И.Б.Бурдонов, А.С.Косачев.
Тестирование компонентов распределенной системы.
Труды Всероссийской научной конференции "Научный сервис в сети ИНТЕРНЕТ", Изд-во МГУ, 2005, стр.63-65.
2 стр.

ТЕСТИРОВАНИЕ КОМПОНЕНТОВ РАСПРЕДЕЛЁННОЙ СИСТЕМЫ

И.Б. Бурдонов, А.С. Косачев

Правильность распределённой системы основана на правильности её компонентов. *Реализация компонента* (РК) правильна, если она *соответствует* функциональным требованиям, задаваемым *спецификацией компонента* (СК). *Функциональность* означает, что требования описывают поведение компонента в терминах его взаимодействия с окружением – другими компонентами. Взаимодействие дискретно. Акт взаимодействия происходит по «обоюдному желанию» компонента и окружения. Такое взаимодействие называется синхронным. Кроме того, компонент может иметь внутреннюю активность (обозначается τ) без взаимодействия с окружением. Если в данный момент времени в компоненте нет внутренней активности, он находится в *стабильном* состоянии и может выполнять только внешнее взаимодействие. Асинхронное взаимодействие – это взаимодействие через опосредующую среду, например, очередь. Если среду рассматривать как дополнительный компонент, асинхронное взаимодействие сводится к синхронному.

Синхронное тестирование – это проверка правильности в эксперименте, когда РК взаимодействует с тестом, подменяющим окружение. Результат эксперимента – трасса: последовательность наблюдений над поведением РК. СК можно понимать как описание трасс, а соответствие – как соотношение между наблюдаемыми трассами РК и трассами СК. Вид трасс и само соответствие определяются: *тестовыми возможностями* – что можно наблюдать и как можно управлять экспериментом; *реализационными гипотезами*, ограничивающими класс тестируемых РК предполагаемыми, но не проверяемыми при тестировании, условиями.

Самый распространённый тип взаимодействия – обмен дискретными сообщениями: РК принимает *стимулы* и выдаёт *реакции*. Наблюдения взаимодействия порождают простейшие трассы – последовательности стимулов и реакций. Другой тип наблюдений – *отказ* от взаимодействия: РК не принимает стимулов, посылаемых тестом, и не выдаёт реакций, принимаемых тестом. Отказ можно наблюдать, если реализационная гипотеза 1) ограничивает время передачи стимула и реакции, а также конечной внутренней активности, и 2) не допускает бесконечной внутренней активности – *дивергенции* («зацикливания»). Для наблюдения отказа в тесте задаётся тайм-аут ожидания взаимодействия, истечение которого трактуется как отказ. Среди отказов выделяют два случая: *стационарность* (обозначается δ) – РК не выдаёт ни одной реакции, *блокировка* стимула x – РК не принимает стимул x.

Для формального определения соответствия нужна удобная математическая модель. Модель СК задаётся самой СК, а для РК принимается *тестовая гипотеза*: существует модель, неотличимая от РК в любом тестовом эксперименте («чёрный ящик»). Для аналитических методов верификации нужно знание модели РК, но для тестирования достаточно её существования. Соответствие – это математическое отношение между моделями РК и СК. Наиболее адекватная и распространённая модель – асинхронный автомат (Input-Output Labelled Transition System [1], Input-Output Automaton [2] и т.п.): набор состояний и переходов между ними, помеченных стимулами, реакциями или τ. Стабильное состояние – состояние, в котором нет τ-перехода. Дивергенция – бесконечная цепочка τ-переходов, в частности, τ-цикл. Отказ в стабильном состоянии – подмножество стимулов и реакций, по которым нет переходов из состояния. Переход выполняется мгновенно (эквивалентно условию 1 в гипотезе об отказах). Трасса заканчивается во множестве состояний, которое моделирует ситуацию после трассы.

Одно их наиболее близких к интуитивному пониманию соответствий – отношение **ioco** (Input-Output COnformance) [1] – основано на представлении о том, что инициатором передачи сообщения является передающая сторона, а принимающая сторона пассивна и всегда готова к приёму. С учётом гипотезы об отказах, стимулы не блокируются, и единственный наблюдаемый отказ – это стационарность (δ). Трассы, кроме стимулов и реакций, могут содержать символ δ; такие трассы называются *трассами с задержками*. Отношение **ioco** требует: в РК может быть данная реакция или δ только тогда, когда она может быть в СК после той же самой общей трассы. При таком определении СК должна быть строго конвергентной (нет дивергенции после любой трассы), а реализационная гипотеза уточняется: после общей трассы в РК 2) нет дивергенции и 3) принимаются все стимулы, имеющиеся в СК. Тестовая возможность, необходимая для проверки **ioco**, – возможность после общей трассы посылать в РК стимул, имеющийся в СК, или принимать все реакции с контролем по тайм-ауту, истечение которого трактуется как δ.

Отношение **ioco** обладает рядом недостатков. *Проблема самоприменимости СК*: СК нельзя рассматривать как одну из правильных РК, что противоречит интуитивному пониманию. Причина в том, что

И.Б.Бурдонов, А.С.Косачев.
Тестирование компонентов распределенной системы.
Труды Всероссийской научной конференции "Научный сервис в сети ИНТЕРНЕТ", Изд-во МГУ, 2005, стр.63-65.
2 стр.

СК не удовлетворяет условию 3 реализационной гипотезы: в одном состоянии после трассы стимул принимается, а в другом – блокируется. Это, однако, не разрешает РК поступать так же – блокировать стимул. Фактически, блокировка в СК по умолчанию заменяется на приём стимула; такое преобразование называется *пополнением*. Пополненная СК уже не содержит блокировок и поэтому самоприменима.

Однако запрет блокировок слишком сильное ограничение для распределённых систем: компонент вовсе не обязан всегда принимать любой стимул, который ему посылают другие компоненты. В данном состоянии он может принимать одни стимулы и блокировать другие. Например, компонент – ограниченная FIFO-очередь, в функциональность которой входит блокировка стимула, когда очередь заполнена. Модель CFSM (Communicating Finite State Machines) рассматривает распределённую систему как совокупность автоматов и очередей, через которые автоматы асинхронно взаимодействуют, что сводится к чисто автоматной модели заменой очереди на моделирующий её автомат. Если очередь ограничена, блокировка может возникнуть при посылке стимула в очередь. Но даже для неограниченных очередей блокировка возможна, когда компоненты не выбирают головной стимул очереди.

Проблема несохранения соответствия при асинхронном тестировании: асинхронный тест ловит ложные ошибки, которые не могут обнаружить синхронные тесты [3]. Это возможно, если стимул не специфицирован после трассы СК: синхронный тест такой стимул не посылает в РК, а асинхронный тест – посылает. Тогда используется *демоническое пополнение*, определяющее приём стимула с любым дальнейшим поведением. Это исключает ложные ошибки, но, поскольку информация о неспецифицированности теряется, вызывает лишние проверки при асинхронном тестировании [4].

Предлагаемое решение этих проблем a) снимает запрет блокировки стимулов, b) допускает толкование неспецифицированного стимула как разрушающего. В автомате появляются переходы по *разрушению*, обозначаемому γ. Семантика такого перехода включает не только произвольное поведение в смысле приёма стимулов и выдачи реакций, но и дивергенцию, а, кроме того, подразумевает возможность разрушения реализации. Пополнение СК может трактовать неспецифицированный стимул не только как принимаемый, но и как блокируемый или разрушающий – ведущий в состояние с γ-переходом. Если при демоническом пополнении такой стимул посылать бессмысленно, то при пополнении разрушения – запрещено. Возможны комбинированные варианты, трактующие неспецифицированный стимул в зависимости от состояния. Например, хорошо отвечает интуиции такое пополнение: в нестационарных состояниях неспецифицированный стимул блокируется, а в стационарных – блокируется или разрушает. Будем считать, что в СК нет умолчаний: она уже пополнена, и неспецифицированный стимул понимается как блокируемый. Трассы могут содержать не только стимулы, реакции и δ, но также блокировки стимулов и γ; такие трассы будем называть *βγδ-трассами*.

В тестовые возможности добавляется тайм-аут на передачу стимула в РК, истечение которого означает блокировку стимула. Тестирование должно быть безопасным – не вызывать разрушение РК. Поэтому проверяются только *безопасные трассы*: не содержащие, не продолжающиеся и не имеющие ответвлений по *опасным символам*. Символ γ всегда опасен. Если есть трасса σ·x·γ, где x стимул, то после σ опасны как x, так и блокировка x (для проверки x или блокировки x тест посылает x). Если есть трасса σ·y·γ, где y реакция, то после σ опасны все реакции и δ (для проверки любой реакции или отсутствия реакций тест принимает все реакции). Заметим, что безопасность символа z после трассы σ не означает наличие трассы σ·z. СК должна быть *безопасной* (не «саморазрушающейся») – пустая трасса безопасна, и *безопасно-конвергентной* – нет дивергенции после любой безопасной трассы. Реализационная гипотеза: если трасса σ безопасна в СК и есть в РК, то 2) после σ в РК нет дивергенции и 3) σ безопасна в РК. Отношение **ioco** обобщается до $\mathbf{ioco}_{\beta\gamma\delta}$: если трасса σ безопасна в СК и есть в РК, то после неё в РК может быть символ, безопасный после σ в СК, только тогда, когда он может быть в СК после σ.

Как для **ioco**, тест генерируется по трассе СК, но под трассой понимается безопасная βγδ-трасса. В тесте эта трасса заканчивается в терминальном состоянии с вердиктом **pass**, и от неё ответвляются переходы в терминальные состояния с вердиктами **pass** или **fail** по тем символам, которые спецификация после предыдущей трассы допускает (кроме следующего символа трассы) или не допускает, соответственно. Эти символы: стимул и/или его блокировка, когда тест посылает стимул, и реакции и/или δ, когда тест принимает все реакции.

ЛИТЕРАТУРА:

1. J. Tretmans "Test Generation with Inputs, Outputs and Repetitive Quiescence" // Software-Concepts and Tools, v. 17, Issue 3, 1996.
2. N.A. Lynch, M.R. Tuttle "Hierarchical correctness proofs for distributed algorithms". Proc. of the 6th ACM SIGACT-SIGOPS Symp. on Principles of Distributed Computing, pp. 137-151, 1987.

И.Б.Бурдонов, А.С.Косачев.
Тестирование компонентов распределенной системы.
Труды Всероссийской научной конференции "Научный сервис в сети ИНТЕРНЕТ", Изд-во МГУ, 2005, стр.63-65.
2 стр.

---

3. C. Jard, T. Jéron, L. Tanguy, C. Viho "Remote testing can be as powerful as local testing", FORTE XII/ PSTV XIX' 99, China, pp. 25-40.
4. M. van der Bijl, A. Rensink, J. Tretmans "Compositional testing with ioco" // Formal Approaches to Software Testing: Third International Workshop, FATES 2003, Canada. LNCS v. 2931, Springer, pp. 86-100.