

УДК 004.021

**АРХИТЕКТУРА ПРОГРАММНОЙ РЕАЛИЗАЦИИ АВТОМАТИЗИРОВАННОЙ  
ТОРГОВОЙ СИСТЕМЫ, ИСПОЛЬЗУЮЩЕЙ ТЕХНИЧЕСКИЙ АНАЛИЗ**

А.В. Лукашев

Проведен анализ существующих подходов к созданию автоматизированных торговых систем и их возможностей. Предложен вариант архитектуры программного решения для создания системы торговли ценными бумагами, использующей технический анализ.

**Ключевые слова:** алгоритм, автоматизированная система, технический анализ, индикатор, ценная бумага, архитектура программного обеспечения.

**Введение**

Под автоматизацией процесса торговли ценными бумагами понимается создание механической торговой системы (МТС). Механическая торговая система – это программа (или устройство, как следует из термина «механическая»), которая осуществляет автоматическое выставление и снятие заявок по заранее заложенной в нее логике, в соответствии с торговой системой (торговой стратегией). Также возможно выполнение программой дополнительных функций на усмотрение автора системы – контроль выставленных заявок, мониторинг сделок, анализ торговли с предоставлением графиков и отчетов и т.д. Вместо слова «механическая» было бы уместнее говорить «автоматическая» торговая система, но, в силу сложившихся традиций участники рынка, используют именно этот термин. В настоящее время существует множество торговых терминалов, позволяющих реализовать торговых роботов (MetaTrader, MetaStock, и т.п.). В силу этого можно найти множество решений, продающихся за небольшие (2000–4000 рублей) деньги. Большинство таких роботов имеют довольно простую стратегию, основанную на выявленных, но довольно неустойчивых принципах. Например, довольно популярная торговая стратегия 20/200 pips [1] действительно приносит определенную прибыль, что доказано испытаниями ее на реальных данных. Но она основана лишь на довольно неустойчивой закономерности рынка одного из множества инструментов.

Рассмотрим еще одну торговую стратегию – «14-дневный RSI» [2], основанную на более реальных правилах. В данной системе используется технический индикатор RSI, причем его показания используются не в классической его интерпретации. В результате получаем работающую, но не безопасную систему, которая работает только на больших временных отрезках.

Можно дальше рассматривать общедоступные системы, но их логика сводится к применению одного индикатора или правила, выявленного для определенного инструмента, что дает работающую, но не очень стабильную систему, допускающую большое количество ошибочных срабатываний. С одной стороны это обусловлено тем, что большинство таких систем реализованы с использованием функционально ограниченных терминалов, а с другой (с точки зрения разработки высоко нагруженных систем реального времени) – технически неудачными решениями. Наряду с ними существуют серьезные игроки, такие как Renaissance Technologies, но их решения недоступны для рассмотрения, так как являются закрытой информацией. В настоящей работе рассмотрим разработку МТС, основными отличительными особенностями которой должны стать:

- работа на любом наборе и комбинации индикаторов технического анализа, которая подразумевает наличие единого механизма принятия решений на основе данных индикаторов;
- возможность ее повторного применения на любых площадках, требующая создания набора классов и интерфейсов, описывающих основные сущности рынка;
- возможность работы на любых временных интервалах, которая может быть реализована созданием внутреннего механизма агрегации котировок;
- возможность использования промежуточных котировок, означающая, что система должна производить промежуточный анализ при получении каждой котировки;
- возможность работы по нескольким инструментам одновременно; достигается фильтрацией и передачей котировок по каждому из инструментов отдельному экземпляру системы принятия решений и набору индикаторов;
- возможность работы с любым брокером.

Состав системы

**Агрегированные котировки.** Система будет работать с котировками, агрегированными по временному интервалу, такие производные данные называют «бары» (англ. bar) или свечи (англ. candlestick). Агрегирование данных позволяет хранить и обрабатывать значительно меньшие объемы данных без потери точности расчетов. Во всех современных торговых терминалах для отображения текущего состояния рынка используют именно такой вид данных, поскольку он позволяет воспринимать, куда большие объемы данных, нежели отображение всех котировок, которые могут появляться несколько раз в секунду. Также это позволяет ориентировать стратегию на определенные временные интервалы.

**Индикаторы технического анализа.** Так как разрабатываемая система должна реализовывать стратегию, построенную на индикаторах технического анализа, необходимо предусмотреть абстрактный класс для индикаторов, позволяющий единым образом передавать индикатору биржевые данные и реализовывать собственный алгоритм расчета. Индикаторы в системе будут давать два вида сигналов – на открытие и закрытие сделки, причем с учетом направления (не для всех индикаторов верно утверждение о том, что сигнал об открытии сделки на понижающийся рынок автоматически означает сигнал на закрытие сделки в сторону повышения). Помимо алгоритма расчета и генерации сигналов, индикатор должен хранить минимально необходимую ему часть исторических данных, поскольку большинство индикаторов для вычисления нового значения требуют данных за прошлый период. В случае возникновения изменения ситуации на рынке с точки зрения конкретного индикатора сигнал должен быть передан в систему принятия конечного решения.

**Обратная связь от индикатора и принятие решений.** Для принятия решений в системе должен существовать класс, получающий данные об изменении прогноза индикатора и на основании всех прогнозов принимающий решение об открытии или закрытии позиции.

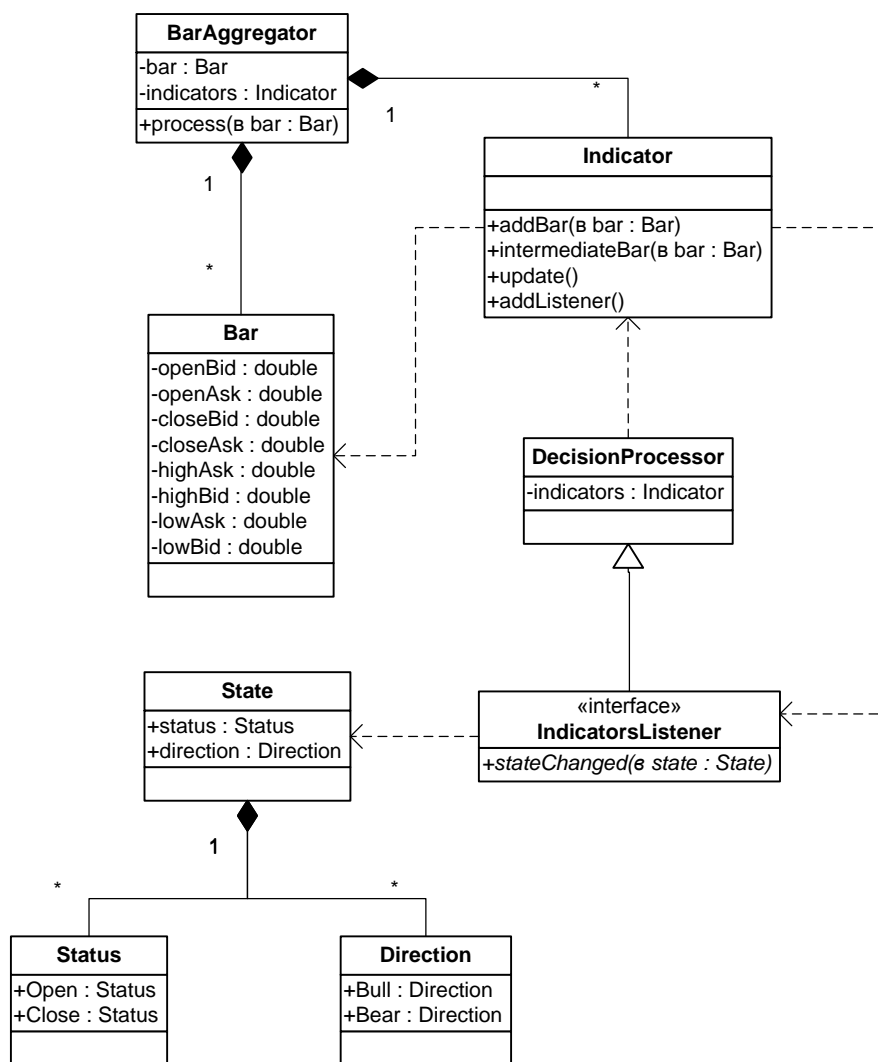


Рис. 1. UML диаграмма части системы, отвечающей за генерацию сигналов стратегии

Поскольку система может использовать множество индикаторов, производить действие нужно лишь в случае, когда все индикаторы системы дают одинаковый сигнал. В таком случае необходимо передать данные по сделке брокеру. Поскольку стратегия в данном случае является источником данных для индикатора, в соответствии с концепциями GOFPatterns [3] следует ввести интерфейс слушателя индикатора, который будет реализован стратегией. Данный шаблон (pattern) называется разрывом зависимости (DependencyInjection) и используется в случае необходимости взаимно связать две сущности, при этом не создавая циклическую взаимозависимость. Таким образом, получаем структуру, отраженную на рис. 1.

В структуре присутствуют следующие классы:

- State – отражает рекомендуемое действие в соответствии с прогнозом движения цены:
  - Status – открытие/закрытие;
  - Direction – направление, по которому делается прогноз;
- BarAggregator – рассчитывает агрегированную котировку за заданный интервал;
- Bar – агрегированные котировки;
- Indicator – индикатор;
- DecisionProcessor – принимает решение о конечном действии;
- IndicatorsListener – организует обратную связь от индикатора к DecisionProcessor.

**Связь с брокером и отслеживание сделок.** Поскольку в списке требований к системе была заявлена возможность работы с любым брокером, введем дополнительный уровень абстракции BrokerGateway, который будет предоставлять возможность выставить ордера и получать информацию об их обновлении. Реализация будет уже зависеть от конкретного брокера, в большинстве случаев это будет использование FIX-протокола, в некоторых случаях (например, брокер FXCM) это будет JavaAPI.

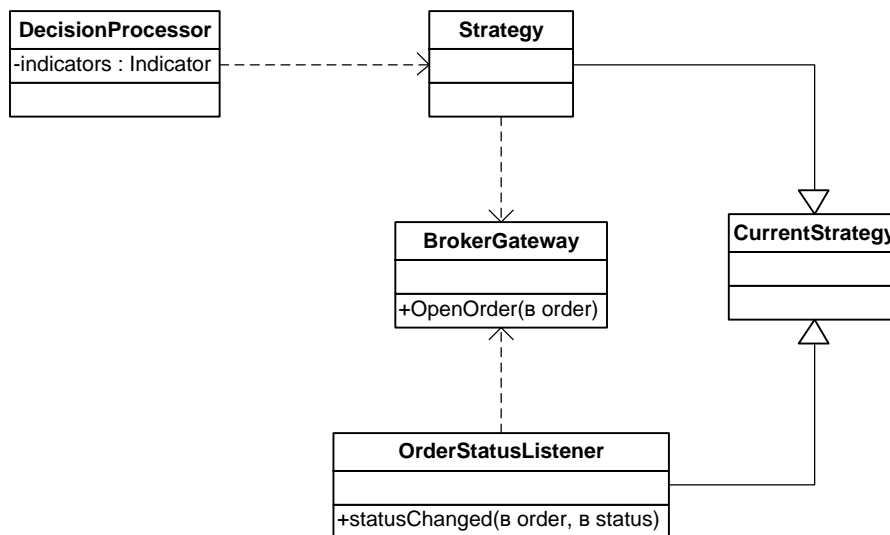


Рис. 2. Подсистема создания и обработки сделок

На рис. 2 представлен набор и связи сущностей, отвечающих за работу со сделками:

- Strategy – интерфейс стратегии;
- BrokerGateway – обеспечивает связь с брокером;
- OrderStatusListener – интерфейс, отслеживающий обработку изменений ордеров;
- CurrentStrategy – реализация стратегии.

При построении конечной системы следует создать отдельный экземпляр стратегии на каждый символ, связав его с единым экземпляром шлюза к брокеру, это позволит реализовать поддержку одновременной работы со многими символами. Общая диаграмма представлена на рис. 3.

### Технологии

Для реализации системы автором был использован Springframework, который позволяет создать конфигурируемое приложение с минимальными усилиями, а также предоставляет шаблонные классы для использования других технологий (например, HibernateTemplate). Для работы с базой данных (используется для сохранения исторических данных) был применен Hibernate, поскольку позволяет избежать написания SQL-запросов.

В качестве дальнейшего развития системы планируется применение искусственных нейронных сетей в двух различных вариантах – для системы принятия решений и в качестве индикатора. На данный

момент разрабатывается алгоритм самообучения, который будет обучать систему в течение всего цикла жизни. Главным результатом ожидается получение интеллектуальной системы, способной подстраиваться под долгосрочные изменения рынка.

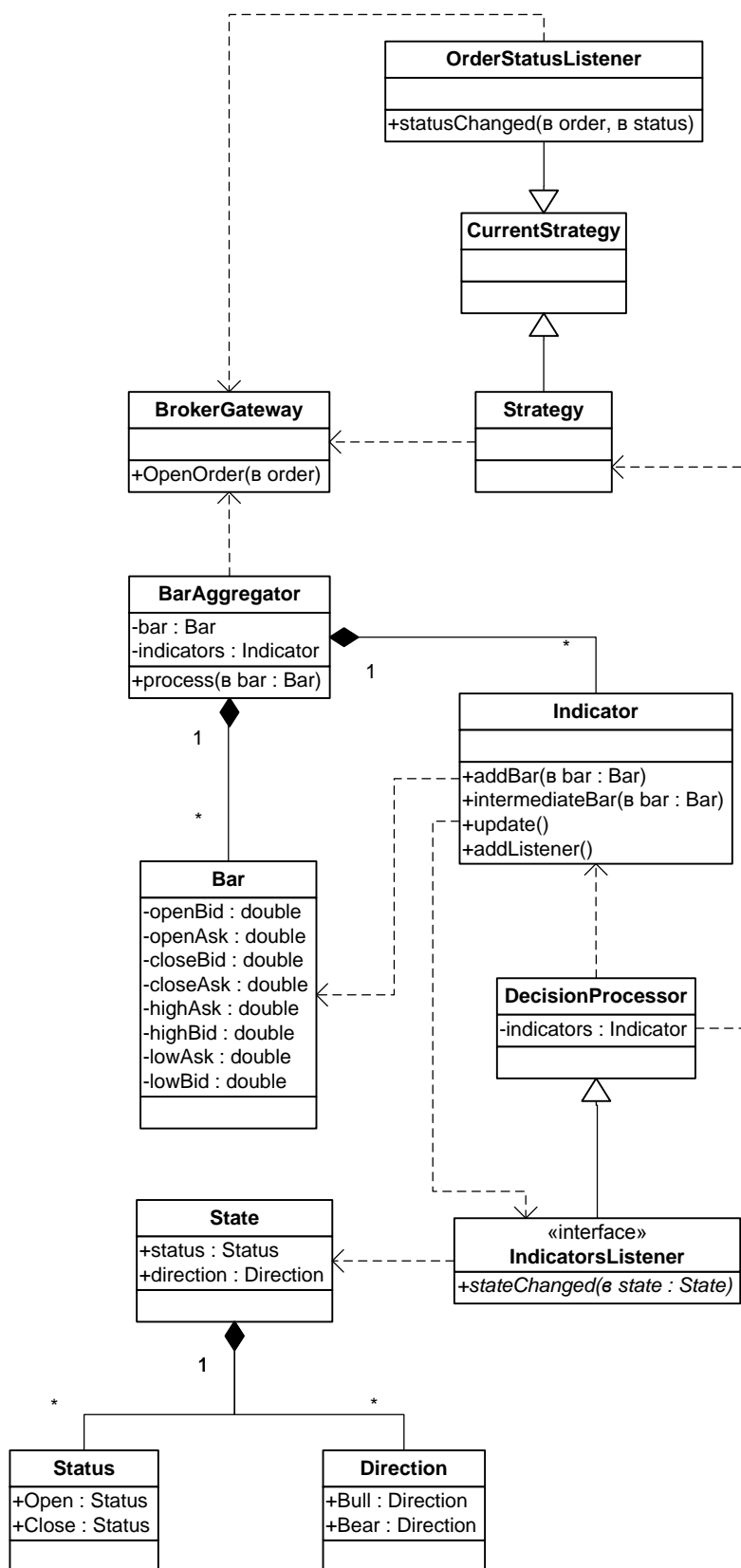


Рис. 3. Диаграмма системы

### **Заключение**

В работе были рассмотрены доступные для экспертизы торговые системы. Исходя из предлагаемых реализаций, был сделан вывод, что они не позволяют вести безопасную внутридневную торговлю и работают на основании математически и фундаментально не подтвержденных правил. Использование комплекса инструментов технического анализа позволяет перекрыть основные требования и создать эффективную систему для работы на внутридневном рынке. В результате анализа проблемы были сформулированы требования, на основании которых разработан архитектурный подход к созданию автоматизированных систем, позволяющий полностью покрыть выставленные требования. Также следует отметить, что применение предложенного подхода не ограничено рынком Forex, при некоторых корректировках его можно перенести на фондовый рынок, а также рынок деривативных продуктов, которые, благодаря более высокой волатильности, позволят вести более прибыльную торговлю. В дальнейшем перспективным решением может быть применение нейронных сетей. С их применением можно создать самообучающийся индикатор достижения локального экстремума цены и силы тренда, а также модуль агрегации данных, поступающих от различных индикаторов. При условии создания эффективного алгоритма самообучения, такая система может иметь серьезные конкурентные преимущества, такие как универсальность и самоорганизация.

### **Литература**

1. Смирнов П.В. Механическая торговая система «20/200 pips». Результаты торговли на 2010 год [Электронный ресурс]. – URL: <http://www.autoforex.ru/lab/otchet-o-testirovanii-2010-god-20-200-v1/otchet-o-testirovanii-2010-god-20-200-v1.php>, свободный. Яз. рус. (дата обращения: 20.12.2010).
2. Медведев М. Механическая торговая система по RSI от Чака Лебо [Электронный ресурс]. – URL: <http://pisali.ru/medvedev70/3997/>, свободный. Яз. рус. (дата обращения: 25.12.2010).
3. Steven John Metsker. Design Patterns Java(TM) Workbook. – Addison-Wesley Professional. – 2002. – 496 с.

*Лукашев Александр Владимирович* – Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», аспирант, [lukashev.alexander@gmail.com](mailto:lukashev.alexander@gmail.com)

УДК 629.7.07

## ЧИСЛЕННЫЙ АНАЛИЗ ВЛИЯНИЯ ТОЧНОСТИ ПРОГНОЗА ПАССАЖИРСКОГО СПРОСА НА ЭФФЕКТИВНОСТЬ ПРОДАЖ АВИАБИЛЕТОВ С УЧЕТОМ СВЕРХЛИМИТНОГО БРОНИРОВАНИЯ

К.А. Мозговая, М.В. Яблочкина, Г.М. Фридман

Представлены числовые результаты компьютерной симуляции процесса продажи авиабилетов с учетом сверхлимитной емкости и процесса предполетной регистрации пассажиров с компенсацией в случае отказа от предоставления места в самолете. Выполнен анализ влияния точности прогноза спроса на эффективность различных стратегий продаж. Установлено, что учет сверхлимитной емкости может привести к увеличению общей прибыли, связанной с продажей билетов, на 5–20% в зависимости от точности прогноза спроса.

**Ключевые слова:** прогноз пассажирского спроса, сверхлимитная (виртуальная) емкость воздушного судна, неявка на регистрацию, возврат билетов, компенсация.

### Введение

Одной из самых старых и, возможно, самых эффективных с финансовой точки зрения стратегий управления доходами является практика сверхлимитных продаж (overbooking). Идея продавать большее, чем имеющееся в наличии, количество продукта, возникает в условиях предварительного резервирования с возможностью отказа от него до наступления момента реализации и (или) наличия неявок покупателей уже в момент реализации. Ярким примером служит индустрия авиаперевозок, где бронирование билетов на рейсы авиакомпаний начинается задолго до времени выполнения рейса. Отказы (cancellation), в зависимости от условий, установленных авиакомпанией, составляют до 50% всех броней, и нередки случаи, когда пассажиры по разным причинам не появляются к моменту вылета (no-show). Чтобы не упустить дополнительную прибыль и не возить незанятые места на рейсах, авиакомпании практикуют сверхлимитные продажи – продажи сверх емкости воздушного судна (ВС). Это приводит к возникновению рисков, связанных с необходимостью выплачивать значительные компенсации в случае, когда число зарегистрировавшихся на вылетающий рейс пассажиров оказывается больше емкости назначенного на этот рейс ВС и соответственно к падению прибыли от продажи авиабилетов. Следовательно, любая авиакомпания заинтересована в минимизации этих рисков за счет определения виртуальной емкости ВС – оптимального с точки зрения общей прибыли количества авиабилетов, которые следует продавать на рейс с назначенным ВС известной физической емкости.

Задача определения виртуальной емкости ВС рассматривалась во многих статьях и книгах, обзор которых можно найти в [1, 2]. Также отдельно упомянем работы [3, 4].

Прогнозы общего пассажирского спроса, неявок пассажиров к регистрации и возвратов авиабилетов служат основными исходными данными к решению задачи. В представленной работе приведены числовые результаты, дающие возможность оценивать влияние их точности на эффективность принятой стратегии сверхлимитных продаж. Все расчеты выполнены при помощи созданного авторами компьютерного симулятора деятельности авиакомпании.

### Математические постановки оптимизационных задач определения виртуальной емкости рейса и пределов бронирования

Положим, что на некоторый рейс продаются авиабилеты (продукты) с ценами  $f_j$ ,  $j=1, \dots, n$ . Предполагается (прогнозируется), что спрос  $D_j$  на каждый продукт  $j$  представляет собой случайную величину, распределенную по нормальному закону распределения со значениями среднего  $\mu_j$  и среднеквадратического отклонения  $\sigma_j$ ,  $j=1, \dots, n$ .

При известных средней цене  $P$  билета на рейс, физической емкости  $Cap$  назначенного на рейс ВС и прогнозе осредненной вероятности  $p$  появления пассажира, купившего билет, к предполетной регистрации (так называемый show rate) задача состоит в нахождении величины  $B$  – виртуальной емкости, при которой достигает максимума выражение  $R(B)$ :

$$R(B) = B \times T - C_R \times T \times B \times (1 - p) - F(B, p, Cap) \rightarrow \max, \quad (1)$$

где  $F(B, p, Cap)$  – компенсационная функция, определяющая размер ожидаемых потерь, связанных с выплатой компенсаций «лишним» пассажирам на рейсе в случае продажи  $B$  билетов на ВС емкостью  $Cap$ ; при этом  $F(B, p, Cap) = 0$  для  $B \leq Cap$ . Различные способы расчета компенсационной функции обсуждены в работах [2, 5]. Коэффициент  $0 \leq C_R \leq 1$  показывает долю средней цены билета  $P$ , возвращаемую пассажиру в случае его неявки к регистрации. Среднюю цену билета на данном рейсе можно, например, вычислить по формуле: