

УДК 004.415.53

Г.Л.Гриппа (асп., каф. ИУС), И.Г.Черноруцкий, д.т.н., проф.

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ МЕТОДОМ РАЗБИЕНИЯ ТЕСТОВЫХ СЦЕНАРИЕВ НА ПОСЛЕДОВАТЕЛЬНОСТЬ ЛОГИЧЕСКИХ МОДУЛЕЙ

Новый подход к разработке автоматизированных тестов сродни идее unit-тестирования, но, в отличие от unit-тестов, оперирует не физическими, а в определенном смысле и логическими модулями тестируемой системы.

Для реализации данного подхода необходимо весь тестовый сценарий разбить на последовательность состояний тестируемой системы и логически завершенных действий – случаев использования в терминологии языка объектно-ориентированного моделирования UML, осуществляющих перевод системы из одного состояния в другое.

Отдельное логически завершенное действие рассматривается как своего рода логический модуль тестируемой системы со своей функциональностью. В таком случае необходимо написать набор тестов на каждое логически завершенное действие из последовательности, составляющей тестовый сценарий, проверяющих правильность состояния системы, в которое она переходит в результате совершенного действия. Так как тестируемая система может состоять из множества связанных подсистем, то и проверка состояния всей системы будет заключаться в проверке всех ее подсистем. Например, подсистемы работы с файловой системой, подсистемы интерпретации кода, и т.д.

Таким образом, формируется набор тестов логически завершенных действий (производимых через API тестируемого продукта или через GUI операционной системы), совершаемых над тестируемой системой. При этом в декларации самих действий должны быть инкапсулированы проверки состояний, в которые тестируемая система переходит в результате этих действий, осуществляемых на различных уровнях тестируемой системы, а также проверки, предшествующие действию и проверяющие возможность его выполнения (т.е. идентифицирующие состояние системы, в котором она находится перед совершением очередного действия). В дальнейшем из этого набора строятся различные последовательности, являющиеся тестовыми сценариями, т.е. тестовым скриптом.

Диаграмма действий, составленная в терминах языка объектно-ориентированного моделирования UML версии 2.0 и описывающая исполнение одного шага тестового скрипта-тестирования одного логически завершенного действия, изображена на рис. 1.

Как видно из диаграммы, при выполнении одного тестового шага прежде всего выполняется проверка как внешних, так и внутренних состояний тестируемой системы на уровне различных подсистем. После чего, если состояние системы не удовлетворяет указанным требованиям, происходит прерывание выполнения тестового сценария и выдается сообщение об ошибке. В противном случае выполняется необходимое воздействие на тестируемую систему.

После произведенного действия система управления воспроизведением тестового скрипта должна дожидаться момента окончания реагирования тестируемой системы на совершенное действие, что должен обеспечить модуль синхронизации действий управляющей системы. Информация об окончании отклика тестируемой системы на воспроизведенное действие должна получаться модулем синхронизации действий непосредственно от самой тестируемой системы посредством ее API.

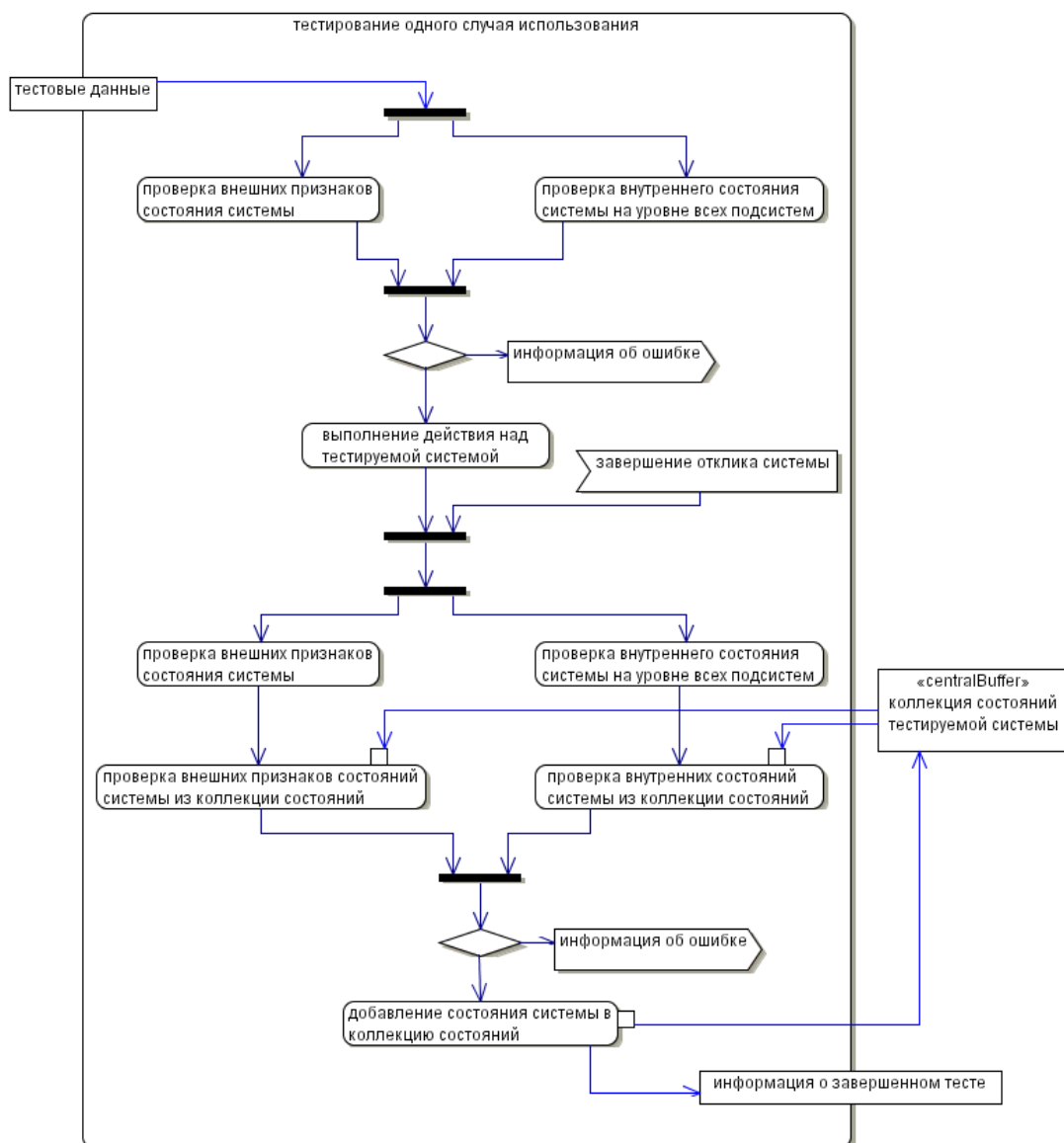


Рис. 1.

После завершения синхронизации необходимо проверить корректность результатов произведенного действия. Здесь на основании входных данных проводится проверка как внешних признаков состояния тестируемой системы, так и внутреннего состояния системы на уровне различных подсистем.

Далее необходимо проверить, что последнее совершенное действие не оказало негативного влияния на не связанную с ним часть тестируемой системы. Так, к примеру, в случае тестирования многооконного текстового редактора необходимо проверить, что при открытии второго файла не было внесено никаких изменений в отношении первого открытого файла, что он не был изменен, удален или закрыт. Для этого вводится некая буферная конструкция, именуемая динамической коллекцией состояний. Ее задача заключается в хранении информации о результатах воздействия предшествующих действий на состояние тестируемой системы. По окончании проведения проверок результата воздействия последнего совершенного над тестируемой системой действия проводится проверка результатов воздействия предшествующих действий, совершенных над тестируемой системой ранее, информация о которых добавляется в динамическую коллекцию состояний после каждого совершенного действия и хранится там до окончания выполнения скрипта. Кроме того, должна быть реализована возможность запоминания внутренних состояний описанной динамической коллекции и последующее обращение к

ним. Это необходимо для тестирования воспроизведения действий, обратных предшествующим (таким как undo/redo).

В случае, если в ходе осуществляемых проверок была обнаружена ошибка, как и в случае проверок, предшествующих выполнению действия, выдается сообщение об ошибке, а выполнение тестового скрипта прекращается.

Однако, стоит заметить, что в данной ситуации возможно просто сбросить информацию об ошибке в какой-нибудь буфер информации и продолжить дальнейшее выполнение тестового скрипта. Это возможно потому, что, в отличие от случая с проверками перед выполнением действия, проверки выполняемые после совершенного действия никоим образом не говорят о возможности или невозможности выполнения следующего действия. Однако, подобная схема работы нежелательна по причине того, что при обнаружении первой же ошибки система входит в непрогнозируемое состояние, и при обнаружении возможных последующих ошибок нельзя с уверенностью сказать являются ли они самостоятельными ошибками или же всего лишь последствиями первой обнаруженной ошибки.

В дальнейшем из таких тестовых шагов строится тестовый сценарий, описанный в файле тестового скрипта.