

УДК 004.4:339.1

ПРОГРАММНОЕ СРЕДСТВО ПОИСКА ПЕРСПЕКТИВНЫХ НАПРАВЛЕНИЙ ДЛЯ ТОРГОВЛИ НА МАРКЕТПЛЕЙСАХ

Крино О.С.

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

Научный руководитель: Косарева Е.М. – магистр, ассистент кафедры ПИКС

Аннотация. Экспериментально исследовано функционирование программного средства поиска перспективных торговых направлений в условиях обработки больших объемов данных. Установлено, что снижение достоверности формируемых рекомендаций наступает при увеличении доли некорректных записей в *JSON*-пакетах свыше 7,4 % от общего объема выборки. Предложена методика контроля бизнес-логики программного средства, основанная на верификации функциональных блоков обработки данных. Разработаны структурные схемы взаимодействия серверной и клиентской частей программного средства для обеспечения точности бизнес-анализа.

Ключевые слова: маркетплейс, бизнес-анализ, программное средство, бизнес-логика, *REST*-архитектура, автоматизация аналитики

Введение. Современные требования к автоматизации аналитических процессов, связанные с преобразованием больших объемов данных с маркетплейсов в обоснованные рекомендации, выдвигают задачу контроля качества бизнес-логики программных средств на одно из первых мест. Программные решения, имеющие распределенную *REST*-архитектуру, обладают повышенной чувствительностью к структуре и полноте входных данных, что требует поиска дефектов в алгоритмах ранжирования на ранних этапах [1]. В данной статье автором показано, что ухудшение качества выходных рекомендаций программного средства наступает при накоплении статистических аномалий, что может привести к принятию ошибочных стратегических решений по развитию торговли.

Основная часть. При контроле функционирования программного средства (ПС), предназначенного для поиска перспективных направлений, необходимо решить две задачи:

- обеспечить достаточную полноту автоматизированного тестирования алгоритмов;
- определить функциональные блоки (ФБ), ответственные за возникновение логических ошибок в аналитических отчетах.

Для обеспечения чистоты эксперимента и проверки корректности взаимодействия компонентов системы процесс анализа был разделен на две области, в соответствии с рисунком 1.

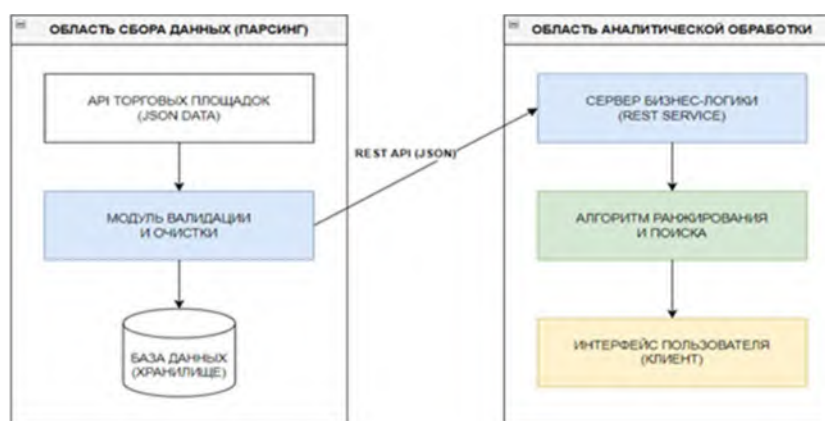


Рисунок 1 – Схема взаимодействия областей сбора данных и аналитической обработки

На представленной схеме (рисунок 1) визуализирована высокоуровневая архитектура программного средства, спроектированная согласно принципам распределенных систем. Взаимодействие компонентов организовано через разделение зон ответственности на две ключевые области: сбор данных и их глубокую аналитическую обработку.

Опишем каждую из областей:

1 Область сбора данных (Парсинг). Данная область отвечает за первичную интеграцию с внешними источниками. Процесс начинается с обращения к *API* торговых площадок, откуда поступают массивы данных о продажах, остатках и ценах в формате *JSON*. Ключевым звеном здесь является модуль валидации и очистки, который выполняет предварительную фильтрацию «сырых» данных. Его задача – это исключение аномальных всплесков (например, самовыкупов или технических ошибок площадки), которые могут исказить итоговый результат. После верификации структурированная информация сохраняется в базе данных (хранилище), обеспечивая историчность данных для последующего трендового анализа.

2 Область аналитической обработки. Эта область представляет собой «ядро» системы, реализованное в виде сервера бизнес-логики (*REST Service*). Использование архитектурного стиля *REST* позволяет изолировать вычислительные процессы от интерфейсной части. Основным функциональным элементом здесь является алгоритм ранжирования и поиска, который на основе математических моделей преобразует сухие цифры в конкретные показатели перспективности рыночных ниш. Конечным этапом является передача обработанных данных в интерфейс пользователя (клиент), где бизнес-аналитик получает визуализированные рекомендации для принятия управленческих решений.

Связующим звеном между областями выступает *REST API (JSON)*. Такое решение обеспечивает масштабируемость системы: модуль сбора данных может работать независимо от аналитического модуля, а передача данных в универсальном формате *JSON* гарантирует высокую скорость обработки и совместимость компонентов. Данная структура полностью соответствует техническим требованиям дипломного проекта по созданию распределенного программного обеспечения с четким разделением клиентской и серверной частей.

Испытуемое программное средство было представлено в виде ряда функциональных блоков, реализованных в архитектурном стиле *REST*. Для тестирования ФБ был разработан набор тестовых программ (ТП), обеспечивающих последовательность верификации бизнес-логики в составе ПС (таблица 1):

Таблица 1 – Последовательность тестирования ФБ МК

№ тестовой программы	Область тестирования
ТП1	верификация корректности парсинга <i>JSON</i> -данных серверной частью
ТП2	тестирование алгоритмов ранжирования в модуле бизнес-логики

На рисунке 2 представлена логическая последовательность работы аналитического модуля, обеспечивающая преобразование входных массивов данных в обоснованные бизнес-рекомендации. Алгоритм включает в себя несколько критических стадий обработки:

1 Инициализация и загрузка. Процесс начинается с получения *JSON*-пакетов через *REST*-интерфейс. На этом этапе программное средство подготавливает среду для обработки данных, извлекая необходимые параметры (цены, объемы продаж, количество конкурентов).

2 Контроль целостности и верификация. Данный этап является ключевым для обеспечения достоверности анализа. Программа проверяет структуру *JSON* и полноту данных. Согласно проведенным исследованиям, именно на этом этапе выявляется «зашумленность» выборки. Если уровень ошибок превышает допустимый предел, система инициирует цикл дозагрузки данных для предотвращения искажения итоговых индексов.

3 Вычисление аналитических показателей. После успешной валидации модуль переходит к расчету интегральных показателей. Тренд спроса: определение динамики продаж за ретроспективный период. Уровень конкуренции: анализ концентрации продавцов в нише. Индекс прибыльности: сопоставление потенциальной выручки и затрат. Формирование

ивывод рекомендаций. На финальном этапе алгоритм классифицирует ниши по степени перспективности и формирует отчет, который передается в интерфейс пользователя (клиентскую часть).

Общий алгоритм функционирования модуля аналитики, отвечающего за поиск перспективных ниш, представлен в виде блок-схемы (рисунок 2). Данная схема отражает логику преобразования «сырых» данных о продажах в конкретные рекомендации.



Рисунок 2 – Алгоритм функционального аналитического модуля программного средства

Контроль функционирования блоков (ФБ) программного средства осуществлялся путем циклической синхронизации с *API* торговых площадок. После прохождения всего набора тестовых сценариев производился глубокий анализ отклонения расчетных показателей от эталонных значений.

Установлено, что выход из строя или некорректный результат работы «модуля аналитики» чаще всего обусловлен спецификой организации бизнес-логики. Данный модуль строится на базе процедур агрегации, обладающих не только режимами хранения информации, но и программными фильтрами для отсекаания аномальных всплесков продаж (самовыкупов). Реализация таких механизмов верификации на стороне сервера делает аналитический блок чувствительным к пропускам или искажениям данных, в отличие от простых транспортных модулей системы.

Заключение. Выполнен анализ влияния качества входных данных на результаты поиска перспективных направлений в разработанном программном средстве. Впервые установлено, что в аналитическом блоке ПС при накоплении ошибок в *JSON*-структурах свыше 7,4 % происходят критические искажения в отчетах, что ведет к некорректному выполнению запрограммированных функций. Определено, что изменения в точности анализа обусловлены

отсутствием эффективной фильтрации аномалий на этапе первичной обработки данных. Предложена методика функционального контроля ПС, основанная на разделении процессов сбора и анализа данных, что позволяет своевременно выявлять логические дефекты в программном коде.

Список литературы

1. Kleppmann, M. *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems* / Martin Kleppmann. – Sebastopol : O'Reilly Media, 2017. – 616 p. – ISBN 978-1-449-37332-0.
2. Fielding, R. T. *Architectural Styles and the Design of Network-based Software Architectures* / Roy Thomas Fielding // *Slovak international scientific journal*. – 2021. – Vol. 2, N 48. – Pp. 14–29
3. Алексеев, В. Ф. Проектирование и разработка распределенных информационных систем на базе REST-архитектуры / В. Ф. Алексеев // *Вестник БГУИР*. – 2022. – № 4. – С. 45–52.
4. Вигерс, К. *Разработка требований к программному обеспечению* / К. Вигерс, Д. Бумти. – М. : БХВ-Петербург, 2019. – 736 с. – ISBN 978-5-9775-3348-5.

UDC 004.4:339.1

SOFTWARE TOOL FOR FINDING PROMISING DIRECTIONS FOR TRADING ON MARKETPLACES

Krino O.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Kosareva E.M. –master of tech. sciences, assistant of the department of ICSD

Annotation. The functioning of a software tool for finding promising trading directions under conditions of processing large volumes of data has been experimentally investigated. It was found that a critical decrease in the reliability of the generated recommendations occurs when the share of incorrect or incomplete records in JSON packets exceeds 7.4% of the total sample volume. A method for controlling the business logic of a software tool based on the verification of functional blocks is proposed. Schemes for the interaction of data collection and processing areas, as well as an algorithm for the analytical module's operation, have been developed.

Keywords: marketplace, business analysis, software tool, business logic, REST architecture, analytics automation