

# Применение технологии UniTesK для тестирования систем с различной конфигурацией активных потоков управления

*С.Г. Грошев, sgroshhev@ispras.ru*

## 1. Введение

В связи с возрастанием размера и сложности программного обеспечения (ПО), которое требуется для удовлетворения потребностей пользователей и поддержки стабильного развития современного общества, задача автоматизации тестирования становится одной из ключевых в разработке качественного ПО. Одним из перспективных подходов к решению этой задачи является UniTesK – технология автоматизированного функционального тестирования на основе формальных методов, разработанная в Институте системного программирования РАН [1, 2, 4, 5]. Данная технология позволяет автоматизировать разработку и выполнение тестов, которые с высокой степенью надежности проверяют корректность поведения тестируемой системы. В дальнейшем в статье вместо термина «тестируемая система» используются термины «целевая система» или просто «система» – во избежание путаницы с похожим термином «тестовая система», который также будет использоваться.

Большинство реальных систем обладает некоторым внутренним состоянием, которое может влиять на выдаваемые системой реакции и изменяться как при получении воздействий извне, так и в процессе собственной работы системы. Поскольку внутреннее состояние системы в общем случае неизвестно, и, кроме того, оно может содержать много несущественных с точки зрения поставленной задачи тестирования деталей, в технологии UniTesK строится модель целевой системы времени тестирования (в дальнейшем – просто «модель»), отражающая ее проверяемые свойства в удобном для тестирования компактном виде и скрывающая несущественные реализационно-зависимые детали. Повышение уровня абстракции позволяет использовать один и тот же комплект тестов для тестирования различных реализаций одного набора функций. Компонент, называемый медиатором, отображает реализацию и модель друг в друга: реализационное состояние целевой системы – в модельное состояние, модельные тестовые стимулы – в реализационно-зависимые воздействия на целевую систему, а реализационно-зависимые

реакции целевой системы – в модельные реакции. В дальнейшем везде, где не оговорено специально, речь будет идти именно о модельных состояниях, стимулах и реакциях.

Тестовый сценарий UniTesK перебирает состояния целевой системы и в каждом из них перебирает подаваемые ей на вход тестовые стимулы вплоть до достижения заданной степени тестового покрытия. Специальный компонент, получаемый на основе спецификаций и называемый оракулом, автоматически анализирует выходные реакции и выдает вердикт об их корректности. Корректность поведения целевой системы определяется его соответствием спецификации. Также автоматически оценивается степень достигнутого тестового покрытия [2, 5].

В технологии UniTesK последовательность тестовых воздействий строится интерактивно, по мере выполнения теста. Для этого используются специальные компоненты, называемые обходчиками. Задача обходчика – обход графа состояний системы, причем этот граф не задается заранее, а строится прямо в процессе взаимодействия тестовой и целевой систем. Очередное тестовое воздействие определяется на основе предыдущей истории взаимодействия. Такой подход обеспечивает простоту написания тестовых сценариев и гибкость тестирования.

Интерактивная природа тестирования порождает ряд проблем, связанных с взаимной активностью тестовой и целевой систем: целевая система может иметь несколько взаимодействующих между собой потоков управления, взаимодействовать непредсказуемым образом с внешними системами, неподконтрольными тестовой среде, выдавать реакции на определенные воздействия по прошествии значительного времени и выполнять какую-то собственную деятельность. Все это может влиять на состояние целевой системы и, в результате, на наблюдаемые тестовой системой реакции. Кроме того, даже при полной пассивности целевой системы может возникнуть необходимость тестирования ее работы в окружении множества параллельно работающих активных потоков, создающих запросы к ней. Во всех случаях тест должен определять корректность поведения целевой системы и добиваться заданной степени тестового покрытия.

Опыт использования технологии UniTesK для тестирования реальных систем показывает, что она подходит для тестирования практически любой конфигурации активных потоков теста, целевой системы и внешних систем, однако различные конфигурации требуют построения различных моделей целевой системы и использования различных инженерных решений в построении тестовой системы. Данная работа посвящена особенностям тестирования различных конфигураций активных потоков с помощью технологии тестирования UniTesK. Целями работы являются:

- исследование возможности использования технологии тестирования UniTesK для построения тестов с различной конфигурацией потоков управления, принадлежащих тестовой системе и существующих независимо от нее;

- систематизация опыта тестирования различных конфигураций активных потоков;
- разработка подходов для случаев, на которые технология UniTesK в существующем на данный момент виде не рассчитана.

## **2. Классификация тестируемых систем в соответствии с конфигурацией потоков управления**

В технологии UniTesK целевая система рассматривается как «черный ящик». На входы ей подаются стимулы, а на выходах наблюдаются реакции. Для целей тестирования «черного ящика» важны только возможные соотношения между подаваемыми стимулами и получаемыми реакциями, и неважно, что именно происходит внутри целевой системы. Именно на основе этого принципа определяются границы целевой системы, и строится предлагаемая классификация.

Например, пусть имеются три различных системы. Система А обладает одним собственным потоком управления и может в ответ на полученный стимул начать некую собственную активность ограниченной продолжительности; в ходе этой активности она может как принимать новые стимулы, так и выдавать реакции на старые. Система Б аналогична по внешнему поведению, но внутри нее имеется несколько собственных потоков управления. Система В никакой собственной активности не ведет, но также может сразу выдать в ответ на стимул несколько реакций, причем между системой В и тестовой системой находится сеть, которая вносит недетерминированные задержки в передачу как стимулов, так и реакций (случай тестирования через сложную среду). С точки зрения тестирования все эти три системы эквивалентны: про них известно только то, что после подачи стимула в течение некоторого времени возможен как приём реакций от них, так и подача новых стимулов. Поэтому с точки зрения тестирования все они относятся к классу систем с отложенными реакциями [6]. В случае систем А и Б это определяется наличием их собственных активных потоков, а в случае системы В – функционированием сетевого оборудования и системного ПО. В последнем случае у теста может отсутствовать возможность определить, по чьей вине реакция была искажена, потеряна или задержана – системы В или промежуточной сети. Поэтому разумно включить в рамки целевой системы также и сеть и тестировать именно совместное поведение системы В и сети – вместе они составляют некую систему с отложенными реакциями В', правильность работы которой и проверяется. Если же среда допускает возможность сделать такую проверку (например, взаимодействие тестовой и целевой систем происходит по протоколу TCP, который маскирует все эти искажения), то система В может быть отнесена к более простому классу, позволяющему применять более простые модели и инструменты тестирования.

В ходе исследований выделены следующие различающиеся с точки зрения тестирования «черного ящика» конфигурации потоков управления.

1. Целевая система может быть представлена в виде API, предполагающего однопоточное выполнение и не обладающего собственным потоком управления. Существует только один активный поток, принадлежащий тестовой системе. Целевая система не выполняет никаких действий и не меняет свое состояние вне вызовов из тестовой системы.
2. Аналогично п.1, но целевой API предполагает работу в многопоточной среде. Собственным потоком управления целевая система не обладает. Необходимо протестировать корректность работы при параллельном вызове методов целевого интерфейса. Существует несколько активных потоков, все они принадлежат тестовой системе. Целевая система не выполняет никаких действий и не меняет свое состояние вне вызовов из тестовой системы.
3. Существует единственный поток управления, не контролируемый тестом. Активный поток иногда передает управление тесту, но тот обязан быстро вернуть управление, и при этом он не может выполнить свою работу за доступное ему за один раз время. Реакции на все или некоторые стимулы поступают только при следующих вызовах теста. Никакие внешние сообщения в целевую систему не поступают.
4. Аналогично п. 3, но возможны внешние воздействия на целевую систему или ее собственная длительная активность.
5. Существуют два потока управления, один из которых принадлежит целевой системе, другой – тесту; при этом в любой момент времени активен ровно один из них.
6. Тест – активный поток, целевая система – один или несколько полуактивных потоков: при получении стимула она может выдавать на него реакции в течение ограниченного времени, после чего стабилизируется и уже не может менять состояние или выдавать реакции до получения следующего стимула. Во время активности целевая система продолжает принимать новые стимулы, причем реакции на них могут выдаваться в произвольном порядке, и получение новых стимулов может влиять на реакции, выдаваемые системой в ответ на старый стимул.
7. Кроме теста, существуют и другие активные потоки. Целевая система может менять состояние и выдавать сколь угодно много реакций в течение неограниченного времени в ответ на полученный стимул и даже при отсутствии стимулов.

Других конфигураций активных потоков, существенно отличающихся с точки зрения тестирования, обнаружить не удалось. Прежде чем приступить к подробному рассмотрению перечисленных видов конфигураций активных потоков, введем несколько определений.

**Определение 1.** Реакция на некоторый стимул называется *немедленной*, если с точки зрения тестовой системы она поступает непосредственно после подачи стимула. Это означает, что между моментом подачи стимула и моментом получения реакции на него на временной линии модели целевой системы не могут происходить никакие события, существенные с точки зрения модели.

Если несколько реакций на один стимул приходят непосредственно после подачи этого стимула, и с точки зрения модели не существенно, какой из них пришел раньше, то они объединяются в одну сложную немедленную реакцию. Например, возврат из метода нескольких выходных параметров можно рассматривать как одну немедленную реакцию. Простой возврат управления также считается немедленной реакцией.

Вообще, к данному классу можно отнести гораздо более широкий спектр реакций, чем может показаться на первый взгляд. Например, вызываемый метод может реально выполняться где угодно (даже на группе удаленных машин) и работать сколь угодно долго, но если вызывающий его поток блокируется на все это время, то возврат управления и выходных значений в вызывающем потоке считается реакцией, происходящей непосредственно после подачи стимула вызова метода. Кроме того, реакция на стимул может приходиться не в тот поток управления, из которого был подан этот стимул, но если в используемой модели реакция и стимул проецируются на единую временную линию, и на этой линии между ними не может произойти никаких других событий, то такая реакция также считается немедленной. Иногда реакцию можно считать немедленной и в тех случаях, когда целевая система блокируется на время обработки стимула (даже если при этом не блокируется вызывающий ее поток) и задерживает новые стимулы до полной обработки текущего стимула и выдачи всех реакций на него. В этом случае события в модели можно переупорядочить таким образом, чтобы стимулы следовали в порядке поступления их в целевую систему, а все реакции на каждый стимул следовали непосредственно после него.

**Определение 2.** Реакция на некоторый стимул называется *отложенной*, если на временной линии используемой модели целевой системы событие применения данного стимула и событие получения данной реакции на него происходят в существенно различные моменты времени. Моменты времени считаются существенно различными, если в промежутке между ними возможны другие события, существенные с точки зрения используемой модели (в том числе, применение других стимулов и получение других реакций).

Как уже было показано выше, любой блокирующий вызов (локальный или удаленный) без пост-эффектов не является стимулом с отложенной реакцией.

**Определение 3.** Целевая система называется *системой с немедленными реакциями*, если с точки зрения используемой модели она производит только немедленные реакции на все стимулы.

**Определение 4.** Целевая система называется *системой с отложенными реакциями*, если на некоторые стимулы она производит отложенные реакции, причем в любой момент можно определить промежуток времени, за которое при отсутствии новых тестовых стимулов система гарантированно придет в стационарное состояние. Состояние целевой системы называется *стационарным*, если при дальнейшем отсутствии стимулов она гарантированно сколь угодно долго не будет менять свое состояние и выдавать новых реакций. Поскольку внутреннее состояние целевой системы нам в общем случае неизвестно, имеется в виду стационарность модельного состояния системы.

**Следствия из определений:**

- в ответ на получение стимула целевая система может выдать не более одной немедленной реакции, а количество отложенных реакций может при этом быть любым;
- реакция является отложенной или немедленной не сама по себе, а с точки зрения модели, используемой для тестирования. Для одной и той же системы могут быть построены разные модели, и если в одной модели допускаются какие-либо события между подачей стимула и получением на него реакции, а в другой – нет, то в первой модели реакция будет отложенной, а во второй – немедленной;
- соответственно, целевая система в целом может являться или не являться системой с отложенными реакциями в зависимости от используемой модели.

Разработчики спецификаций и тестов могут выбирать ту или иную модель в зависимости от свойств целевой системы и поставленной задачи тестирования. Модели с отложенными реакциями включают в себя всю выразительную мощь моделей без отложенных реакций, но более сложны. Рекомендуется всегда выбирать для тестирования из множества моделей, отражающих все проверяемые свойства системы, модель наименьшей возможной сложности.

### **3. Особенности построения тестовой системы и модели целевой системы в зависимости от конфигурации активных потоков**

#### **3.1. Пассивные последовательные системы с немедленными реакциями. (Единственный поток управления принадлежит тестовой системе.)**

**Примеры:** большинство подсистем стандартной библиотеки libC.

Целевая система считается относящейся к данному классу, если она удовлетворяет следующим требованиям:

- в ответ на каждый подаваемый стимул она выдает пустую или немедленную реакцию;

- вне вызовов из тестовой системы целевая система не выполняет никаких действий и не меняет свое состояние;
- поставленная задача тестирования не включает проверку работы целевой системы в многопоточном окружении, или такая работа невозможна.

Исходно технология UniTesK разрабатывалась для тестирования именно данного класса систем. Рассмотрим вкратце способ тестирования, применяемый в данном случае.

При создании тестового сценария используется абстракция обобщенных состояний, где каждое обобщенное состояние соответствует некоторому классу эквивалентности реализационных состояний целевой системы. Обычно в модели обобщенных состояний используется значительно более высокий уровень абстракции, чем в модели, используемой для проверки корректности работы целевой системы. В любой момент времени целевая система находится в некотором однозначно определенном обобщенном состоянии и может изменить его только в результате применения тестового стимула. После применения некоторого стимула в некотором обобщенном состоянии целевой системы анализируется полученная реакция и вычисляется новое обобщенное состояние.

Задача тестового сценария состоит в обходе ребер ориентированного графа, вершины которого соответствуют обобщенным состояниям целевой системы, а ребра – применяемым к ней обобщенным стимулам, где каждый обобщенный стимул соответствует некоторому классу эквивалентности возможных стимулов, применяемых к целевой системе. Граф строится интерактивно, по мере выполнения теста, на основе правил вычисления текущего обобщенного состояния целевой системы и правил перебора стимулов, применяемых в зависимости от обобщенного состояния.

Модель обобщенного состояния и обобщенных стимулов строится так, чтобы при повторном применении обобщенного стимула в некотором обобщенном состоянии целевая система всегда переводилась в то же самое обобщенное состояние, в которое она ранее переходила при применениях этого стимула в том же состоянии. Также накладывается дополнительное требование сильной связности графа обобщенных состояний. Практика тестирования реальных систем показывает, что для систем, удовлетворяющих минимальным требованиям к детерминизму поведения, построение модели, удовлетворяющей изложенным требованиям, возможно.

Способы тестирования, применяемые в технологии UniTesK для данной конфигурации активных потоков, более подробно описаны в [2, 4]. Случай недетерминированного поведения целевой системы частично рассматривается ниже, но в целом выходит за рамки данной работы [7]

### 3.2. Пассивные параллельные системы с немедленными реакциями. (Множественные потоки управления, принадлежащие тестовой системе.)

**Примеры:** реализация любой подсистемы стандартной библиотеки libC, заявленная как устойчивая к многопоточности («thread-safe»). Библиотека управления многопоточностью libThreads.

В технологии UniTesK задача тестирования систем данного класса решается с помощью специальных сценариев, определяющих наборы параллельных воздействий [4]. Кроме проверок поведения целевой системы, выполняемых после применения каждого тестового стимула, проверяется следующий постулат сериализации: *«Результат любого параллельного применения стимулов к целевой системе аналогичен некоторому последовательному их применению»*. Именно необходимость проверки данного постулата не позволяет отнести такие системы к классу 1. Для проверки постулата модель целевой системы включает в себя сериализацию событий (примененных стимулов и полученных реакций), которая строится следующим образом.

- Все события (применение стимулов и получение связанных с ними реакций), произошедшие в ходе выполнения тестового сценария, протоколируются.
- На множестве событий строится частичный порядок: линейный порядок определен для всех событий, произошедших в пределах одного потока управления; на основании дополнительных знаний о целевой и тестовой системах может быть определен частичный порядок событий, произошедших в разных потоках.
- Строятся все возможные линейные порядки событий, удовлетворяющие построенному частичному порядку.
- Работа целевой системы признается корректной, если найден хотя бы один такой линейный порядок событий, который удовлетворяет спецификации целевой системы.

**Замечание:** В процессе выполнения теста существует несколько потоков управления, принадлежащих тестовой системе, и, соответственно, в используемой в это время модели целевой системы может иметься несколько временных линий. Однако, после построения сериализации все события выстраиваются на единой временной линии, так что к моделируемой системе становится полностью применимым данное выше определение системы с непосредственными реакциями. Именно на этом основании системы данного класса отнесены к классу систем с немедленными реакциями.

### 3.3. Тестирование с отложенными реакциями без собственного активного потока. (Тестовая система не обладает собственным активным потоком. Целевая система изолирована. Отложенные реакции.)

**Пример:** Object Request Broker (ORB) RACE, разработанный для однопроцессорных встроенных систем. И тест, и целевая система – объекты, существующие под управлением этого ORB. Стимулы и реакции – сообщения ORB, причем сообщения передаются между объектами только в промежутке между получением ими управления.

В данном случае для задачи тестирования неважно, сколько реально существует активных потоков (см. ниже обсуждение случая 6). В любом случае тест не обладает собственным потоком управления и может в течение некоторого времени после применения тестового стимула получать реакции на него от целевой системы и выдавать новые стимулы. Именно эти критерии заставляют относить целевую систему к данному классу систем.

Если активный поток не является единственным, то требуется построение модели, аналогичной той, которая используется в случае 6 (см. ниже). Если такой поток – единственный, то данный случай проще случая 6, поскольку не требуется специальное построение сериализации событий. Однако случай 3 не может быть сведен к случаям 1 или 2, потому что возможны отложенные реакции целевой системы.

Задача построения модели для систем данного класса решается аналогично случаю 6. Однако для системы с конфигурацией такого рода потребовалась особая реализация обходчика: обычный тест UniTesK выполняется в собственном потоке управления с собственным стеком, в котором хранятся вызовы методов и их локальные переменные; в данном же случае тест не обладает собственным стеком вызовов и не имеет возможности выполнять какие-либо длительные действия.

Упрощенный алгоритм работы тестовой системы в случае 3 выглядит следующим образом:

1. Обеспечить регулярное получение управления в активном потоке (например, подписаться на получение сигналов от таймера);
2. При очередном получении управления:
  - 2.1 Проверить реакции от целевой системы, полученные за время ее самостоятельной работы, и состояние, в котором она находится; при необходимости сообщить об ошибках;
  - 2.2 Если тест находится в состоянии ожидания реакций от целевой системы:
    - 2.2.1 Если получены не все ожидаемые реакции и время ожидания еще не вышло, вернуть управление;
    - 2.2.2 Если все ожидаемые реакции получены, перейти к шагу 2.4;
    - 2.2.3 Иначе сообщить об ошибке;

- 2.3 Если тест находится в состоянии ожидания некоторого состояния целевой системы:
  - 2.3.1 Если не все ожидаемые реакции получены и время ожидания еще не вышло, вернуть управление;
  - 2.3.2 Если нужное состояние достигнуто, перейти к шагу 2.4;
  - 2.3.3 Иначе сообщить об ошибке;
- 2.4 Вычислить очередной стимул, который следует применить к целевой системе в текущем состоянии;
- 2.5 Если такой стимул найден, применить его, проанализировать немедленные реакции, при необходимости перейти в состояние ожидания реакций и вернуть управление;
- 2.6 Если существует состояние целевой системы N, которое уже существовало в процессе выполнения теста, но в котором были применены не все возможные стимулы, то попытаться перейти в него; найти в уже обойденной части графа состояний путь из текущего состояния в искомое, вычислить и подать нужные стимулы, перейти в режим ожидания состояния N и возвратить управление;
- 2.7 Если такое состояние не найдено, значит во всех обнаруженных состояниях применены все возможные стимулы; тест завершает работу.

### 3.4. Полуконтролируемое тестирование с отложенными реакциями без собственного активного потока. (Поток управления не принадлежит тестовой системе. Существуют внешние воздействия на целевую систему. Отложенные реакции.)

**Пример:** аналогично случаю 3, но целевая система регулярно выполняет по расписанию какую-то работу, влияющую на ее состояние. В качестве такого дополнительного источника воздействий на целевую систему при этом выступает таймер, неподконтрольный тестовой среде. Если в данном примере тестовая среда имеет возможность изолировать таймер и создавать стимулы от него целевой системе тогда и только тогда, когда это необходимо для целей тестирования, то такую целевую систему следует отнести к классу 3.

Из-за наличия неконтролируемых источников воздействий на целевую систему данный случай не может быть сведен к случаям 3, а тем более случаям 1 и 2, поскольку в рассмотренных выше случаях предполагается полная подконтрольность всех стимулов, получаемых целевой системой.

При этом для целей тестирования неважно, сколько активных потоков реально существует и откуда поступают воздействия на целевую систему: от внешних активных потоков или от других объектов, существующих на аналогичных условиях под управлением единого потока управления. В любом случае, когда возможна неконтролируемая тестом активность целевой системы, сложность

модели целевой системы времени тестирования будет аналогична сложности модели, используемой в случае 7, который обсуждается ниже.

С точки зрения особенностей реализации тестовой системы, она аналогична случаю 3.

### 3.5. Два потока управления, работающих строго поочередно.

**Пример:** тестовая конфигурация такого рода была построена для тестирования Verilog-моделей [3].

В данном случае, как и в случае 3, целевая система обладает собственным активным потоком управления, на который можно влиять извне; отличается только реализация теста. При этом и целевая, и тестовая система ведут себя каждая как единственный активный поток, что позволяет запускать целевую систему в обычном для нее режиме, пользуясь при этом классической для UniTesK архитектурой тестового сценария, рассчитанной на единственный поток с немедленными реакциями. Выполнение в каждый момент времени в точности одного из этих потоков достигается средствами межпоточной синхронизации. Такая реализация удобна тем, что оба потока управления (и целевой системы, и теста) обладают собственными стеками.

Модель целевой системы в данном случае аналогична используемой в случае 1, различие только в реализации тестовой среды. Схема взаимодействия компонентов при этом выглядит следующим образом:

1. Целевая система инициализируется и вызывает встроенный в нее медиаторный код, который запускает в отдельном потоке тестовый сценарий UniTesK;
2. Тестовый сценарий инициализируется и переходит в режим ожидания сигнала;
3. Целевая система вызывает медиаторный код, который посылает потоку тестового сценария сигнал, описывающий текущее состояние целевой системы, и переходит в режим ожидания сигнала;
4. Тестовый сценарий получает сигнал от медиатора и анализирует его корректность в соответствии со спецификацией;
5. Тестовый сценарий выбирает очередное тестовое воздействие по правилам, описанным для случая 1; если такое воздействие не найдено, то он завершает работу: иначе сценарий передает медиатору сигнал, описывающий воздействие, и переходит в режим ожидания;
6. Получив сигнал, медиатор выполняет описанное в нем тестовое воздействие и получает ответную реакцию целевой системы;
7. Медиатор посылает тестовому сценарию сигнал, описывающий реакцию целевой системы и ее состояние, и переходит в режим ожидания;
8. Переходим к шагу 4.

Более подробно об архитектуре тестовой системы, применяемой в рассматриваемой конфигурации, можно прочитать в [3].

### 3.6. Система с отложенными реакциями. (Тест – активный поток, целевая система – один или несколько полуактивных потоков.)

**Пример:** FTP-сервер при одновременной работе нескольких клиентов. После получения запроса на сканирование каталога в течение какого-то времени выполняется сканирование, и клиент получает сетевые пакеты с результатами запроса. При этом параллельно другие клиенты могут модифицировать содержимое того же каталога. Поскольку стандарт не гарантирует однозначность результата сканирования при одновременных модификациях сканируемого каталога, результаты запроса могут различаться в зависимости от действий других клиентов, выполняемых в процессе обработки запроса.

В данном случае для задачи тестирования неважно, сколько реально существует активных потоков. В любом случае, в течение некоторого времени после применения тестового стимула возможны как получение от целевой системы реакций на этот стимул, так и подача новых стимулов. Именно эти критерии заставляют относить целевую систему к данному классу систем.

Поскольку в случаях 1, 2 и 5 предполагаются только немедленные реакции, к ним не могут быть сведены системы данного класса. Для систем с отложенными реакциями в технологии UniTesK разработана специальная архитектура теста [6].

В тестовом сценарии имеется один основной активный поток. Этот поток регулярно опрашивает входные каналы для анализа реакций, полученных от целевой системы. При необходимости создаются вспомогательные потоки управления, в которых выполняются так называемые «кэтчеры» (*catcher*). Задача кэтчера – регистрация реакций, полученных от целевой системы, и передача их во входные каналы основного потока управления тестового сценария. Кэтчеры могут также встраиваться в существующие потоки управления целевой системы.

Модель целевой системы времени тестирования включает в себя информацию о тех ранее примененных стимулах, в ответ на которые еще возможно получение реакций (стационарным считается такое модельное состояние целевой системы, в котором это множество пусто). При получении реакции проверяется, допускает ли спецификация получение реакции такого рода в данный момент. Получение неожиданной реакции (в том числе получение любой реакции в стационарном модельном состоянии системы), а также неполучение ожидаемой реакции в течение допустимого времени рассматриваются как ошибки.

Тестовый сценарий перебирает серии тестовых стимулов и применяет их к целевой системе. После подачи такой серии сценарий ждет перехода целевой

системы в стационарное состояние, анализируя при этом все полученные от нее реакции.

Задача тестирования определяется как обход ребер ориентированного графа, вершины которого соответствуют обобщенным состояниям целевой системы, а ребра – применяемым к ней обобщенным стимулам, где каждый обобщенный стимул соответствует некоторому классу эквивалентности серий применяемых к целевой системе стимулов. Граф состояний строится интерактивно, по мере работы теста.

Все события протоколируются, а после завершения теста строится их сериализация. Правила построения частичного порядка событий описываются спецификацией целевой системы и, в общем случае, могут быть более сложными, чем в случае 2. Работа целевой системы признается корректной, если найдена хотя бы одна сериализация событий, удовлетворяющая спецификации.

### **3.7. Полуконтролируемое тестирование. (Тестирование в окружении неконтролируемых тестом активных потоков, влияющих на наблюдаемое поведение целевой системы, которая может обладать отложенными реакциями.)**

**Пример:** таймер. Стимул может быть единственным (подписка на получение сигналов от таймера) или вообще отсутствовать, реакции наблюдаются сколь угодно долго.

К данному классу относятся системы, работающие в активном окружении, если окружение способно влиять на наблюдаемое поведение целевой системы. К нему также относятся системы, обладающие собственными активными потоками управления, если они могут самопроизвольно менять свое наблюдаемое поведение вне зависимости от тестовых стимулов.

С точки зрения тестирования, для таких систем несущественно количество активных потоков, реально существующих вне теста. Все источники внешних воздействий на целевую систему, а также ее собственная активность, вызывающая неконтролируемые изменения состояния, могут быть промоделированы неким единым неконтролируемым источником внешних воздействий. Все системы такого рода могут быть представлены, например, в виде недетерминированного конечного автомата, допускающего переходы по пустому входному символу, возможно, с выдачей непустого выходного символа. Поэтому все такие системы отнесены к единому классу. Для целей тестирования приходится так или иначе моделировать все, что может повлиять на наблюдаемые тестовой системой реакции, и тестировать получившуюся составную систему.

В принципе, практически любая система обладает той или иной степенью недетерминизма, но именно способность целевой системы данного типа менять свое состояние и поведение в широких пределах при отсутствии

тестовых стимулов существенно усложняет тестирование и не позволяет отнести ее к рассмотренным выше классам.

В отделе операционных систем ИСП РАН ведутся исследования по вопросам тестирования недетерминированных систем, но пока что не найдена единая методика тестирования, пригодная для любых систем такого рода. Разработаны методы тестирования систем, допускающих в определенных пределах недетерминизм выдаваемых реакций и изменений внутреннего состояния [7], однако, в общем случае, невозможно достоверно оценивать корректность поведения системы, допускающей сколь угодно много реакций и изменений наблюдаемого снаружи состояния при отсутствии тестовых воздействий.

Понятно, что проверять корректность поведения целевой системы можно только при наличии некоторых ограничений на возможные последствия неконтролируемых воздействий. Именно из доступных ограничений такого рода и нужно исходить при моделировании. Мы можем предложить только некоторые общие рекомендации по уменьшению степени недетерминизма модели, позволяющие свести систему такого рода к одному из более простых случаев.

- По возможности изолировать целевую систему, помещая ее в «песочницу» («sandbox»), в которой все связи с внешним миром заменены связями с соответствующими заглушками – компонентами тестовой системы. Это классическая техника, однако в ряде случаев она не может быть применена, например, если целевая система – большой монолитный компонент, обладающий собственными активными потоками управления, и отсутствует доступ к исходному коду.
- Моделировать только ту часть целевой системы, поведение которой не зависит от неконтролируемых активных потоков. Корректность реакций целевой системы на воздействия, неподконтрольные тесту (и, возможно, непосредственно им не наблюдаемые), при этом не проверяется.
- Конструировать модель целевой системы так, чтобы она отражала только те свойства целевой системы, которыми она должна обладать при любых сценариях ее работы, независимо от возможных последствий взаимодействия с неконтролируемыми потоками управления. При этом моделируемые аспекты являются инвариантами модели относительно неподконтрольных воздействий. Технология UniTesK изначально приспособлена для такого способа решения проблемы, потому что основана на вычислении предикатов над поведением целевой системы: спецификация описывает не точное поведение системы, а только условия, которым это поведение должно удовлетворять. Такой подход позволяет описывать и тестировать системы с достаточно высоким уровнем недетерминизма [2, 5, 7].

С помощью приведенных методов в большинстве случаев задача тестирования может быть сведена к задаче тестирования системы более простого класса: 1, 2 или 6. Однако общей методики решения этой задачи не существует, как не существует и общей для любых систем способа автоматического построения их моделей для целей тестирования. Задача построения моделей целевой системы и преобразования моделей в более «удобный» для тестирования вид требует творческого подхода, и не существует единого метода, решающего эти задачи для любых целевых систем

#### **4. Заключение.**

В данной статье проведена классификация тестируемых систем в соответствии с видом конфигурации активных потоков и рассмотрены особенности тестирования систем из каждого класса с помощью технологии UniTesK. Рассмотрены способы применения технологии UniTesK для тестирования без собственных активных потоков теста, которые ранее не рассматривались. Приведены некоторые способы уменьшения при моделировании недетерминизма поведения, которые позволяют привести тестируемую систему к виду, допускающему тестирование с помощью UniTesK. К сожалению, не существует единого способа моделирования, пригодного для тестирования систем со сколь угодно высокой степенью недетерминизма поведения, однако постоянно ведется работа по расширению класса систем, допускающих тестирование с помощью технологии UniTesK.

#### **Литература**

1. <http://www.unitesk.com/ru/> — сайт, посвященный технологии тестирования UniTesK и поддерживающим ее инструментам.
2. В. В. Кулямин, А. К. Петренко, А. С. Косачев, И. Б. Бурдонов. Подход UniTesK к разработке тестов. Программирование, 29(6), стр. 25-43, 2003.
3. В. П. Иванников, А. С. Камкин, В. В. Кулямин, А. К. Петренко. Применение технологии UniTesK для функционального тестирования моделей аппаратного обеспечения. Препринт 8. Институт системного программирования РАН. Москва, 2005.
4. I. Bourdonov, A. Kossatchev, A. Petrenko, and D. Galter. KVEST: Automated Generation of Test Suites from Formal Specifications. FM'99: Formal Methods. LNCS 1708, pp. 608-621, Springer-Verlag, 1999.
5. V. Kuliainin, A. Petrenko, I. Bourdonov, and A. Kossatchev. UniTesK Test Suite Architecture. Proc. of FME 2002, LNCS 2391, pp. 77-88, Springer-Verlag, 2002.
6. V. Kuliainin, A. Petrenko, N. Pakoulin, I. Bourdonov, and A. Kossatchev. Integration of Functional and Timed Testing of Real-time and Concurrent Systems. Proc. of PSI 2003, LNCS 2890, pp. 450-461, Springer-Verlag, 2003.
7. И. Б. Бурдонов, А. С. Косачев, В. В. Кулямин. Неизбыточные алгоритмы обхода графов: недетерминированный случай. Программирование, 30(1), стр. 2-17, 2004.