

А. В. НИКОЛАЕНКОВ

АСПЕКТНЫЕ ТЕХНОЛОГИИ В СИСТЕМАХ С ПРЕОБЛАДАЮЩЕЙ ПРОГРАММНОЙ КОМПОНЕНТОЙ

Понятие аспекта положено в основу ряда подходов к разработке вычислительных систем, от технологии к технологии его трактовка различается. Проведен анализ использования понятия аспекта в технологиях создания программного обеспечения и в аспектной технологии проектирования встраиваемых систем.

Ключевые слова: аспектная технология, аспектно-ориентированное программирование, субъектно-ориентированное программирование, композиционные фильтры, адаптивное программирование.

Введение. Аспектные технологии [1] позволяют решать задачу проектирования встраиваемых систем (ВсС) на абстрактном уровне. Аспект используется для декомпозиции задачи проектирования и контроля качества ее решения. Схожие идеи положены в основу аспектных подходов к разработке программных систем. В настоящей работе анализируется трактовка аспекта в технологиях создания программного обеспечения (ПО) и аспектного проектирования ВсС.

Несмотря на значительные различия между ВсС и системами с доминирующей программной компонентой, на концептуальном уровне возможно использование схожих подходов к работе с аспектами. Сопоставление и анализ способов использования аспектов в области разработки программных систем будут полезны два развития аспектной технологий.

Сопоставление аспектной технологии проектирования и аспектных технологий программирования выполнялось в рамках работы [1], в настоящей статье приводится более детальный анализ.

Аспектные технологии программирования. Технология объектно-ориентированного проектирования (ООП) программных систем основана на принципах иерархической декомпозиции. В работе [2] наглядно показан ее главный недостаток: при построении иерархии между объектами реального мира предполагается наличие не зависящих от контекста рассмотрения иерархических связей. Для построения иерархии выделяется доминирующий критерий и второстепенные связи вытесняются „за скобки“. Отсутствие достаточного внимания к второстепенным критериям декомпозиции при написании программы приводит к появлению в программном коде повторяющихся конструкций (scattering) и перемешиванию реализации второстепенной функциональности с основной (tangling). Для работы с второстепенными (горизонтальными) связями вводится понятие аспекта. Анализ литературы выявил четыре основных реализации этих идей: аспектно-ориентированное программирование (АОП), субъектно-ориентированное программирование (СОП), композиционные фильтры (КФ), адаптивное программирование (АП).

В АОП наиболее развита реализация аспектных идей [3]. Основное инструментальное средство — AspectJ. Аспект рассматривается как единица модульности высокоуровневого объектно-ориентированного языка программирования и объединяется с элементами основной иерархии с помощью „точек включения“ (point-cuts). В рамках технологии АОП прорабатываются языковые средства для работы с аспектами и эффективные методы реализации модульности на основе аспектной идеи.

Идеи СОП были представлены в статье [4]. В рамках СОП-реализации организуются „субъекты“ и предлагаются методы для работы с ними. Идеи СОП развились в подход многомерного разделения ответственностей [5]. Несмотря на использование этих идей в инструментальном средстве Nurer/J, активных исследований в этой области не отмечено.

Подход КФ [6] позволяет рассматривать взаимодействующие объекты как обменивающиеся сообщениями сущности и предлагает набор инструментов для гибкой работы с сообщениями. На основе предложенных инструментов реализуются базовые механизмы ООП (наследование, агрегация) и АОП.

В АП программный код отделяется от структуры программы с последующим объединением непосредственно перед этапом выполнения программы. Аспектные идеи воплощаются в АП путем установки связей между реализациями аспектов и „точками включения“. Детальное описание АП представлено в статье [7]. Авторами проделана значительная работа по доказательству корректности предлагаемого подхода, рассмотрено внедрение идей АП не только в объектно-ориентированное, но и в функциональное программирование [8]. Одним из наиболее известных достижений АП можно считать формулирование закона Деметры [9], определяющего правильное построение связей между элементами программы.

Аспектная технология проектирования (АТП) [1] — высокоуровневая методика проектирования ВcC. В качестве аспектов АТП выделяет частные проблемы проектирования и предлагает подходы для работы с ними на всех этапах жизненного цикла ВcC. Примерами аспектов могут служить „надежность“, „стоимость“, „энергопотребление“. Задача проектирования рассматривается АТП как задача многокритериальной оптимизации в рамках определенных аспектами ограничений.

Широкая трактовка аспекта необходима для разработки ВcC с контролируемыми свойствами. Основное назначение аспекта АТП — построение абстракций для анализа свойств системы. Введение аспектных ограничений обеспечивает гибкость работы с проектным пространством и способствует проведению более тщательного анализа проектных решений.

Аспектные технологии программирования трактуют аспект как единицу модульности, упрощающую анализ системы и повышающую коэффициент повторного использования. Введение аспектов приводит к увеличению числа компонентов системы, а явное выделение уровня связей — к росту ее сложности и общей связности. Это обуславливает ограниченное применение аспектной технологии и использования ее для реализации ортогональных аспектов, содержащих небольшое количество простых механизмов. В качестве аспектов в программных системах выделяются журналирование, авторизация, аудит и др. Эффективная работа с аспектными технологиями при разработке программных систем требует поддержки аспектов на этапах проектирования и разработки требований.

В представленных подходах назначение аспектов существенно различается. Аспектные идеи, заложенные в основу рассматриваемых технологий, позволяют вводить дополнительные способы выделения компонентов и методы работы с ними. При использовании аспектных идей отмечаются общие проблемы организации взаимодействия между аспектами, анализа аспектных отношений, отсутствия унифицированных методов работы с аспектами на всех этапах жизненного цикла вычислительной системы.

Заключение. Для систем с преобладающей программной компонентой необходимо развивать средства высокоуровневого проектирования. Существующие аспектные методы могут выступать в качестве эффективного средства архитектурного проектирования при условии распространения понятия аспекта и его инструментальной поддержки на все этапы создания вычислительной системы.

СПИСОК ЛИТЕРАТУРЫ

1. *Платунов А. Е.* Теоретические и методологические основы высокоуровневого проектирования встраиваемых вычислительных систем: Автореф. дис. ... д-ра техн. наук. СПб: СПбГУ ИТМО, 2010. 39 с.
2. *Ostermann K.* Modules for hierarchical and crosscutting models. Technischen Universitat Darmstadt, 2003.

3. Kiczales G., Mezini M. Aspect-oriented programming and modular reasoning // Proc. 27th Intern. Conf. on Software engineering — ICSE '05. NY, USA: ACM Press, 2005. P. 49.
4. Harrison W., Ossher H. Subject-oriented programming // ACM SIGPLAN Notices. 1993. Vol. 28, N 10. P. 411—428.
5. Tarr P. et al. N Degrees of Separation: Multi-Dimensional Separation of Concerns // Proc. 21st Intern. Conf. on Software engineering — ICSE '99. NY, USA: ACM Press, 1999. P. 107—119.
6. Bergmans L., Aksits M. Composing crosscutting concerns using composition filters // Communications of the ACM. 2001. Vol. 44, N 10. P. 51—57.
7. Lieberherr K., Orleans D., Ovlinger J. Aspect-oriented programming with adaptive methods // Communications of the ACM. ACM Press. 2001. Vol. 44, N 10. P. 39—41.
8. Chadwick B. Functional Adaptive Programming. Northeastern University, 2010.
9. Lieberherr K., Holland I., Riel A. Object-oriented programming: an objective sense of style // ACM SIGPLAN Notices. 1988. Vol. 23, N 11. P. 323—334.

Сведения об авторе

Алексей Витальевич Николаенков — аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедры вычислительной техники; E-mail: nickolaenkov@gmail.com

Рекомендована кафедрой
вычислительной техники

Поступила в редакцию
08.02.12 г.