

Эффективное функциональное тестирование программного обеспечения с помощью системы "Альфа-Тест"

Бураков М.А., Котов А.С.

По подсчетам национального института стандартов и технологии (NIST) ежегодный ущерб, приносимый экономике США сбоями программного обеспечения, составляет 59,5 миллиарда долларов или 0,6% валового внутреннего продукта. Одной из причин подобного явления является, на наш взгляд, тот факт, что развитие, как теоретических основ, так и практических инструментов проектирования и разработки программного обеспечения значительно опережает развитие методик и средств его тестирования. На данном этапе сформировалась устойчивая парадигма тестирования, в которой основная роль отводится низкоуровневому тестированию исходного кода (структурному тестированию) и нагрузочному тестированию программного обеспечения [2]. Недостатком первого направления является невозможность выявления скрытых логических ошибок, не приводящих к краху системы. Основной целью нагрузочного тестирования является выявление характеристик производительности системы. Автоматизация поиска высокоуровневых логических ошибок и ошибок интеграции является наиболее трудоёмкой и наименее разработанной методикой тестирования. Данная статья представляет собой высокоуровневое описание методов поведенческого или функционального тестирования программного обеспечения [1], а также системы "Альфа-Тест", позволяющей эффективно автоматизировать данный вид тестирования.

В функциональном тестировании поведение программного обеспечения моделируется в виде направленного графа, в котором каждый узел соответствует функционально обособленной единице тестируемой системы (подпрограмма, функция, компонент, хранимая процедура БД и т.д.), для которой можно четко определить входы и выходы (рис. 1). Отношения между узлами соответствуют дугам графа. С точки зрения структуры графа возможны четыре типа узлов:

- входной узел - узел без входящих связей;
- выходной узел - узел без исходящих связей;
- узел ветвления - узел с двумя и более исходящими связями;
- вычислительный узел - узел с двумя и более входящими и одной исходящей связью.

Следует отметить, что узел ветвления может не соответствовать оператору ветвления в языке, на котором написана тестируемая система. Каждому узлу ветвления в системе функционального тестирования (СФТ) "Альфа-Тест" соответствует предикат P ,

являющийся отображением множества входных значений I , на множество выходных значений $P(I)$. Если $P(I) = \{true; false\}$, то предикат является логическим, если же $|P(I)| > 2$, то предикат является алгебраическим. Помимо узлов с предикатами (узлов ветвления), системой предусматриваются вычислительные узлы с несколькими входами и одним выходом. Вычислительные узлы используются для построения графов потоков данных. Необходимость в отдельном моделировании потоков данных обуславливается тем, что общепринятая на данный момент идеология объектно-ориентированного программирования использует потоки данных в качестве методологической основы.

Таким образом, узлы графа представляют собой некоторый эквивалент "черного ящика". Не имея доступа к физической реализации функциональности узлов, тестировщик, прежде всего, обращает внимание не на работу отдельных составляющих системы, а на их взаимодействие и влияние на общую производительность, корректность и отказоустойчивость системы. Фактически функциональное тестирование представляет собой следующую ступень после структурного тестирования компонентов и работает на более высоких уровнях абстракции, чем исходный код. При тестировании потоков данных подразумевается, что отдельные объекты, соответствующие вершинам графа, были должным образом протестированы на более низком уровне, так что можно не сомневаться в их надежности и заменить каждый из них соответствующим узлом. Главная проблема с интеграцией не в том, работают ли интегрированные компоненты (это должно быть проверено в ходе структурного тестирования модулей), а в том правильно ли они соединены и общаются друг с другом. Каждый компонент моделируется узлом, а граф потока данных может комбинироваться с графом управления.

Функциональное тестирование представляет собой иерархический процесс, включающий три уровня: тестирование потока управления, тестирование потока данных и тестирование потока транзакций. СФТ "Альфа-Тест" обеспечивает эффективную поддержку всех трёх уровней, позволяя тестировщикам оценить корректность и отказоустойчивость системы с различных углов зрения. На уровне тестирования потоков управления, СФТ "Альфа-Тест" позволяет проверить покрытие либо отдельных критических путей, либо всех узлов и связей графа и фактически является средством тестирования соответствия функциональности системы заявленной спецификации. Поэтому одним из преимуществ функционального тестирования является тот факт, что, в отличие от структурного тестирования, объектом которого является исходных код тестируемой системы, разработка тестов в рамках поведенческого тестирования ведется параллельно с проектированием тестируемой системы на основе спецификаций, и к

моменту завершения кодирования, тестировщики могут сразу же приступить к тестированию.

Все ошибки, для поиска которых предназначено тестирование потока управления, можно найти и на уровне тестирования потока данных, которое включает в себя тестирование потока управления как составной элемент. Тестирование потока данных - более эффективный метод, чем тестирование потоков управления. Он основан на определении модели потоков данных и использовании этой модели в качестве основы для проектирования тестов. СФТ "Альфа-Тест" позволяет тестировщику выбрать узлы, корректность выходов которых необходимо проверить. После этого система подготавливает тестовые сценарии, в состав которых входят все порожденные подграфы. Каждый порожденный подграф получается движением от выходов тестируемых узлов ко всем входным узлам графа. В случае выявления ошибок в потоке выходных данных тестируемого узла, путем визуального анализа порожденных подграфов, можно проследить все цепочки вычислительных и управляющих узлов вплоть до входных параметров тестируемой системы и выявить узлы, являющиеся причиной некорректного поведения системы. Под путем в данном контексте не обязательно подразумевается путь в коде программы. В поведенческом тестировании рассматриваются пути в моделях, описывающих поведение программного обеспечения.

На третьем уровне, СФТ "Альфа-Тест" работает с графами потока транзакций. Графы потока транзакций используют в тестировании систем оперативной обработки транзакций (OLTP-систем) и программного обеспечения пакетной обработки данных. К классу OLTP-систем относятся: банковские системы, системы электронной коммерции, системы автоматизации продаж авиа- и железнодорожных билетов. Граф потока транзакций обладает свойствами как потока управления, так и потока данных. Модель потока транзакций обычно используется как высокоуровневая модель. Наиболее часто её используют в системном тестировании, обычно это делают на стадии, предшествующей интеграции. При тестировании потока транзакций первоочередное внимание следует обращать не на корректную работу отдельных процессов, а на систему в целом. Особенное внимание следует уделять:

- корректности интерфейсов между компонентами;
- корректности маршрутизации транзакций между компонентами;
- организации и дисциплине очередей;
- узлам слияния, поглощения, расщепления и порождения, синхронизации транзакций, одновременности, созданию и уничтожению транзакций, а также дублированию и потере транзакций.

Считается, что идеальная система мониторинга поведения программного обеспечения проходит все возможные пути в тестируемой системе и предоставляет собранную информацию об успешных и неуспешных запусках системы аналитику или средству выявления закономерностей в данных. Однако нагрузка создаваемая при использовании подобного подхода на тестируемую систему и сроки выполнения тестирования не могут устроить ни одного конечного пользователя или тестировщика. В качестве возможного пути решения данной проблемы можно использовать выборочный обход узлов в графе потока управления.

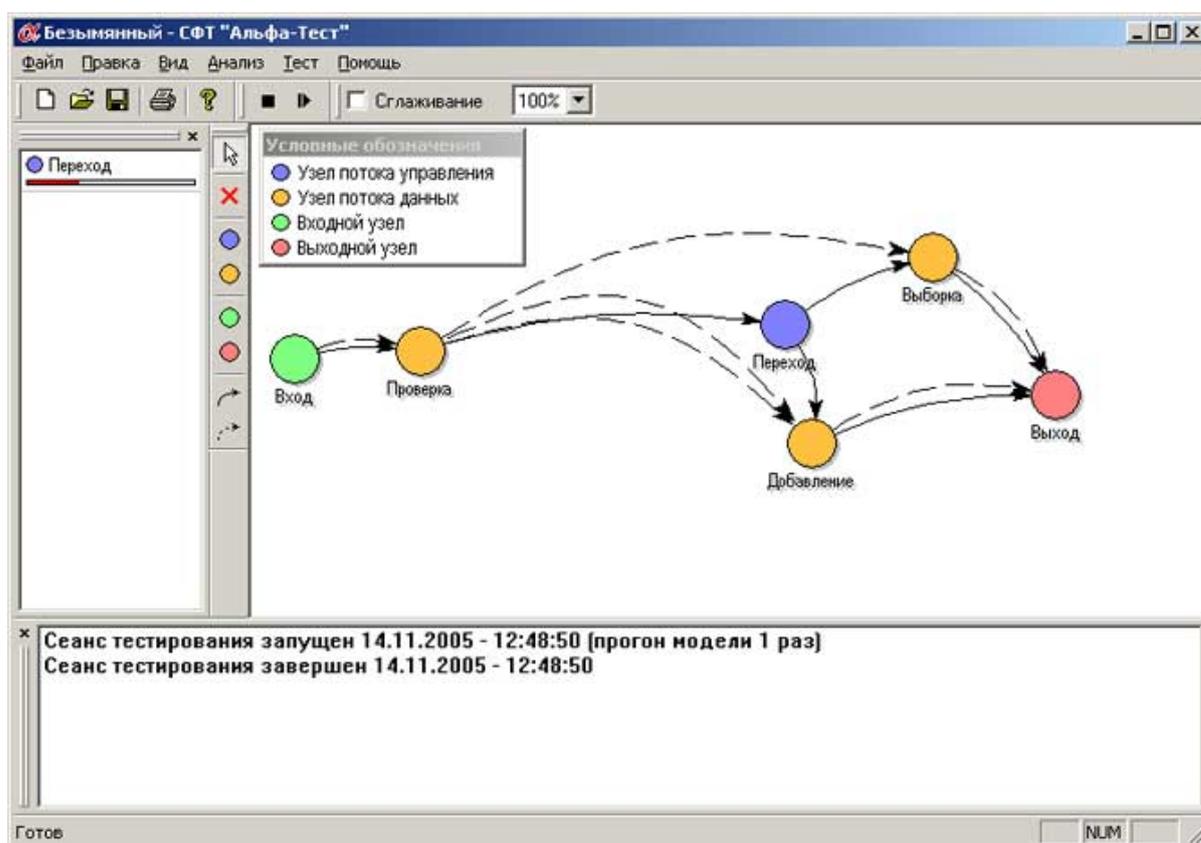


Рис. 1. Центр управления СФТ "Альфа-Тест"

Например, СФТ "Альфа-Тест" позволяет тестировщику выделить узел на графе и протестировать корректность работы данного узла. Для этого необходимо задать пространство выходных значений конечного узла в виде системы алгебраических неравенств, а также указать множества значений входных переменных для каждого из входных узлов подграфа. Таким образом, из исходного графа выделяется подграф, соответствующий некоторому сегменту тестируемой системы. Для генерации тестовых сеансов в СФТ "Альфа-Тест" используются методы тестирования доменов (подпространств входного пространства, в котором задается обработка, выполняемая

тестируемой системой), являющиеся альтернативой общему эвристическому методу тестирования экстремальных значений и предельных значений входных величин. Модель тестирования доменов, а также анализ, необходимый для построения этой модели, эффективны при проверке спецификаций на неопределенности и противоречия. Таким образом, ещё одним преимуществом использования систем функционального тестирования является возможность корректировки спецификаций на ранних этапах жизненного цикла программного обеспечения. Методы тестирования доменов позволяют существенно сократить количество проходов подграфа за счет выбора только тех векторов входного домена, которые могут потенциально вызвать отклонения в поведении системы.

После генерации СФТ приступает к исполнению тестовых сеансов. По мере выполнения проходов подграфа (запусков) формируется статистика событий. Под событием понимается как получение значений, попадающих в допустимую область выходных данных тестируемого узла, так и значений в эту область не попадающих. При этом также регистрируются значения предикатов при прохождении узлов тестируемого подграфа, а также возможные исключительные ситуации, возникающие при выполнении тестовых сеансов.

После исполнения тестовых сеансов информация о событиях обрабатывается аналитическими средствами СФТ "Альфа-Тест", которые построены на базе самых последних теоретических достижений в области статистических методов анализа тестовых данных с целью выявления ошибок [3] [4]. Развитые аналитические средства системы "Альфа-Тест" позволяют выявить условия, при которых проявляются отклонения в поведении тестируемой системы, а также указать на узлы графа функциональности системы, в которых предположительно содержатся ошибки, и ранжировать их по степени вероятности.

СФТ "Альфа-Тест" поддерживает тестирование систем обработки транзакций и работает с СУБД корпоративного уровня (Oracle, MS SQL Server).

Литература:

1. Бейзер Б. "Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем". Санкт-Петербург: Питер, 2004. - 320с.
2. Котов С.Л., Гибин Ю.В., Котов А.С. "Средства обеспечения надежности функционирования информационных систем". Тверь: "Программные продукты и системы", №2, 2002. - с. 17-21.

3. Liblit B., Naik M., Zheng A. X., Aiken A., Jordan M. I. "Scalable statistical bug isolation". In Proceedings of the ACM SIGPLAN' 05 International Conference, Chicago, IL, 2005. - pp. 15-26.
4. Liu C., Yan X., Yu H., Han J., Yu P.-S. "Mining behavior graphs for "back-trace" of noncrashing bugs". In Proceedings of the SIAM Data Mining Conference, Newport Beach, CA, 2005. - pp. 45-52.