СИНТЕЗ ПРОГРАММНОГО АЛГОРИТМА И АРХИТЕКТУРЫ ПРИЛОЖЕНИЯ ПО АВТОМАТИЗИРОВАННОМУ СБОРУ ИНФОРМАЦИИ В СЕТЕВЫХ АГРЕГАТОРАХ

К.В. Портнов, *к.т.н., доцент, Самарский государственный технический университет, sk7@mail.ru;*
М.А. Фошин, *Самарский государственный технический университет, valp@mail.ru.*

УДК 004.9

Аннотация. Настоящая статья посвящена техническим вопросам, связанным с автоматизированным поиском данных на страницах *web*-браузеров и трансформацией этих данных в необходимые формы. Данная проблема актуальна для множества сфер, где необходимо собирать данные для их последующей обработки. В качестве прикладной сферы для разработки прототипа алгоритма и программы сбора данных выбрана оценочная деятельность. Авторами проведен системный анализ предметной области, описаны процессы проведения оценки, выделены наиболее трудоемкие процессы, которые связаны со сбором данных с веб-страниц и требуют первичной автоматизации. Разработано приложение для сбора данных для целей оценки недвижимости.

Ключевые слова: интерфейс браузера; сбор данных с веб-страниц; парсинг данных; автоматизация оценочной деятельности; алгоритм сбора данных; системный анализ процессов оценки недвижимости; оценка квартир.

SYNTHESIS OF SOFTWARE ALGORITHM AND ARCHITECTURE OF APPLICATION FOR AUTOMATED COLLECTION OF INFORMATION IN NETWORK AGGREGATORS

K.V. Portnov, *candidate of technical science, associate professor, Samara State Technical University;*
M.A. Foshin, *Samara State Technical University.*

***Annotation*.** This article is devoted to technical issues related to automated data search on web browser pages and transformation of this data into the necessary forms. This problem is relevant for many areas where it is necessary to collect data for subsequent processing. Assessment activities were chosen as the applied area for developing a prototype algorithm and data collection program. The authors conducted a systematic analysis of the subject area, described the assessment processes, identified the most labor-intensive processes that are associated with collecting data from web pages and require primary automation. An application has been developed to collect data for real estate valuation purposes.

Keywords: browser interface; data collection from web pages; data parsing; automation of valuation activities; data collection algorithm; system analysis of real estate valuation processes; apartment valuation.

Введение

Интернет становится местом концентрации большого количества информации, которая может быть использована в различных отраслях. Несмотря на открытость некоторых источников, отсутствие *API*-интерфейса для получения к ним доступа составляет проблему для их автоматизированной обработки. На большинстве открытых ресурсов данные в них представлены в форме, мало

пригодной для программной обработки, а некоторые ресурсы специально ограничивают даже ручное копирование открытых данных (объявлений, отчетов, статистики). В подобных случаях становится актуальной задача сбора данных посредством преобразования «аналоговых» данных – растровых картинок, защищенного текста – в данные, необходимые для обработки или решения практических задач.

Задачи автоматизированного сбора данных актуальны для составления экономических, статистических, социологических отчетов, проведения различного рода исследований, накопления статистических данных и т.п.

Одной из таких задач является задача по сбору данных на агрегаторах недвижимости для формирования перечня аналогичных объектов при проведении оценки объектов недвижимости. Оценка объектов недвижимости играет ключевую роль в различных аспектах бизнеса и инвестиций, таких как сделки купли-продажи, ипотечное кредитование, управление активами, финансовая отчетность и другие. Время, необходимое для изготовления одного отчета об оценке объекта недвижимости, может значительно варьироваться в зависимости от различных факторов, таких как тип недвижимости, объем работы, сложность объекта, доступность информации, методы оценки, требования заказчика и другие. В среднем, оценка недвижимости может занять от нескольких дней до нескольких недель.

Цель работы – разработка математического, алгоритмического и программного обеспечения для парсинга сетевых страниц на примере оценочной деятельности, способного к адаптации под другие задачи сбора однотипных данных с *html*-страниц.

Анализ процессов в оценочной деятельности

Процесс оценки объекта недвижимости представляет собой однообразный процесс по изготовлению типовых отчетов об оценке объекта со стандартными разделами.

Первым делом для автоматизации оценочной деятельности необходимо провести системный анализ процессов для построения модели действия оценщика, для понимания возможности программной автоматизации данной процедуры.

В общем виде процесс подготовки отчета об оценке выглядит как однообразный процесс составления отчета об оценке на основании документов, полученных от заказчика, результатов осмотра и других данных. Контекстная диаграмма процесса оценки недвижимости в общем виде представлена на рис. 1.

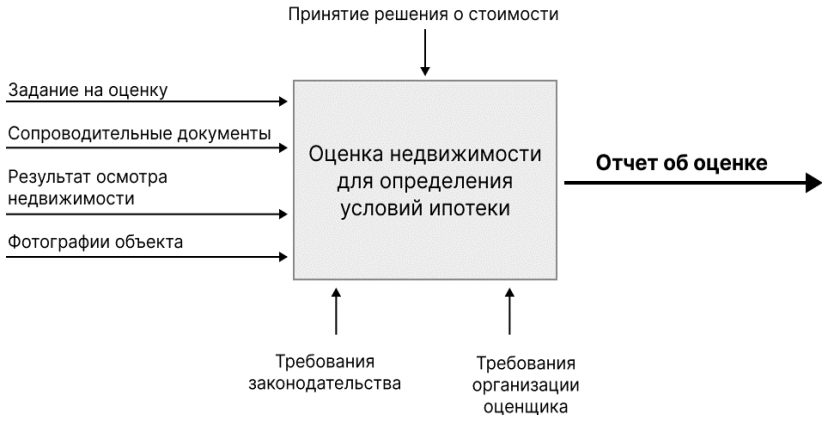

Рисунок 1

Тем не менее, несмотря на шаблонность большинства процедур процесса оценки, эта деятельность в настоящее время осуществляется с помощью ручного труда оценщика, заполняющего *DOC*-шаблон.

Информацию, содержащуюся в отчете об оценке, можно свести к следующим классам:

- типовые формы (титульный лист и т.п.);
- типовая стандартная информация (сведения об оценщике, сведения об организации, в которой работает оценщик, принятые при проведении оценки объекта допущения, применяемые стандарты при проведении оценки, обоснование выбора подходов к оценке объектов, описание подходов к оценке объектов, социально-экономическое развитие региона и т.п.);
- индивидуальная информация (информация о заказчике, информация об объекте недвижимости и т.п);
- динамическая информация – информация, которая меняется (анализ рынка недвижимости, таблица аналогичных объектов).

Анализ информации, содержащейся в отчете об оценке объекта недвижимости, позволил произвести ее классификацию. Классификация информации представлена на рис. 2.

Рисунок 2

Как видим, большинство информации остается достаточно однообразной и из 100 страниц, содержащихся в отчете, 80% информации является типовой информацией, которая не сильно меняется из-за отчета в отчет. Однако копирование данной информации не составляет высоких трудозатрат, поэтому автоматизация заполнения шаблонных документов может быть одним из будущих направлений развития данной работы.

Анализ процесса составления отчета позволил выделить следующие процедуры:

1. Прием документов.
2. Выездной осмотр объекта недвижимости.
3. Формирование *DOC*-шаблона отчета об оценке объекта недвижимости.
4. Заполнение типовых разделов отчета об оценке (класс общей информации).

5. Заполнение раздела с индивидуальными характеристиками объекта (класс специальной информации).

6. Поиск объектов-аналогов и сравнительный анализ объектов (класс анализа рынка).

7. Принятие решения об окончательной стоимости.

8. Компиляция (сборка) отчета как документа.

Наиболее трудоемкой и долговременной процедурой является поиск оценщиком на разных интернет-агрегаторах недвижимости аналогичных объектов и ручное копирование характеристик найденных объектов с последующим переносом информации в отчет об оценке (информация в данном случае представлена таблицей аналогов с основными характеристиками и *url*-ссылкой на источник).

После того как найден аналогичный объект, необходимо произвести поочередно копирование следующих его характеристик:

- наименование объекта недвижимости;
- адрес объекта недвижимости;
- цена объекта недвижимости;
- общая площадь объекта недвижимости;
- этаж;
- материалы стен;
- краткое описание;
- *URL*-ссылка;
- и прочие характеристики.

Перечисленные данные в типовом объявлении (например, с сайта *Avito*) размещены в разных местах, их необходимо поочередно выделить, скопировать и поместить в таблицу. Данная процедура на копирование характеристик одного объекта-аналога может занимать от 1 до 5 мин. А количество аналогов для высокой адекватности оценки должно быть достаточно большим (20-30). Указанный процесс настолько однообразен, что при правильной постановке задачи может быть частично автоматизирован программно. Таким образом, задача сводится к возможности на нужной *HTML*-странице объекта-аналога указать программе расположение требуемых характеристик (создание шаблона) и на последующих страницах-аналогах произвести автоматизированное копирование необходимой информации и дальнейший перенос этих данных в таблицу аналогичных объектов недвижимости.

Авторами предлагается программный алгоритм автоматизации поиска и переноса из агрегаторов данных об объектах-аналогах. Процедура не может быть полностью автоматической, так как критерий схожести объектов очень субъективен и опирается на множество факторов, а решение, какой объект считать аналогом, принимается самим экспертом-оценщиком.

Частичная автоматизация будет заключаться в создании программы и алгоритма, которые позволяют для конкретного агрегатора создавать шаблон расположения необходимой информации в объявлении-аналоге, с дальнейшим ее перемещением в конечную таблицу-аналогов, являющуюся результатом работы проектируемого программного продукта.

Подобная организация процесса позволит существенно сократить временные затраты оценщика, что обеспечивает увеличение производительности труда.

Синтез алгоритма и архитектуры приложения

Характеристики об объекте-аналоге находятся на сайте-агрегаторе и представляют из себя элементы *HTML*-разметки страницы. В большинстве популярных браузеров существуют специальные инструменты, предназначенные для веб-разработчиков. Среди функционала данных инструментов присутствует возможность посмотреть разметку страницы и все расположенные на ней элементы в формате *HTML*.

Получив полную разметку страницы и зная некоторую информацию о элементе с интересующей характеристикой объекта (например, текст, сопровождающий значение характеристики), можно определить «маркеры», чтобы программно-алгоритмически получить эти значения для последующего переноса в таблицу аналогов.

Получив значения характеристик с выбранной страницы, необходимо каким-либо образом отправить в файл с электронной таблицей. Проблема заключается в том, что возможности браузерных расширений ограничены в целях безопасности. Для этого создана дополнительная программа для переноса скопированных характеристик объектов в *xlsx* файл. Для пересылки данных можно воспользоваться технологией *WebSocket*, что позволит наладить связь между браузерным расширением и десктоп-приложением.

Таким образом, конечный программный продукт будет состоять из двух частей:

1) Браузерное расширение (сбор данных со страницы объявления о продаже объекта недвижимости и переправка их через веб-сокет в десктоп-приложение);
2) Десктоп-приложение (принятие отправленных данных и запись их в *xlsx*-таблицу объектов-аналогов).

Полученный процесс взаимодействия разных частей приложения изображен на рис. 3:

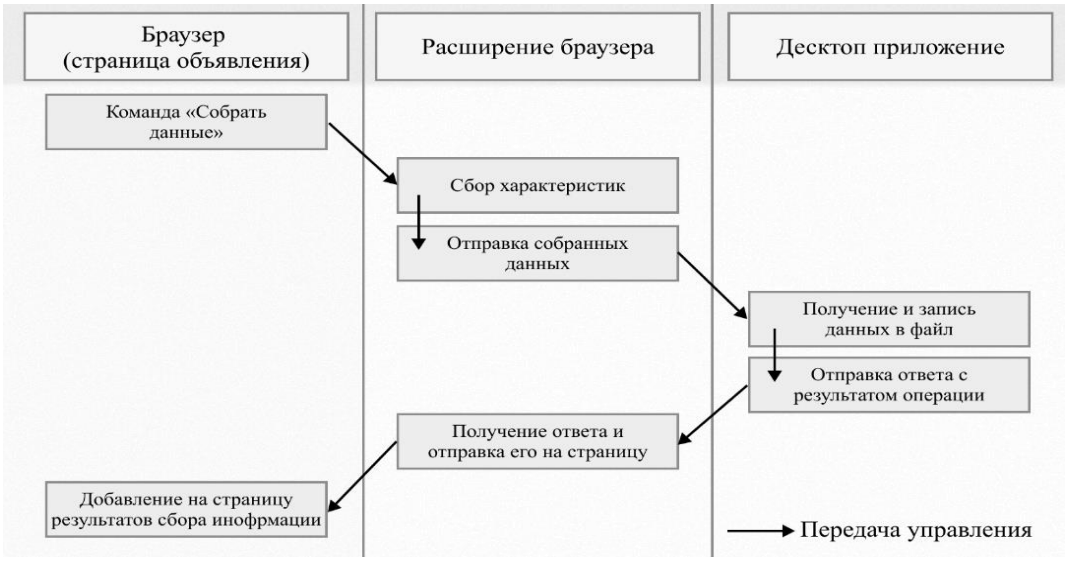

Рисунок 3

Моделируя процесс выгрузки информации с разных сайтов агрегаторов недвижимости, нельзя не отметить важную проблему: разные сайты имеют разную разметку и оформление информации о объектах. Следовательно, нельзя заранее определить, где на разметке страницы будут располагаться необходимые данные, так как на разных сайтах разметка будет отличаться. Тем не менее, определив расположение элемента однажды, не составит труда сохранить путь к нему для

последующего использования при повторных посещениях текущей страницы и аналогичных страниц сайта.

Для реализации подобного подхода введем понятие «Шаблон». Шаблоном будет являться объект, указывающий какие данные необходимо выгрузить в таблицу, где эта информация располагается на разметке страницы и для какого сайта актуален сам шаблон. Шаблон будет создаваться самим оценщиком.

Возможность создания шаблона лучше всего реализовать непосредственно в браузере с помощью создания браузерного расширения, интегрировав необходимый интерфейс на страницу с объявлением, чтобы оценщик мог выбрать все интересующие его элементы, не прибегая без необходимости к непосредственному анализу разметки страницы. Созданные шаблоны можно сохранять для последующего использования прямо в браузере. Это возможно благодаря доступу к *API* хранилища, предоставляемом браузером всем расширениям.

В результате общий процесс выгрузки информации для конечного пользователя будет состоять из этапов, отраженных на рис. 4.

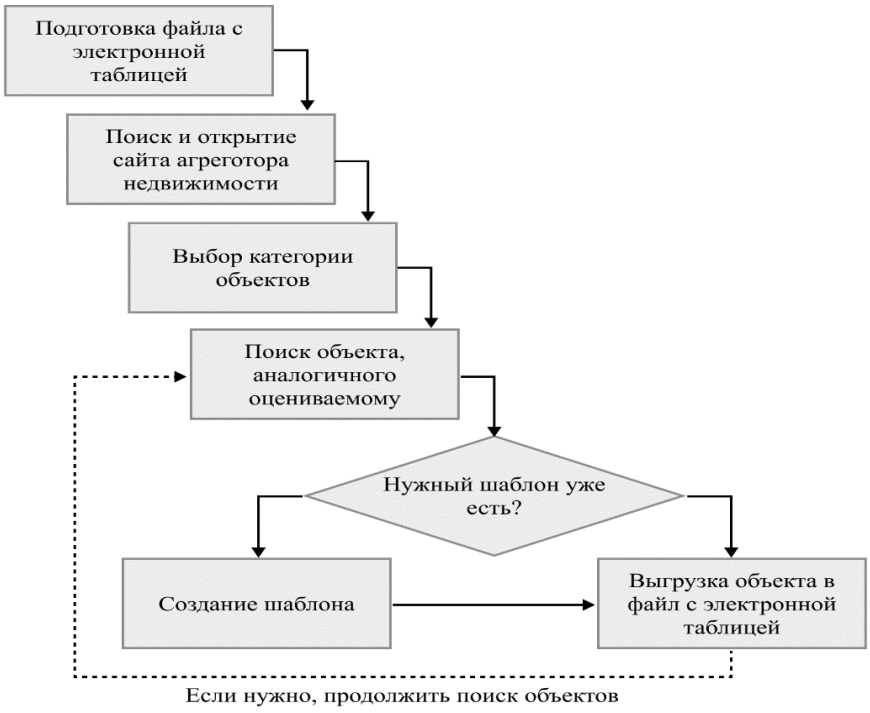

Рисунок 4

Браузерное расширение должно реализовывать следующий функционал:

- возможность создавать шаблоны сбора информации;
- возможность просматривать, редактировать и удалять созданные шаблоны;
- возможность использовать созданные шаблоны для сбора информации со страниц сайтов;
- отправка собранной со страницы информации через веб-сокет в десктоп-приложение;

Ориентируясь на требования к функционалу расширения, введем необходимые интерфейсы. В качестве отправной точки, из которой можно будет получить доступ ко всем функциональным возможностям, станет *Popup*-окно. Находясь на какой-либо интернет странице, из него можно вызывать окно создания и редактирования шаблонов или непосредственного сбора данных. Созданные

шаблоны можно просмотреть на отдельной странице «*allTemplatesPage*», перейти на которую можно все из того же поп-ап окна.

Каждый из фронта энд-скриптов ответственен исключительно за свои функциональные возможности и лишь отражает текущее состояние браузерного расширения, информацию о котором он получает из бэкграунд-контекста. При возникновении необходимости изменения этого состояния (например, команда создания или изменения шаблона), фронт энд скрипты лишь оповещают об этом бэкграунд скрипт, который уже взаимодействует с хранилищем браузера и десктоп-приложением посредством веб-сокета.

Структура браузерного расширения представлена на рис. 5.

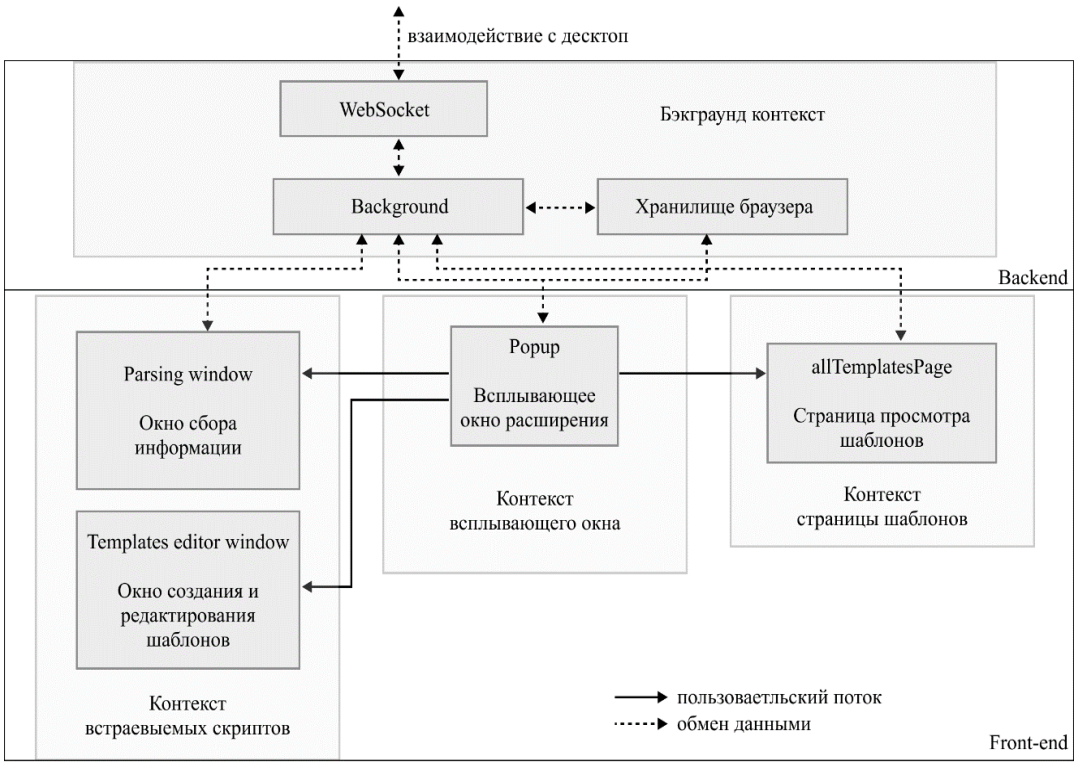

Рисунок 5

Созданные шаблоны используются при сборе информации. Сценарий пользовательского опыта парсинга характеристик объекта изображен на рис. 6.

Доступ к окну сбора информации осуществляется через всплывающее окно расширения или через специальную команду. Как и в случае с окном редактирования шаблонов, на команду реагирует бэкграунд-контекст. Он имеет доступ к доменному имени активной вкладки, что позволяет ему собрать все ассоциированные с ним шаблоны. Собранные шаблоны и команда перенаправляются в контекст контент-скрипта.

Получив запрос на сбор информации, контент-скрипт начнет проверку полученных шаблонов на актуальность. Для этого он попытается собрать все элементы, селекторы которых содержатся в шаблонах. В случае, если элемент не удается найти, шаблон отбраковывается как неактуальный. Таким образом, пользователю будут предоставлены только те шаблоны, которые подходят для текущей страницы. Псевдокод алгоритма проверки шаблонов можно представить следующим образом:

Шаблоны = аргумент функции

КорректныеШаблоны = новый пустой массив

Для каждого Шаблон внутри Шаблоны:
Корректный: = True
Для каждого Поле внутри Шаблон.Поля:
Элемент: = Найти в разметке HTML элемент по Поле.Селектор
Если Элемент == Null:
Корректный: = False
Если Корректный == True
Добавить Шаблон в КорректныеШаблоны
Вернуть КорректныеШаблоны

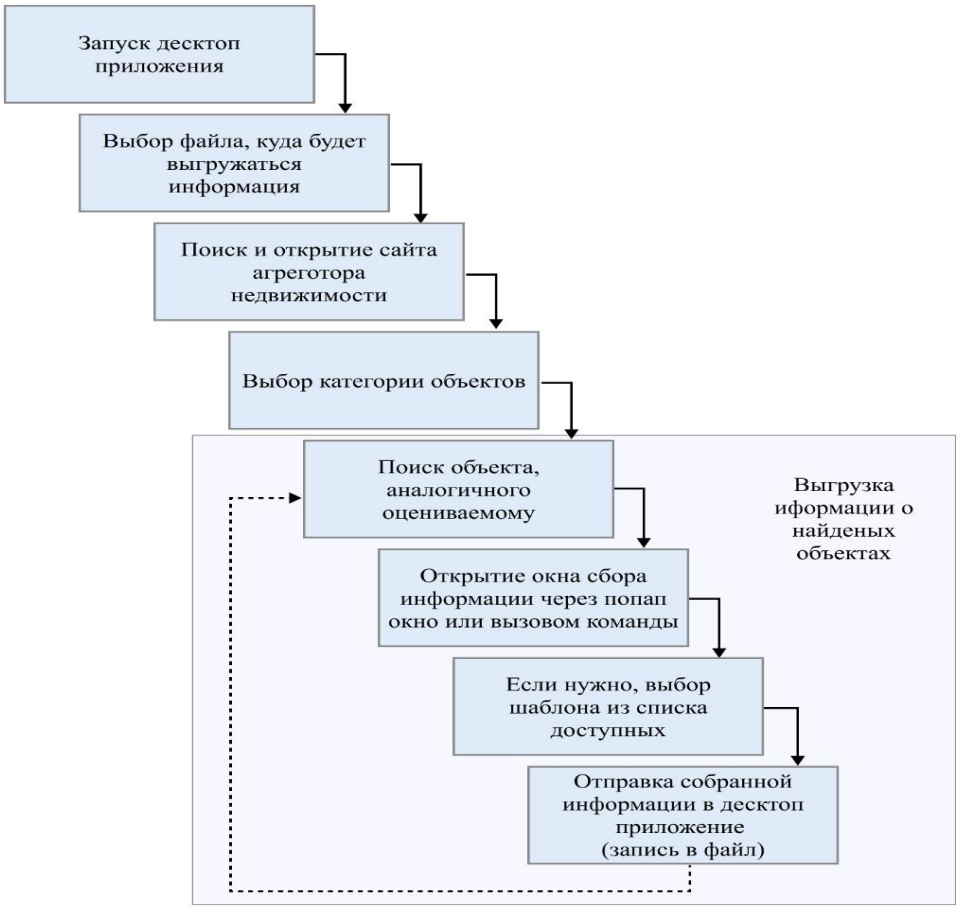

Рисунок 6

После процесса проверки контент-скрипт интегрирует на страницу окно сбора информации. Внутри него присутствуют все актуальные шаблоны. Выбор какого-либо из доступных шаблонов приведет к сбору информации на странице согласно полям шаблона и отображению результатов сбора в интерфейсе. Отображается как сырая (необрезанная по маркерам) информация, так и обработанная (та, которая будет отправлена в десктоп-приложение). Пример отображения собранных данных приведен на рис. 7.

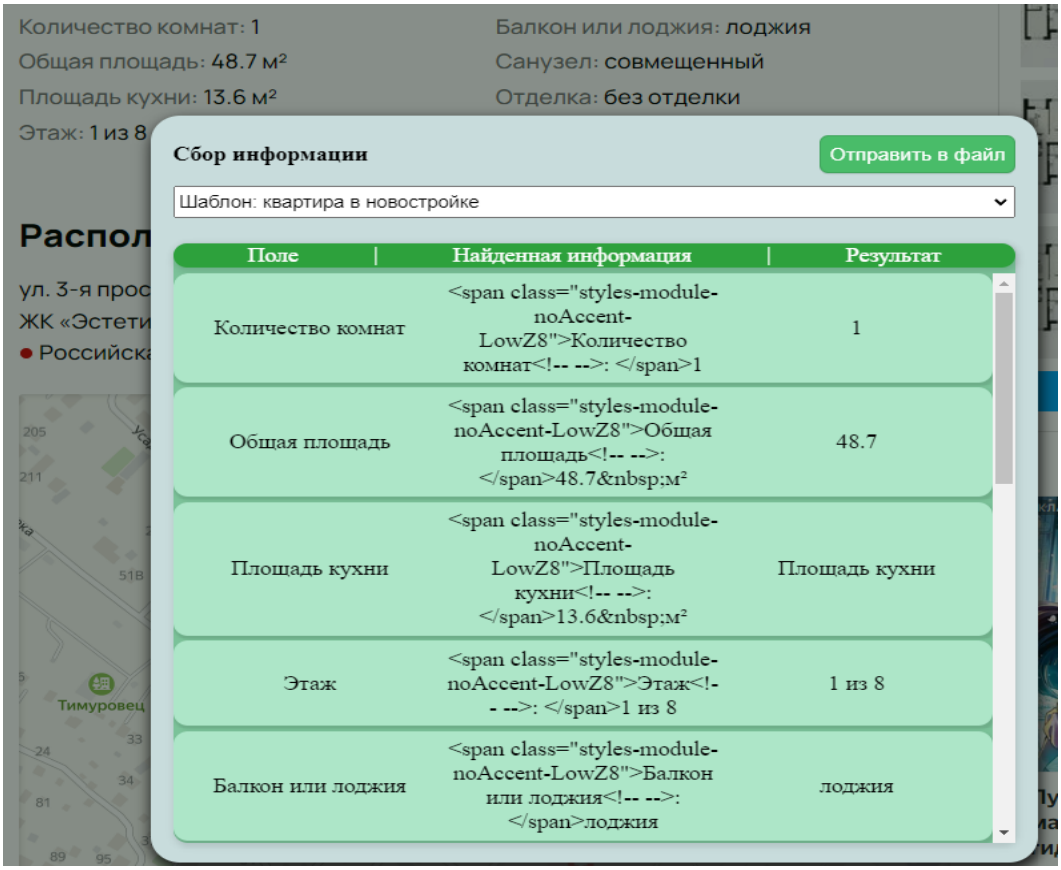

Рисунок 7

Получив запрос на пересылку данных в десктоп приложение, бэкграунд-скрипт попытается установить с ним связь через *WebSocket*. Как только связь будет установлена, он отправит данные по открывшемуся каналу и станет ожидать ответа от приложения. При получении ответа бэкграунд-скрипт закроет соединение и оповестит о результате контент-скрипт активной вкладки. В случае возникновения ошибки сформируется сообщение о провале операции, которое также будет переслано контент-скрипту.

Среди списка ошибок может быть неуспешность установления соединения (когда десктоп-приложение не запущено или не выбран выходной файл, где должна формироваться таблица аналогов), провал записи (когда неверно указано смещение внутри электронной таблицы или номер объекта) или провал открытия файла для записи (когда файл не существует или занят другой программой).

В любом случае в окне сбора информации отобразится результат операции, оповещающий пользователя о успешности процедуры или о возникших ошибках. Полный процесс сбора и отправки данных приведен на рис. 8.

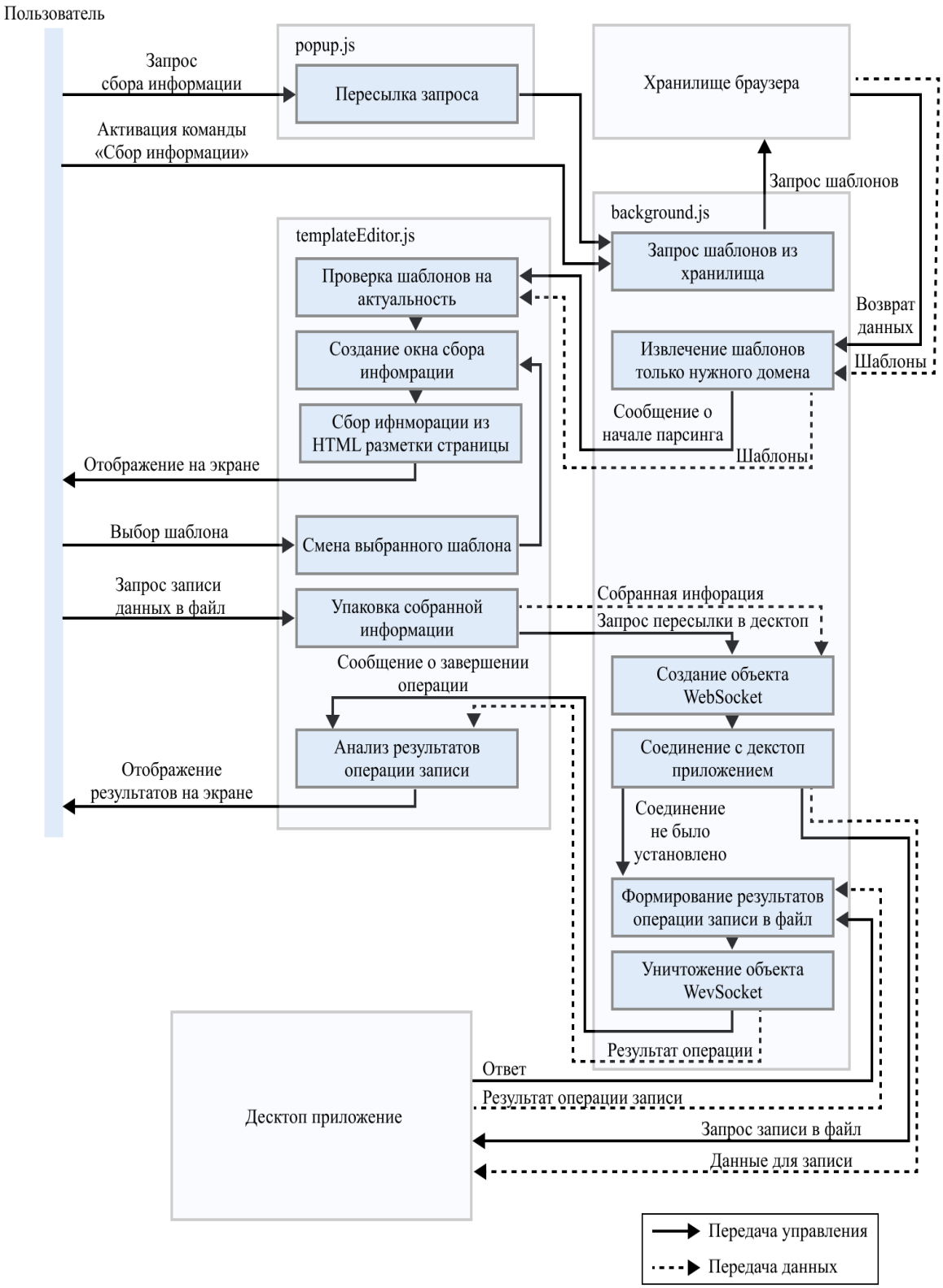
Пользователь
popup.js
Запрос сбора информации
Пересылка запроса
Хранилище браузера
Активация команды «Сбор информации»
Запрос шаблонов
background.js
templateEditor.js
Запрос шаблонов из хранилища
Проверка шаблонов на актуальность
Возврат данных
Создание окна сбора инфомрации
Извлечение шаблонов только нужного домена
Шаблоны
Сбор ифнморации из HTML разметки страницы
Сообщение о начале парсинга
Шаблоны
Отображение на экране
Выбор шаблона
Смена выбранного шаблона
Запрос записи данных в файл
Упаковка собранной информации
Собранная инфорация
Запрос пересылки в десктоп
Создание объекта WebSocket
Сообщение о завершении операции
Соединение с декстоп приложением
Анализ результатов операции записи
Отображение результатов на экране
Соединение не было установлено
Формирование результатов операции записи в файл
Уничтожение объекта WevSocket
Результат операции
Ответ
Результат операции записи
Десктоп приложение
Запрос записи в файл
Данные для записи
Передача управления
Передача данных

Рисунок 8

После сбора информации, используемый шаблон должен быть помечен как последний используемый для данного доменного имени. При открытии окна сбора информации именно он должен выбираться в первую очередь, если это возможно.

Результаты разработки приложения

Программа автоматизации сбора информации реализована в виде двух компонентов: браузерное расширение и десктоп-приложение. Расширение рассчитано на установку в интернет-браузеры, основанные на движке *Chrome* («Яндекс.Браузер» «*Google Chrome*»). Десктоп-приложение скомпилировано в исполняемый файл (*exe*) и рассчитано на использование в рамках персональных компьютеров на базе ОС *Windows* 10 и выше, обладающих шестидесяти четырехразрядной архитектурой процессора.

Пример шаблона, созданного посредством окна редактирования, приведен на рис. 9.

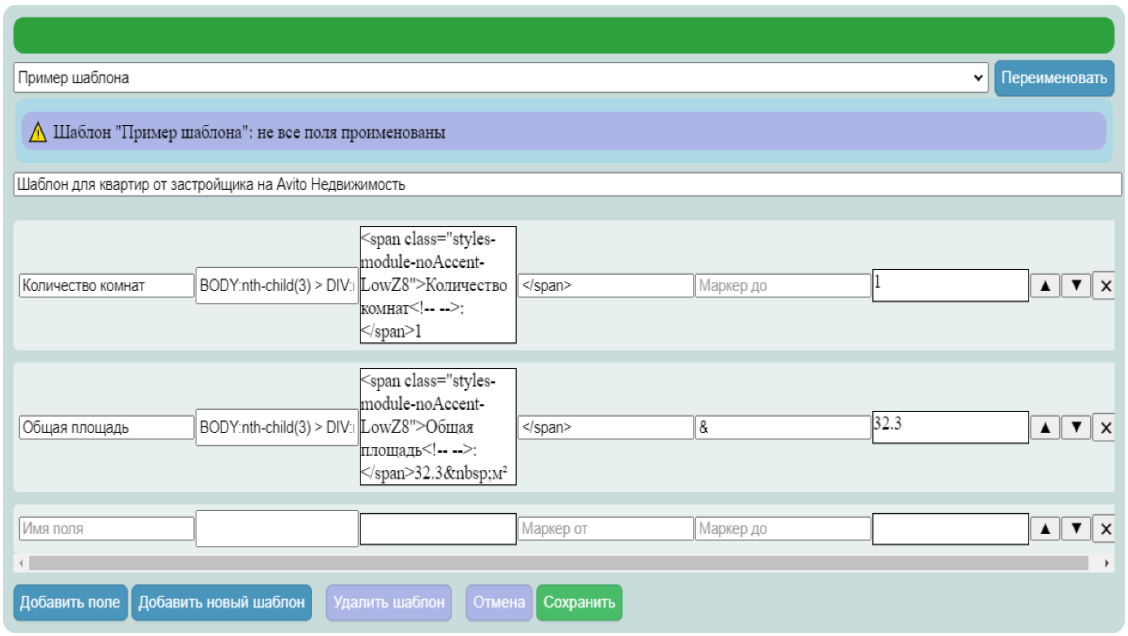

Рисунок 9

На рис. 10 продемонстрирован результат записи собранных данных, произведенных десктоп-приложением, в файл *Microsoft Office Excel*. Данные собраны посредством ранее сформированного шаблона.

	A	B	C	D	E	F	G
1	Номер объекта	Расположение	Общая площадь	Количество комнат	Этаж	Отделка	Цена
2	Объект №1	пос. Мехзавод, 1 кв-л/Московское ш./ул. Николая Баженова, жилые дома	62.8	2	7 из 10	предчистовая	6200000
3	Объект №2	Самарская обл., Самара, посёлок Прибрежный, ул. Труда, 22	60.4	3	2 из 2	кухня, хранение одежды	3150000
4	Объект №3	тер. 18 км Московского шоссе, д. 7А	69.3	2	2 из 17	без отделки	4800000

Рисунок 10

Заключение

Авторами решена актуальная проблема сбора данных с веб-страниц в случаях, когда требуется производить сбор однотипных данных с разных сайтов. Программно-алгоритмическое решение позволяет адаптировать программу к сбору разных наборов данных по созданным шаблонам.

Авторами достигнуты следующие результаты:

1) Проведен анализ деятельности по оценке недвижимости.
2) Определены наиболее трудоемкие и шаблонные процедуры в процессе оценки, подлежащие первичной автоматизации.
3) Проведен анализ технической возможности автоматизации процесса сбора информации со страниц агрегаторов недвижимости.
4) Разработана архитектура приложения, состоящая из двух взаимосвязанных частей.
5) Разработан алгоритм поиска информации на страницах агрегаторов недвижимости.
6) Разработан алгоритм переноса информации в конечную форму документа.
7) Реализован программный комплекс сбора информации с сайтов агрегаторов недвижимости.
8) Определены перспективы развития разработанного приложения.

Программно-алгоритмические решения позволяют адаптировать данную программу для любых предметных областей, связанных со сбором данных с веб-страниц.

Литература

1. Портнов К.В. Анализ цифровой трансформации бизнес-процессов. Актуальные проблемы общества, экономики и права в контексте глобальных вызовов: Сборник материалов X Международной научно-практической конференции, Москва, 17 мая 2022 года. Редколлегия: Л.К. Гуриева [и др.]. – Москва: Общество с ограниченной ответственностью «ИРОК», ИП Овчинников Михаил Артурович (Типография Алеф), 2022. – С. 49-58. – DOI 10.34755/IROK.2022.92.13.091. – EDN EJPTQW.
2. Портнов К.В. Генетические алгоритмы и поиск эффективных порядков индикаторов в биржевой торговой стратегии на основе пересечения трех скользящих средних // Вестник Самарского государственного технического университета. Серия: Технические науки, 2005. – № 32. – С. 72-76. – EDN JWUXKZ.
3. Портнов К.В. Информационные технологии в оценке показателя лояльности клиентов // В мире научных открытий, 2011. – № 3 (15). – С. 254-258. – EDN OCSJNX.
4. Смагина З.А. Технология интернет вещей и ее влияние на современную экономику // Теоретические и прикладные вопросы экономики, управления и образования: Сборник статей II Международной научно-практической конференции. В 2-х томах, Пенза, 15-16 июня 2021 года. Том II. – Пенза: Пензенский государственный аграрный университет, 2021. – С. 182-186. – EDN AJTHBC.
5. Портнов К.В. Анализ задачи оценки лояльности в деятельности компаний в сфере профессиональных услуг // Проблемы развития предприятий: теория и практика, 2020. – № 1-2. – С. 241-244. – EDN HDSWOD.
6. Свидетельство о государственной регистрации программы для ЭВМ № 2023664735 Российская Федерация. Система учета товаров на складе интернет-магазина: № 2023660391: заявл. 24.05.2023: опубл. 06.07.2023 / К.В. Портнов; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Самарский государственный технический университет». – EDN VFHCBC.
7. Латушкина Т.С. Исследование возможностей интернет-продвижения и настройка рекламной компании // Московский экономический журнал, 2023. – Т. 8. – № 5. – DOI 10.55186/2413046X_2023_8_5_280. – EDN RFPBDO.

8. Сахбиева А.И., Калякина И.М., Косников С.Н., Латушкина Т.С., Майорова И.А. Цифровизация экономики и обеспечение безопасности данных // Московский экономический журнал, 2021. – № 8. URL: https://qje.su/ekonomicheskaya-teoriya/moskovskij-ekonomicheskij-zhurnal-8-2021-28.
9. Иноземцев В.Л. На рубеже эпох. Экономические тенденции и их неэкономические следствия [Текст]. – М.: Экономика, 2003. – 730 с.
10. Латушкина Т.С., Харитонова Е.А., Майорова И.А. Анализ подходов к ESG на примере металлообрабатывающего предприятия // Экономика и предпринимательство, 2022. – № 7 (144). – С. 1059-1064.
11. Латушкина Т.С., Майорова И.А. Использование и применение JAVASCRIPT-фреймворков (REACT, ANGULAR, VUE.JS) для разработки WEB-приложений // Экономика и предпринимательство, 2023. – № 9 (158). – С. 1374-1376.
12. Портнов К.В. Актуальные проблемы и задачи автоматизированных систем в сфере ЖКХ // Журнал монетарной экономики и менеджмента, 2024. – № 2. – С. 230-236. – DOI 10.26118/2782-4586.2024.35.72.033. – EDN AEQRFJ.
13. Портнов К.В. Разработка информационной системы на основе многофакторной логистической регрессии // Информационные технологии. Радиоэлектроника. Телекоммуникации, 2012. – № 2-3. – С. 129-133. – EDN PEDEUX.
14. Портнов К.В. Анализ оценки неопределенности инвестиционного портфеля. Математическое моделирование и краевые задачи: Труды Третьей Всероссийской научной конференции, Самара, 29-31 мая 2006 года. Редколлегия: В.П. Радченко (ответственный редактор), Э.Я. Рапопорт, Е.Н. Огородников, М.Н. Саушкин (ответственный секретарь). Том Часть 4. – Самара: Самарский государственный технический университет, 2006. – С. 80-82. – EDN TGOHNF.