise Library Integration Pack for Windows Azure и Aspect.NET [Текст] / Д. А. Григорьев, А. В. Григорьева, В. О. Сафонов // Компьютерные инструменты в образовании. — 2012. — № 4. — С. 3–15.

В изданиях, индексируемых в реферативных базах Scopus и Web Of Science

4. Configure and refactor cloud applications with Enterprise Library Integration Pack for Microsoft Azure using Aspect.NET [Текст] / V. O. Safonov, D. A. Grigoriev, A. N. Safonova, A. V. Grigorieva // Proceedings of the international conference on numerical analysis and applied mathematics 2014 (ICNAAM-2014). — 2015. — Vol. 1648. — P. 310011.

Другие публикации

5. Григорьева, А. В. Реализация механизма доступа к динамическому контексту в точках применения аспектов для системы Aspect.NET [Текст] / Д. А. Григорьев, А. В. Григорьева, В. О. Сафонов // СПИСОК-2014: Материалы всероссийской научной конференции по проблемам информатики. — 2014. — С. 142–147.
6. Григорьева, А. В. Бесшовная интеграция аспектов в облачные приложения на примере библиотеки Enterprise Library Integration Pack for Windows Azure и Aspect.NET [Текст] / А. В. Григорьева, Д. А. Григорьев, В. О. Сафонов // СПИСОК-2013: Материалы межвузовской научной конференции по проблемам информатики — 2013. — С. 107–111.
7. Григорьева, А. В. Аспектно-ориентированный рефакторинг облачных приложений MS Azure с помощью системы Aspect.NET [Текст] / А. В. Григорьева, Д. А. Григорьев, В. О. Сафонов // Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада “Технологии Microsoft в теории и практике программирования”. — 2013. — С. 65–66.
8. Григорьева, А. В. Аспектно-ориентированная разработка облачных приложений на платформе MS Azure с помощью Aspect.NET [Текст] / А. В. Григорьева, Д. А. Григорьев, В. О. Сафонов // Материалы межвузовского конкурса-конференции студентов, аспирантов и молодых ученых Северо-Запада “Технологии Microsoft в теории и практике программирования”. — 2012. — С. 57–58.
9. Григорьева, А. В. Аспектно-ориентированная разработка облачных приложений с помощью системы Aspect.NET [Текст] / А. В. Григорьева, В. О. Сафонов, Д. А. Григорьев // СПИСОК-2012: Материалы всероссийской научной конференции по проблемам информатики. — 2012. — С. 101–104.
10. Григорьева, А. В. Применение системы Aspect.NET для облачных приложений на платформе Microsoft Windows Azure [Текст] / А. В. Григорьева, В. О. Сафонов, Д. А. Григорьев // СПИСОК-2011: Материалы 2-й межвузовской научной конференции по проблемам информатики. — 2011. — С. 107–111.

Подписано в печать 27.06.2016 Формат 60x84¹/₁₆ Цифровая Печ.л. 1.0
Тираж 100 экз. Заказ № 20/06 печать

Типография «Фалкон Принт»
(197101, г. Санкт-Петербург, ул. Большая Пушкарская, д. 41, литер Б,
сайт: falconprint.ru)