

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПРИЛОЖЕНИЙ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ

В настоящее время разработка программного обеспечения является достаточно длительным и сложным процессом. Одним из этапов жизненного цикла программного обеспечения является процесс тестирования. Этот процесс требует, как аппаратных ресурсов, так и существенных затрат времени специалистов по тестированию. Для уменьшения количества затрачиваемого времени применяется автоматизация процессов тестирования. Можно выделить отдельный класс программного обеспечения – графические приложения. Особенностью тестирования такого программного обеспечения является непосредственное взаимодействие с элементами интерфейса пользователя. На данный момент не создано универсального средства тестирования, которое удовлетворяло бы всем потребностям разработчиков программного обеспечения, поэтому задача создания системы тестирования остаётся актуальной и на сегодняшний день.

Цель данной работы – разработка системы тестирования, для автоматического тестирования приложений с графическим интерфейсом.

Система тестирования состоит из двух частей: агента, который встраивается в тестируемое приложение и тестирующего приложения. Связь между ними происходит по протоколу TCP. Структура системы тестирования представлена на рис. 1.

Перед началом тестирования необходимо подготовить наборы тестов (эксперименты).

Процедура тестирования следующая: эксперимент выбирается пользователем и запускается на выполнение. Для этого эксперимент тестирующее приложение выбирает тесты из базы данных и передаёт их агенту, который разбирает тесты, выполняет их и возвращает результат. Результаты тестов записываются в ту же базу данных. По результатам выполнения тестов определяется результат эксперимента.

Разработанная система была реализована в ОС Linux Mandrake 9.2 и IDE Kylix 3 при использовании СУБД Firebird.

В результате получена система, позволяющая производить тестирование графических приложений, которые имеют открытый исходный код и используют библиотеку CLX.

Агент реализуется в той же среде разработки, что и приложение, в которое он встраивается. В качестве тестирующего приложения, может выступать приложение, которое способно передавать тесты, сформированные определенным образом, агенту.

Система позволяет выполнять взаимодействие с графическими компонентами, находящимися на основной форме приложения, а так же на дочерних формах и фреймах. Поиск компонентов происходит обходом иерархии компонентов, начиная от главной формы. Агент позволяет имитировать события QT, с помощью соответствующих функции генерации событий, благодаря этому может производиться два типа воздействий: непосредственный вызов методов компонентов или через события QT. В настоящее время система поддерживает основные компоненты CLX. Агент может осуществлять над компонентами воздействия следующих типов:

- активные (нажатия, изменения, выбор и так далее);
- пассивные (чтение значений различного типа).

Имеется возможность расширения набора компонентов и методов.



Рис. 1. Структура системы тестирования

Для улучшения качества тестирования была реализована логика выбора следующего теста, в зависимости от удачного или неудачного выполнения предыдущего теста.

Одним из преимуществ разработанной системы является возможность проводить стресс-тестирование, т.е. воздействие на приложение со скоростью существенно большей, чем может пользователь.

Можно выделить следующие направления развития системы: запуск случайного теста, запуск/перезапуск приложения, в зависимости от результата тестирования, создание отчётов со статистикой тестирования.

В настоящее время, система успешно используется для тестирования реального приложения, которое содержит более ста тысяч строк кода и имеет развитый пользовательский интерфейс, состоящий из большого числа активных и пассивных компонентов различных типов. Система может производить тестирование приложений, заменяя специалиста по тестированию, а значит в любое удобное время, например в выходные дни, что позволяет непрерывно проверять работоспособность и выявлять ошибки.

На основании опыта эксплуатации можно сделать вывод об эффективности применения разработанной системы.