

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Григорьев Дмитрий Алексеевич

Реализация подсистемы применения аспектов в системе Aspect.NET

05.13.11 - Математическое и программное обеспечение вычислительных комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата физико-математических наук

Санкт-Петербург

2009

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

1.1 Актуальность работы. Рассматриваемое в работе аспектно-ориентированное программирование (АОП) является одним из наиболее перспективных направлений в развитии технологий программирования. Новизна и актуальность АОП в том, что данный подход предлагает простые и надежные методы разработки и модификации для *сквозной функциональности (cross-cutting concerns)* программных систем: проверки свойств безопасности (security checks), реализации протоколирования (logging) и т.д. Подобная функциональность обычно рассредоточена по коду программы, поэтому ее трудно сопровождать, отлаживать и тестировать. АОП обеспечивает компактное описание сквозной функциональности в виде особого рода модулей – *аспектов*, – применение которых (*внедрение – weaving*) в целевых программах осуществляется в соответствии с правилами, заданными в спецификации аспекта. Еще одной важной особенностью АОП является то, что данная технология может дополнить существующие подходы к разработке ПО, в частности, объектно-ориентированный подход.

В настоящее время АОП активно развивается, проделывая непростой путь от исследовательских разработок до практического применения в реальных проектах. Однако, спустя более 10 лет после своего появления, технология АОП все еще недостаточно популярна среди разработчиков программного обеспечения. Большинство инструментов для работы с АОП ориентированы на платформу Java, что существенно ограничивает их гибкость и применимость.

Проведенный в процессе написания работы анализ предметной области показал, что применение инструментов АОП для перспективной платформы Microsoft.NET не выходит пока за рамки экспериментов. Такое положение дел объясняется новизной этой платформы и ограниченными возможностями механизма рефлексии для гибкого использования сборок .NET. Для преодоления этих трудностей требуются дополнительные исследования. Платформа Microsoft.NET активно развивается, число ее пользователей растет, поэтому средства АОП для этой платформы необходимы и востребованы.

1.2 Цели и задачи диссертационной работы. Данная диссертационная работа выполнена в контексте разработки технологии Aspect.NET. Целью работы является реализация в технологии Aspect.NET следующих возможностей:

- расширенного метаязыка для спецификации аспектов;
- подсистемы применения аспектов (weaver);
- метода интеграции подсистемы применения аспектов с Aspect.NET Framework (GUI).

Для достижения поставленных целей были поставлены и решены следующие задачи:

– исследование истории развития АОП, анализ текущего состояния на момент выполнения исследования, а также путей и возможностей дальнейшего развития АОП;

– обоснование необходимости применения АОП для платформы Microsoft.NET;

– изучение и усовершенствование предложенной научным руководителем архитектуры системы Aspect.NET, в частности, средств спецификации аспектов и подсистемы применения аспектов;

– реализация архитектуры в рамках исследовательского программного продукта Aspect.NET.

1.3 Предметом исследования является проектирование и реализация подсистемы применения аспектов в системе Aspect.NET.

1.4 Методы исследования. В диссертации используются теоретический аппарат объектно-ориентированного и аспектно-ориентированного программирования.

1.5 Обоснованность и достоверность полученных результатов обусловлена:

– корректным применением теории объектно-ориентированного и аспектно-ориентированного программирования;

– результатами сравнительного анализа разработанных методов с другими подходами к АОП;

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

1.6 Научные результаты. В диссертации разработаны методы реализации языка спецификации аспектов в АОП для платформы Microsoft.NET, проанализированы и обоснованы преимущества предложенных методов. Основными результатами диссертации являются:

- оригинальная методика сравнения реализаций языков АОП;
- обоснование расширения метаязыка АОП Aspect.NET.ML;
- метод интеграции подсистемы АОП в среду Microsoft.NET;
- модель программного продукта на основе АОП в рамках платформы Microsoft.NET;
- спроектированная и реализованная подсистема применения аспектов в системе Aspect.NET.

1.7 Положения, выносимые на защиту.

1. Набор методов разработки и использования систем АОП, которые могут быть интегрированы в среду Microsoft.NET.
2. Обоснование эффективности использования Aspect.NET для разработки программных продуктов на платформе Microsoft.NET.
3. Архитектура подсистемы применения аспектов.
4. Описание метаязыка спецификации аспектов.

1.8 Практическая значимость работы

Результаты работы могут быть использованы для решения широкого круга практических задач разработки надежного и безопасного программного обеспечения. Диссертация содержит исторический очерк развития систем АОП, описывает основные понятия АОП, также отдельно взятую подсистему Aspect.NET, включает в себя сравнительный анализ подсистем АОП, их положительные характеристики. С помощью выполнения ряда пилотных проектов выявлена эффективность использования подсистем АОП в рамках технологии Microsoft.NET, показана их актуальность, гибкость и перспективность.

1.9 Апробация работы. Основные результаты и выводы диссертации докладывались на следующих научных конференциях: «.NET Technologies 2006» (г. Пльзень, Чехия), «Технологии Microsoft в теории и практике программирования» (Санкт-Петербургский Политехнический университет, 2006, 2007 и 2008 гг.), международная конференция IEEE, посвященная 110-летию изобретения радио (Санкт-Петербург, 2006). Результаты работы также неоднократно докладывались на семинарах кафедры информатики и лаборатории Java-технологии математико-механического факультета СПбГУ. По материалам диссертации опубликовано 7 печатных работ [1 – 7].

Система Aspect.NET, ключевым компонентом которой является разработанная подсистема применения аспектов, используется для решения исследовательских задач разработки программного обеспечения, а также для обучения современным технологиям программирования в университетах. Позитивный опыт и результаты внедрения подтверждаются многочисленными положительными отзывами пользователей системы. Наши разработки в сфере АОП активно используются коллегами из университета Fortaleza (Бразилия), Миланского университета (Италия), Арабской Академии ИТ (Египет) и ряда других университетов.

2. СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность работы, формулируются цели, задачи и предмет диссертационного исследования, показана научная новизна и практическая ценность полученных результатов, описывается структура работы.

В первой главе проводится исторический обзор развития АОП, определяются основные концепции АОП, рассматриваются прикладные области применения АОП. Анализируются концепции, на базе которых строится технология АОП.

Дается краткий исторический очерк исследований, посвященных теме диссертации. На основе исследований развития АОП дается определение аспекта – основной единицы, на которой базируется аспектно-ориентированный подход.

Аспект является модулем, реализующим сквозную функциональность. Данный модуль вызывается в программе посредством специальных точек внедрения аспекта. В свою очередь, аспектно-ориентированный подход рассматривается как возможность обеспечения локализации сквозной функциональности в аспектах для того, чтобы избежать проблем с затратами лишнего времени на тестирование и сборку конечного кода и улучшить общий дизайн системы, а также сократить производственные расходы на внесение необходимых изменений и дополнений в исходный код программы.

Делается акцент на актуальности развития АОП, его гибкости и возможности интеграции с современными платформами для разработки программного обеспечения, в том числе с платформой Microsoft.NET как одной из наиболее перспективных.

Во второй главе диссертации производится сравнительный анализ существующих подсистем АОП. Приводятся основные характеристики существующих языков АОП, каждый из которых детально анализируется. Приводятся примеры, реализуемые с помощью основных инструментов АОП, которые наглядно иллюстрируют область применения аспектов и демонстрируют неоспоримые положительные стороны АОП, области его применения. Также рассматривается аспектный подход при разработке ПО посредством методов АОП. На примерах поясняются языковые конструкции и особенности языков АОП.

В третьей главе описывается практический подход к методам подсистем АОП. Обосновываются преимущества использования АОП в рамках платформы Microsoft.NET. Рассматриваются принципы аспектного подхода к программированию с помощью языков АОП. Проведена классификация по следующим методам применения аспектов в существующих АОП-системах для платформ Java и .NET:

- статическое применение аспектов на стадии посткомпиляции (AspectJ, PostSharp, Aspect.NET);
- перехват потока управления с помощью прокси-объектов на уровне COM (Spring.NET, AspectDNG);

- трансформация потока управления во время загрузки сборки (Weave.NET).

Также предложена классификация по способу определения аспектов:

- метаязык (AspectJ, Compose*, Aspect.NET);
- конфигурационные XML-файлы (Spring.NET, Weave.NET);
- аннотирование пользовательскими атрибутами (PostSharp, AspectDNG, Aspect.NET).

В соответствии с предложенной классификацией сформулировано общее состояние текущего развития АОП-инструментов для платформы .NET.

Определены основные понятия аспектно-ориентированного подхода: точка внедрения аспекта (JoinPoint), как строго определяемая точка выполнения программы для применения действий аспекта и набор (срез) точек внедрения удовлетворяющих заданному условию (Pointcut). Рассмотрен набор инструкций языка AspectJ. Выявлена разница между языковыми группами аспектных и неаспектных подсистем АОП. Рассмотрены основные способы внедрения аспектов и производительность программных продуктов, разработанных с применением АОП.

Рассмотрены методы интеграции языка АОП в среду Microsoft.NET на примере Aspect.NET. Система Aspect.NET выбрана как наиболее подходящая для реализации целей и задач, определенных в научной работе. Подробно рассмотрены теоретические основы и архитектура Aspect.NET.

В четвертой главе рассмотрены практические основы применения технологии Aspect.NET, сформулированы основные принципы метаязыка спецификации аспектов Aspect.NET.ML. Рассмотрена семантика и приведено описание данного метаязыка. Проанализированы методы практического использования метаязыка АОП, выделены основные качества конечного программного продукта, реализованного с помощью системы Aspect.NET в рамках платформы Microsoft.NET. Приведен и проанализирован пример применения аспектов в существующий программный продукт. На основе его практического применения сформулированы преимущества и эффективность использования АОП при написании программного продукта.

Проанализированы положительные стороны использования системы Aspect.NET на конкретном примере кода аспекта:

```
% aspect BankManagement
{
    % instead % call *BankAccount.withdraw(float) && args(..)
    % action
        public static float WithdrawWrapper(float amount)
        {
            BankAccount acc = (BankAccount) % TargetObject;
            if (isLicenceValid(TargetMemberInfo.Name))
                return acc.withdraw(amount);

            Console.WriteLine
                ("Withdraw operation is not allowed");
            return acc.Balance;
        }
}
```

Данное *действие аспекта* (action) определяет метод, вызов которого будет вставлен вместо каждого вызова целевого метода *withdraw*. Внутри действия происходит проверка прав на выполнение операции *withdraw* и вызывается исходная операция с тем же набором параметров, если авторизация прошла успешно.

Рассматривается процесс внедрения аспекта в исходный код программы, процесс компиляции и сборки конечного продукта.

Также рассмотрена авторская разработка технологии подсистемы применения аспектов (weaver), задача которой заключается в интеграции аспекта в конечный код на стороне клиента, что обеспечит дальнейшую популяризацию программы Aspect.NET. Таким образом, приведен и описан пример внедрения аспекта и его реализация.

Описаны методы усовершенствования технологии Aspect.NET, которые демонстрируют, насколько могут быть расширены возможности моделей АОП.

В пятой главе на основе полученных результатов предложены основные методы применения языков АОП в программировании, перспектива которых достаточно очевидна на настоящий момент. Приведены примеры организаций в

сфере разработки ПО, которые развивают системы АОП и активно внедряют их при создании собственных программных продуктов.

Описана функциональность Aspect.NET, обладающая следующими преимуществами при своем использовании:

- использование привычной среды разработки (система Aspect.NET реализована как надстройка над средой разработки ПО Microsoft Visual Studio);
- отсутствие необходимости изучать новый язык, так как существует возможность описывать аспекты на одном из языков платформы .NET (в текущей версии – на C#);
- фильтрация точек внедрения аспекта и навигация к ним;
- возможность исследовать структуру итоговой программы с внедренными аспектами.

В **заключении** подводятся итоги и делаются выводы по проделанной работе, формируются общие установки на продолжение работы в данной области и обозначаются результаты, полученные в процессе выполнения поставленных задач.

3. ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

Основные выводы, приведенные в диссертации, подтверждают решение поставленных задачи анализа перспективности систем АОП в среде разработки ПО. Это показывает, с одной стороны, что системы АОП являются реализацией нового перспективного подхода к программированию, с другой стороны, что освоение и развитие АОП пока находится в начальной стадии.

В ходе выполнения работы достигнуты следующие результаты.

1. Проведена классификация систем АОП, анализ их развития и использования в рамках платформы Microsoft.NET.
2. Разработана спецификация метаязыка определений аспектов.
3. Разработан и реализован алгоритм применения аспектов на уровне исполняемых сборок .NET.

4. Разработан метод пользовательского управления и контроля за точками применения действий аспектов.

Все полученные результаты являются новыми.

Предложенная классификация систем АОП обобщает их текущие возможности, формулирует их достоинства и недостатки при разработке ПО и позволяет сформулировать требования для аналогичного инструмента для платформы .NET.

По сравнению с определением аспектов с помощью XML-файлов (Spring.NET, Weave.NET), описание на предложенном в работе метаязыке позволяет не отделять спецификацию аспекта от его реализации и объединить реализацию аспекта и его спецификацию в одном модуле. Дополнительное использование пользовательских атрибутов .NET позволяет реализовать АОП-функциональность с помощью стандартных языков платформы .NET и ее инструментов PostSharp, AspectDNG. Однако это требует лишнего «инструментовочного» кода в исходных текстах, который затрудняет восприятие достаточно сложных правил применения. Макроязыки AspectJ, Compose* решают эти проблемы, однако их недостатками является как то, что эти макроязыки реализованы на уровне исходного текста и ограничены лишь одним языком программирования (AspectJ), так и необходимость создавать отдельную среду разработки (Compose*). В системе Aspect.NET описание аспекта на метаязыке конвертируется в пользовательские атрибуты, что дает возможность наглядно описывать сложные правила применения аспекта, избежать зависимости от конкретного языка и работать с существующим инструментарием разработки ПО для платформы .NET. Кроме того, Aspect .NET ML дает действию аспекта доступ к контексту потока в точке выполнения, поддержка которого отсутствует в аналогичных АОП-системах, основанных на пользовательских атрибутах.

Приложения, полученные в результате применения аспектов при посткомпиляции MSIL-сборок (PostSharp, Aspect.NET), обладают более высокой производительностью, чем те, которые получают вставку действий во время выполнения программы (Spring.NET, AspectDNG) или во время начальной загрузки сборки (Weave.NET). С учетом того, что во время выполнения необходимо

постоянно выполнять проверку условий для определения точки выполнения, а также ограниченности доступа к окружающему контексту с помощью технологии COM-перехвата, применимость основанных на ней АОП-систем ограничена. В случае применения аспектов на стадии загрузки сборки проблемы аналогичны, за исключением того, что большая часть накладных расходов переносится на момент запуска приложения. По сравнению с компилятором AspectJ, подсистеме применения аспектов Aspect.NET не требуются исходные тексты проекта. Если мы знаем организацию целевой системы на уровне вызовов API (например, из документации, из функциональной спецификации или из декомпилированного кода), то для применения аспектов необходимы лишь сборка приложения и соответствующего аспекта, сборка целевого приложения, а также, собственно, подсистема применения аспектов. PostSharp также применяет аспекты на уровне MSIL-сборок, однако в отличие от Aspect.NET вставляет в них сервисный код, который замедляет итоговое приложение и вносит в него дополнительные зависимости. Более того, результат применения Aspect.NET в ряде случаев быстрее вставки аналогичной функциональности «вручную». Во-первых, при применении аспектов используются специальные MSIL-инструкции, которые недоступны на уровне исходного текста (напр. ldtoken). Во-вторых, итоговые .NET-сборки создаются с помощью среды построения компиляторов MS Phoenix, в которую встроен ряд оптимизаций.

Наконец, предложенная архитектура подсистемы применения аспектов способствует предотвращению нежелательных эффектов применения аспекта (например, «лишних» вставок в код, которые в данной точке не имеют смысла) и разрешению коллизий (например, таких, как наложение действий разных аспектов в одной точке). Процессы обнаружения точек применения и применение аспектов в целевой сборке разделены. После этапа обнаружения, список найденных точек с информацией об их местонахождении в исходном тексте целевого проекта передается пользователю. Проанализировав местонахождение обнаруженных точек, пользователь может отфильтровать их список и передать в процесс применения аспектов. На данный момент ни в одном АОП-инструменте, кроме

Aspect.NET, невозможно отфильтровать обнаруженные точки применения аспектов.

Результаты, полученные в процессе выполнения проектов с использованием разработанной технологии, позволяют сделать вывод о работоспособности и эффективности программ для платформы Microsoft.NET, разработанных с помощью системы Aspect.NET.

4. СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Григорьев Д.А., Грачев М.К., Масленников А.И., Сафонов В.О. Aspect.NET: инструментальный аспектно-ориентированного программирования для платформы [Microsoft.NET](#). // Тезисы докладов конкурса-конференции «Технологии Microsoft в теории и практике программирования». СПб.: изд-во СПбГПУ, 2006.
2. Григорьев Д.А., Грачев М.К., Масленников А.И., Сафонов В.О. Адаптация методологии АОП для практического применения на платформе [Microsoft.NET](#). // Тезисы докладов конкурса-конференции «Технологии Microsoft в теории и практике программирования». СПб.: изд-во СПбГПУ, 2007.
3. Григорьев Д.А., Грачев М.К., Масленников А.И., Сафонов В.О. Адаптация методологии АОП для применения к задачам протоколирования на платформе [Microsoft.NET](#). // Тезисы докладов конкурса-конференции «Технологии Microsoft в теории и практике программирования». СПб.: изд-во СПбГПУ, 2008.
4. Григорьев Д.А. Реализация и практическое применение аспектно-ориентированной среды программирования для Microsoft .NET // СПб.: Научно-технические ведомости, СПбГПУ. 2009. № 3. С. 225-232.
5. Grigoriev D., Safonov V. O. [Aspect.NET](#): aspect-oriented programming for Microsoft.NET in practice. // NET Developer's Journal. 2005. №7. P. 28-33.
6. Grigoriev D., Safonov V. O. [Aspect.NET](#) - an aspect-oriented programming tool for [Microsoft.NET](#). // Proceedings of the 110th Anniversary of Radio Invention; St. Petersburg; 2006. P. 11-21.
7. Grigoriev D. et al. Aspect.NET – aspect-oriented toolkit for Microsoft.NET based on Phoenix and Whidbey. // Knoop J, Skala V, editors. .NET Technologies 2006. University of West Bohemia. Campus Bory. Full papers proceedings; Pilsen, Czech Republic; 2006. P. 19-34.