

Рассмотрим полученную матрицу с точки зрения тестового набора регрессионного тестирования. Положим, строки этой матрицы, соответствующие элементам множества A – это множество всех функциональных модулей программы, которые должна подвергнуться проверке, а столбцы – соответствующие элементам множества – тесты. Тогда полученная матрица – и есть минимальный тестовый набор для регрессионного тестирования.

Рассмотрим на небольшом примере работу данного алгоритма. Пусть дана матрица Z .

	1	2	3	4	5	6	7	8	
1	1	0	0	1	1	0	1	1	a
2	0	0	1	1	0	1	0	1	b
3	0	1	0	1	0	0	0	0	c
4	0	0	0	1	1	1	0	1	d
5	0	0	0	1	1	0	1	1	e
6	1	0	1	0	1	0	1	1	f

Рис.1 – матрица Z

Составим для матрицы Z матрицу различий, строки которой соответствуют парам строк диагностической матрицы и показывают, какими компонентами отличаются строки в этих парах. При этом будем использовать покомпонентную операцию сложения по модулю два, выполняя её над всеми парами строк диагностической матрицы.

	1	2	3	4	5	6	7	8	
1	1	0	1	0	1	1	1	0	ab
2	1	1	0	0	1	0	1	1	ac
3	1	0	0	0	0	1	1	0	ad
4	1	0	0	0	0	0	0	0	ae
5	0	0	1	1	0	0	0	0	af
6	0	1	1	0	0	1	0	1	bc
7	0	0	1	0	1	0	0	0	bd
8	0	0	1	0	1	1	1	0	be
9	1	0	0	1	1	1	1	0	bf
10	0	1	0	0	1	1	0	1	cd
11	0	1	0	0	1	0	1	1	ce
12	1	1	1	1	1	0	1	1	cf
13	0	0	0	0	0	1	1	0	de
14	1	0	1	1	0	1	1	0	df
15	1	0	1	1	0	0	0	0	ef

Рис. 2 – Матрица различий для матрицы Z

К данной матрице можно применить правила редукции, т.е. удалить из неё поглощающие строки и поглощаемые столбцы. В результате получаем, что одним из кратчайших столбцовых покрытий данной матрицы является совокупность столбцов 1, 3, 5 и 6. Соответствующая совокупность признаков является минимальной для заданной диагностической матрицы.

В рамках работы был предложен метод минимизации тестового набора на основе задачи о поиске минимального безусловного диагностического теста. В предложенном методе использовался подход, который обеспечивает безопасность и целесообразность уменьшения тестового набора. Анализ результатов проведённого на практике регрессионного тестирования подтверждает работоспособность и эффективность предложенного метода и средства поддержки регрессионного тестирования программного обеспечения: как было показано, уменьшение первоначального тестового набора достигает 25%.

Список использованных источников:

1. Котляров, В.П. Основы тестирования программного обеспечения: Учебное пособие. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. – 285 с.
2. Закревский А. Д. Основы логического проектирования. В 3 кн. Кн. 1. Комбинаторные алгоритмы дискретной математики / А. Д. Закревский, Ю. В. Поттосин, Л.Д. Черемисинова – Мн.: ОИПИ НАН Беларуси, 2004. – 226 с.

МОДЕЛИРОВАНИЕ ОПЦИОНОВ И РАЗРАБОТКА ТОРГОВОЙ СИСТЕМЫ ДЛЯ БИРЖЕВОГО ТРЕЙДЕРА

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Солоненко И.В.

Поттосина С.А., канд. физ.-мат. наук, доц.

Предлагается разработанная модель торговой системы биржевого трейдера, которая используя модель Блэка-Шоулза для расчета опционов, отображает графические данные и диаграммы прибыли и потерь в режиме реального времени.

Опционная торговля - бурно развивающийся вид торговли ценными бумагами. Разнообразие задач, решаемых на опционном рынке, гибкость опционных комбинаций, их сложность требует для опционной торговли использования мощного математического аппарата, а перспективы доходности определяют привлекательность этого вида трейдинга.

Математическое выражение для расчета цены европейского опциона «call» известное как формула Блэка-Шоулза, является основным и наиболее широко распространенным способом расчета цен опционов благодаря общности модели и ее простоте для применения в реальных условиях [1]. Это позволяет использовать модель в качестве эталона для оценки самых сложных финансовых инструментов – опционов. Формула Блэка-Шоулза обладает минимальным количеством параметров, что облегчает ее применение в реальных условиях. Согласно формуле цена европейского опциона колл равна математическому ожиданию выплат по нему. Предположив, что цена базового актива ведет себя как геометрическое фрактальное броуновское движение с показателем ХерстаН, получим:

$$C = \int_X^{\infty} (S - X) f(S) dS = \int_X^{\infty} S f(S) dS - \int_X^{\infty} X f(S) dS ;$$

где С – цена европейского опциона колл; S – цена базового актива; X – цена страйк; f(s) – функция плотности вероятности цены базового актива на момент экспирации опциона.

Несмотря на все достоинства формулы Блэка-Шоулза, которые позволяют ей выступать в качестве эталона для оценки самых сложных финансовых инструментов – опционов, нетрудно заметить, что на практике применение формулы Блэка-Шоулза приводит к усредненному и идеализированному взгляду на поведение базового актива.

Для устранения этих неточностей предлагается использование программного приложения.

Разработанное приложение предназначено для анализа ситуации на фондовых рынках и прогнозирования дальнейших тенденций в изменении цен на опционы и акции. Основным функционалом приложения использует модель Блэка-Шоулза для расчета опционов и отображения графического материала, в частности при построении графиков исторических данных и диаграмм прибыли и потерь.

Интерфейс приложения представляет собой панели обрабатываемых данных:

- список «торговых идей» – символов, которые на сегодняшний день выбраны системой как наиболее выгодные сделки, указаны цены квот, их краткосрочный и долгосрочный тренды. К списку можно применить фильтрацию по медведям и быкам, размерам компании, секторам рынка.

- информация по текущей выбранной позиции – сверху название символа, имя компании, краткосрочный и долгосрочный тренды, другие параметры квоты. Снизу расположены графики изменения цены на опцион.

- система предлагает пользователю 3 сценария действий, каждый из которых проиллюстрирован графиком, соответствующей стоимостью сделки и стратегией. Снизу находится симулятор прибылей и потерь, в котором можно настраивая дату окончания опциона, цену акций и волатильность, предсказывать прибыль или потери в абсолютном и процентном значениях.

Дополнительно отображается общий тренд рынка в текущий момент времени.

Одним из требований к разрабатываемой системе является обновление рыночной информации в режиме реального времени. Для веб-сайтов, использующих протокол HTTP, единственным способом получения новых данных от сервера до недавнего времени было инициирование нового сообщения (проще говоря, послать еще один HTTP запрос). При этом, особенно если используется протокол HTTPS, обеим сторонам необходимо выполнить довольно трудоемкую операцию по установлению и проверке соединения. Для того, чтобы сократить накладные расходы на установление соединения, требуется установить соединение лишь однажды и держать его постоянно открытым.

Однако данный подход идет вразрез с протоколом HTTP. Именно поэтому консорциумом W3C был принят стандарт RFC 6455 (протокол WebSocket) [2]. WebSocket – это протокол полнодуплексной связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени. Данный протокол поддерживается всеми современными веб-браузерами и серверами.

За основу веб-приложения была взята инструментальная среда ASP.NET MVC 5. Одним из преимуществ стека технологий на платформе Microsoft .NET является простота создания веб-сервисов. Так, в платформе ASP.NET можно создавать веб-службы (ASMX). Кроме того, имеется возможность использовать более функциональную в этом плане технологию WCF для создания веб-служб. Но последние версии платформы ASP.NET MVC (да и стека технологий .NET 4.5) предлагают еще один инструмент для создания веб-служб – Web API. Именно таким образом будет налажен интерфейс между клиентской частью приложения и бизнес-логикой.

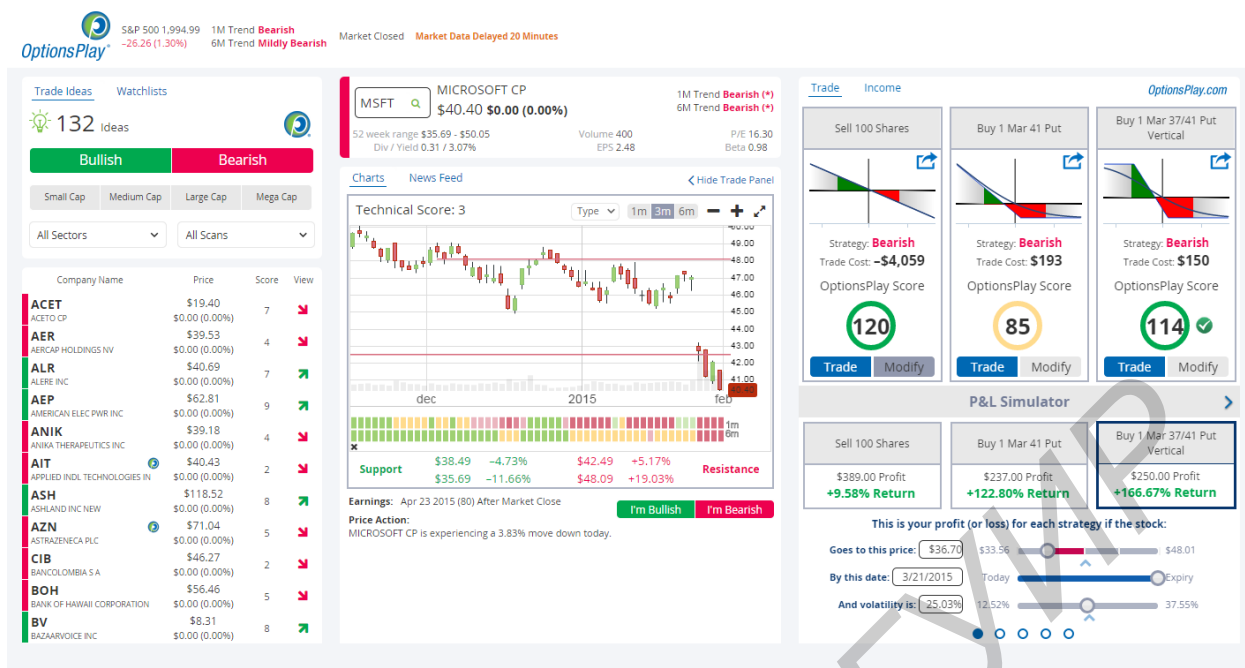


Рисунок 1. Главный экран Web-приложения

Список использованных источников:

1. Макмиллан Л.Г. Опционы как стратегическое инвестирование/Пер. с англ. М.: Евро, 2003 г., 1225 с.
2. TheWebSocketProtocol [Электронный ресурс]. — Режим доступа: <http://tools.ietf.org/html/rfc6455>. — Дата доступа: 21.05.2014.

ПРОГРАММНАЯ ПОДДЕРЖКА АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Иванова В.Л.

Поттосина С.А. – канд. физ.-мат. наук, доц.

В настоящее время программное обеспечение стремительно совершенствуется и приобретает новые формы. Не редкостью сейчас является то, что одно и то же приложение может быть представлено в нескольких формах: оно может быть настольным, а также иметь web- и мобильную версии (например, Skype, Slack). И, несмотря на то, что для тестирования всех форм приложения может использоваться одинаковый или схожий набор тестов, нет универсального инструмента автоматизации тестирования, который бы позволил протестировать все виды и осуществить одновременную проверку всех компонент программного продукта.

Любая система состоит из модулей, которые взаимодействуют друг с другом и с внешними системами. Для успешной реализации проекта важно, чтобы эти модули работали корректно. Поэтому на протяжении всего жизненного цикла разработки программного обеспечения необходимо производить тестирование на различных уровнях. Уровень тестирования определяет то, над чем производятся тесты: над отдельным модулем, группой модулей или системой в целом [1]. Выделяют следующие уровни тестирования, представленные в таблице 1.

Таблица 1 – Уровни и цели тестирования

Уровень тестирования	Цели тестирования
Компонентное (модульное) тестирование	Проверка работоспособности отдельных модулей системы (функции или класса)
Интеграционное тестирование	Нахождение проблем взаимодействия модулей (компонент) системы
Системное тестирование	Проверка полной (интегрированной) системы на соответствие системным требованиям и показателям качества
Приемочное тестирование	1 Определение уровня соответствия системы приемочным критериям 2 Принятие решения о том принимается система или нет

Автоматизацию тестирования как web-, так и мобильных приложений, можно успешно внедрять на всех перечисленных выше уровнях. Однако, не всегда очевидно, какие тесты автоматизировать не стоит. Если в web-приложении, например, отображается какая-то графика, то наверняка будет лучше и проще взглянуть на нее глазами человека, чем изобретать робота. Автоматизировать стоит лишь то, что хорошо