УДК 004.056
DOI 10.34822/1999-7604-2019-3-52-60

МЕТОДЫ ИНТЕГРАЦИИ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ УНИВЕРСАЛЬНОГО АНАЛИЗАТОРА ОНЛАЙН-ИНФОРМАЦИИ

К. Ш. Багаутдинов
Российский государственный университет нефти и газа (национальный исследовательский университет) им. И. М. Губкина,
bagautdinov@asugubkin.ru

На сегодняшний день не существует единого формализованного алгоритма решения задачи интеграционного взаимодействия. В представленной работе рассматривается анализ вопроса интеграции, выявлены некоторые закономерности, а также выбран и раскрыт подход к решению указанной задачи. Данное решение позволяет обеспечить автоматизацию процессов выявления мошенничества с залоговым обеспечением и верификацию (оценка) принимаемых в залог объектов недвижимости/ипотечных портфелей (антифрод-система).

Ключевые слова: интеграция, информационные системы, универсальный анализатор данных, антифрод-системы, подходы к интеграции данных.

INTEGRATION METHODS OF INFORMATION SYSTEMS BASED ON UNIVERSAL ANALYZER OF ONLINE INFORMATION

K. Sh. Bagautdinov
National University of Oil and Gas "Gubkin University",
bagautdinov@asugubkin.ru

Today there is no unified algorithm for solving problems of integrated interaction. This article analyses the issue of integration and reveals some patterns. In addition to it, an approach to problem solving is distinguished. The considered solution provides automation of the processes of detection of collateral fraud and verification (assessment) of real estate or mortgage portfolios accepted as collateral (anti-fraud system).

Keywords: integration, information systems, universal data analyzer, anti-fraud systems, data integration approaches.

В современном мире одной из сложных задач в сфере информационных технологий (ИТ) является интеграция систем. Рано или поздно перед компаниями встанет задача автоматизации бизнес-процессов. При этом интеграция информационных систем делится на две части: это интеграция приложений и интеграция данных.

Под интеграцией данных в информационных системах (далее – ИС) следует понимать разработку и поддержку интерфейса для работы с данными из различных независимых неоднородных источников как с единой моделью данных. Источником данных могут быть различные системы баз данных (далее – БД), файлы структурированных данных, веб-сайты и т. п. С увеличением объема и потребностью совместного использования данных роль интеграции увеличивается.

Процесс интеграции данных актуален как в коммерческих задачах (когда разным компаниям нужно объединять БД), так и в научных (для комбинирования результатов исследований из различных источников).

В данной работе рассматриваются методы интеграции информационных систем, архитектуры систем интеграции, механизмы отображения моделей данных, современные подходы к интеграции данных.

Интеграция информационных систем. Под интеграцией информационных систем подразумевают создание единого информационного пространства организации, которое можно осуществить, объединив все внедренные и планируемые к внедрению автоматизированные системы в одну интегрированную систему. Проблема интеграции автоматизированных систем возникает при анализе следующих направлений:

- разработка технологий для возможности выпуска большого количества видов продукции, корректировка производственных заданий;
- необходимость в организации системы функционирования организации на основе гибких методов и подходов;
- устранение барьеров для обмена информацией внутри организации;
- повышение управляемости за счет организационных мер путем обеспечения прозрачности информационных потоков, оперативного управления, принятия согласованных решений;
- развитие универсального объединенного информационного пространства для всех автоматизированных систем организации, под которым подразумевается потенциальная возможность взаимообмена в реальном времени между разными компонентами и модулями АС.

Можно выделить следующие направления интеграции: программная, функциональная, информационная, техническая и организационная.

Программная необходима для обеспечения совместимости функциональных особенностей программных средств, которые используются для решения конкретных детерминированных задач.

Функциональная объединяет в себе базовые элементы информационной и технической инфраструктуры, включая компоненты, для каждого из которых разрабатывается методика расчета критерия эффективности, базовая функционально-аналитическая модель, информационные и функциональные взаимосвязи между модулями систем. В итоге обеспечивается консолидированное пространство локальных направлений функционирования всех модулей.

Информационная смотрит на универсализацию подходов к хранению, сбору, использованию и представлению данных на различных уровнях системы управления. В этом направлении обеспечивается взаимосвязанное движение потоков информации между модулями системы [1].

Техническая необходима для обеспечения единых средств аппаратно-вычислительной техники и локальных сетей, которые в своей совокупности являются базовыми элементами для абсолютно всех направлений интеграции.

Организационная необходима для сопровождения и поддержания деятельности персонала в области управления.

Необходимо постоянно иметь в виду тот неоспоримый факт, что существует большое разнообразие автоматизированных систем. Тогда становится очевидным, что задачи интеграции могут быть самыми разнообразными.

Разносторонние проблемы. Проблема интеграции данных очень многообразна и имеет много важных аспектов. Сложность и характер методов, которые используют для решения данной проблемы, напрямую зависят от требуемого уровня интеграции, а также от свойств множества источников данных и отдельных исходных источников.

Системы интеграции данных способны обеспечивать интеграцию данных как на физическом, так и на логическом и семантическом уровнях. С теоретической точки зрения интеграция данных на физическом уровне сводится к сбору данных из различных хранилищ в единый формат их физического представления. Интеграция данных на логическом уровне организует доступность данных, содержащихся в разных источниках, в терминах объединенной глобальной схемы, описывающей их общее представление, при этом учитывая структурные и поведенческие свойства данных (при этом семантические свойства данных не учитываются). Интеграция данных на семантическом уровне обеспечивает поддержку единого отображения данных, учитывая их семантические свойства в контексте объединенной онтологии предметной области [2].

Источник данных могут обладать различными свойствами, которые, в свою очередь, являются очень существенными для выбора методов интеграции данных:

- поддержка представления данных в терминах той или иной модели данных;
- источники данных могут быть динамическими и статическими;
- источники данных могут быть неоднородными или однородными относительно характеристик, соответствующих требуемому уровню интеграции.

Неоднородность источников данных. Неоднородность источников данных в большинстве случаев отображается в системах интеграции данных с разных сторон. В данном случае описываются неоднородности характеристик источников данных в соответствии с требуемым уровнем интеграции.

Так, например, при интеграции на физическом уровне в исходных источниках данных могут применяться различные форматы файлов. В то время как на логическом уровне интеграции может существовать неоднородность применяемых моделей данных для различных исходных источников или могут различаться схемы данных, несмотря на то, что применяется одна и та же модель данных. Одни источники могут быть объектными базами данных, а другие – веб-сайтами и т. д.

Постановка задачи. В связи с ростом потребностей организации в получении данных от внешних источников появляется необходимость автоматизировать процесс интеграции данных. Для этого нужно разработать архитектуру взаимодействия систем, благодаря которой другие ИС смогут легко и безопасно получать данные из внешних источников.

Часто сторонние сервисы по тем или иным причинам не хотят открывать свои API, в таких случаях приходится вытаскивать информацию самим, отсюда возникают задачи, которые решает универсальная система сбора данных:

1. Извлечение частично структурированной текстовой информации с сайтов.
2. Большинство сайтов с данными устроены таким образом, что для получения доступа к информации необходимо эмулировать поведение пользователя, потому что некоторые сайты используют специальные фильтры, чтобы выдавать роботам другую информацию, т. е. анализатор должен выгружать информацию, оставаясь незаметным для сайта.
3. Приведение формата данных, специфичных для сайтов, в промежуточный, схожий (но не вполне идентичный). Кроме того, необходимо отслеживать изменение формата данных и уведомлять об этом.
4. Отслеживание явных дубликатов, обновление измененной информации.
5. Разбор текстовой информации с сайтов и преобразование в полностью структурированный формат.

Требуемая архитектура необходима для системы, обеспечивающей автоматизацию процессов выявления мошенничества с залоговым обеспечением и верификацию (оценка) принимаемых в залог объектов недвижимости/ипотечных портфелей (антифрод-система).

Для выявления признаков мошенничества (внешнего и внутреннего) система производит анализ как по ранее собранным данным, так и по текущим рыночным данным о залогах.

Система встроена в онлайн режиме в технологический процесс выдачи ипотечных кредитов как инструмент автоматизированного принятия решения и АРМ-верификатора залогов.

Для оценки залогов в системе используется сравнительный (рыночный) подход, основанный на сравнении объекта оценки с аналогичными объектами недвижимости, в отношении которых имеется информация о ценах сделок с ними или ценах предложении о продаже.

По характеристикам объекта подбираются аналоги на рынке недвижимости и по их ценам рассчитывается стоимость объекта.

Система используется как для оценки индивидуальных объектов, так и для оценки и анализа портфеля объектов.

Сбор информации о предложениях по продаже жилой недвижимости осуществляет программная часть необходимой системы. Данная программа должна выполнять следующие задачи:

- «обход» интернет-сайтов и загрузка HTML-страниц, открытых для публичного использования. Используемое количество сайтов – 120. Объем ежедневно загружаемых объявлений составляет порядка 25 гигабайт;
- перевод прочитанных объявлений, имеющих специфичный для каждого сайта вид, в некоторое промежуточное унифицированное представление (набор и содержание полей);
- отслеживание изменений, происходящих в формате описания документов на сайте. (Предполагается, что структура страниц сайта меняется в среднем 1–2 раза в год, тогда при скачивании информации со 120 сайтов следует ожидать, что периодичность доработки сайтов может быть ежедневной);
- отслеживание явных дубликатов объявлений, а также учет обновлений информации в объявлении;
- подсчет срока экспозиции объявления на сайте;
- сохранение архива объявлений (HTML + Изображение страницы в графическом формате) в первичной БД с возможностью поиска по ID, URL;
- в системе предусматривается ряд алгоритмов, которые могут быть встроены в различные ее части для решения внутренних задач (например, разбор адреса, выявление дубликатов), которые могут простираться по многим подсистемам (расчет срока экспонирования объявления).

Полученные неструктурированные данные из внешних источников должны храниться в упорядоченном формате.

Подход к решению поставленной задачи. При развитии организаций происходит смена внутренней архитектуры, бизнес-процессов, пользовательских интерфейсов, ускорение процессов, и в такой ситуации задача интеграции превращается в серьезную проблему. Продолжая развиваться, компании становятся крупными, задачи становятся комплексными, и появляется распределенность, которая требует поддержки. Кроме того, во многих организациях, в зависимости от бюджета и поставленных задач, используются различные платформы и инструменты от разных производителей (гетерогенность), которые приходится поддерживать в дальнейшем. Также не стоит забывать про морально устаревшие технологии, системы либо неформализованные структурные данные, которые тоже приходится использовать. Задачи интеграции не ограничиваются пределами организации, все чаще необходимо получать данные от партнеров, клиентов, подрядчиков, государственных структур и других внешних источников (межсистемная интеграция) и др.

Параметры, отвечающие за сложность интеграции:

1. Концептуальная разница – основа данной разницы заключается в том, что разработчики разных ИС приняли концептуально разные допущения, поэтому данные системы не могут корректно состыковаться [3]. Решается такая проблема при помощи введения еще одного слоя абстракции, который не будет противоречить концептуально обеим системам. При этом можно выделить два варианта реализации:
   - получившаяся система является централизованной, а все остальные интегрируемые системы превращаются в подсистемы;
   - применение архитектуры брокера, посредника, который не является центром, а только обеспечивает прослойку между системами. Все интегрируемые системы остаются независимыми.
2. Технологическая разница – основывается на несовместимости форматов обмена данными, протоколов взаимодействия и интерфейсов. Решается использованием брокеров, прослоек, конвертов и других не вполне красивых, но достаточно надежных средств.

Общая задача выглядит так: необходимо интегрировать N информационных систем, учитывая данные факторы, с минимизацией количества прослоек. При первом варианте решения задачи получается, что между N системами будет $\frac{N(N-1)}{2}$ связей при двустороннем взаимодействии $N \times (N-1)$ интерфейсов.

Существует много решений данной задачи, но были выделены следующие классические методы:

- интеграция на уровне данных – смысл данного метода заключается в том, что некоторый массив приложений посылает запрос в одну единую БД или в несколько разных БД, которые связаны репликациями [4];

- интеграция на уровне брокеров – суть данного метода заключается в разработке и реализации дополнительного (не основного) программного модуля, обращающегося во множество систем;

- стандартизация – типичный классический метод, который требует применять различные стандарты (как международные, так и государственные, а также отраслевые) при разработке систем для обратной совместимости;

- интеграция на уровне пользователей – данный метод является нежелательным, можно сказать, не является автоматизацией. Смысл представленного метода заключается в том, что пользователи обмениваются данными между системами несистемным путем, а другими доступными способами. Например, через почту, копирование данных, перенос файлов и т. д. Данный метод используется в крайнем случае, когда во время обновления, модернизации невозможно заморозить бизнес-процесс;

- интеграция на уровне сервисов – данный метод содержит в себе очень изящную концепцию интеграции, при которой фиксируются интерфейсы и форматы данных со всех сторон. При использовании данного метода интеграции получается очень легко отработать и настроить межсистемную логику [5].

Сравнение методов интеграции представлено в табл. 1.

Таблица 1

Сравнение методов интеграции

Метод	Преимущества	Недостатки
Стандартизация	Совместимость	Долгий ввод новых стандартов при необходимости
Интеграция на уровне брокеров	Универсальность	Сложность, трудоемкость, высокая стоимость разработки
Интеграция на уровне данных	Низкая стоимость интеграции	Разные приложения могут приводить данные в противоречивые состояния; дублирование кода
Интеграция на уровне сервисов	Быстрая отработка межкорпоративной бизнес-логики	Присутствует фиксация
Интеграция на уровне пользователя	Используется в крайнем случае	Не автоматизированная интеграция

Выбор подхода к интеграции и средств разработки. Интеграция данных будет производиться не только на физическом, но и логическом и семантическом уровнях. Для данной разработки важны сами данные, а также их логическая связь в источниках с учетом природы этой информации [6].

Поэтому, исходя из анализа решаемых задач и требований к ним, было решено автоматизировать процесс интеграции неструктурированных данных путем разработки ETL системы, которая позволит реализовать гибкое, легко поддерживаемое, расширяемое решение, основанное на open source технологиях.

У представленной системы будет своя база данных для консолидации информации из целевых источников, отслеживания изменений и передачи данных другим ИС и приложениям.

Архитектура разработанной системы. После выбора подхода к интеграции и средств разработки, была предложена архитектура системы сбора данных. Всю архитектуру (рис. 1) можно разделить на две части.

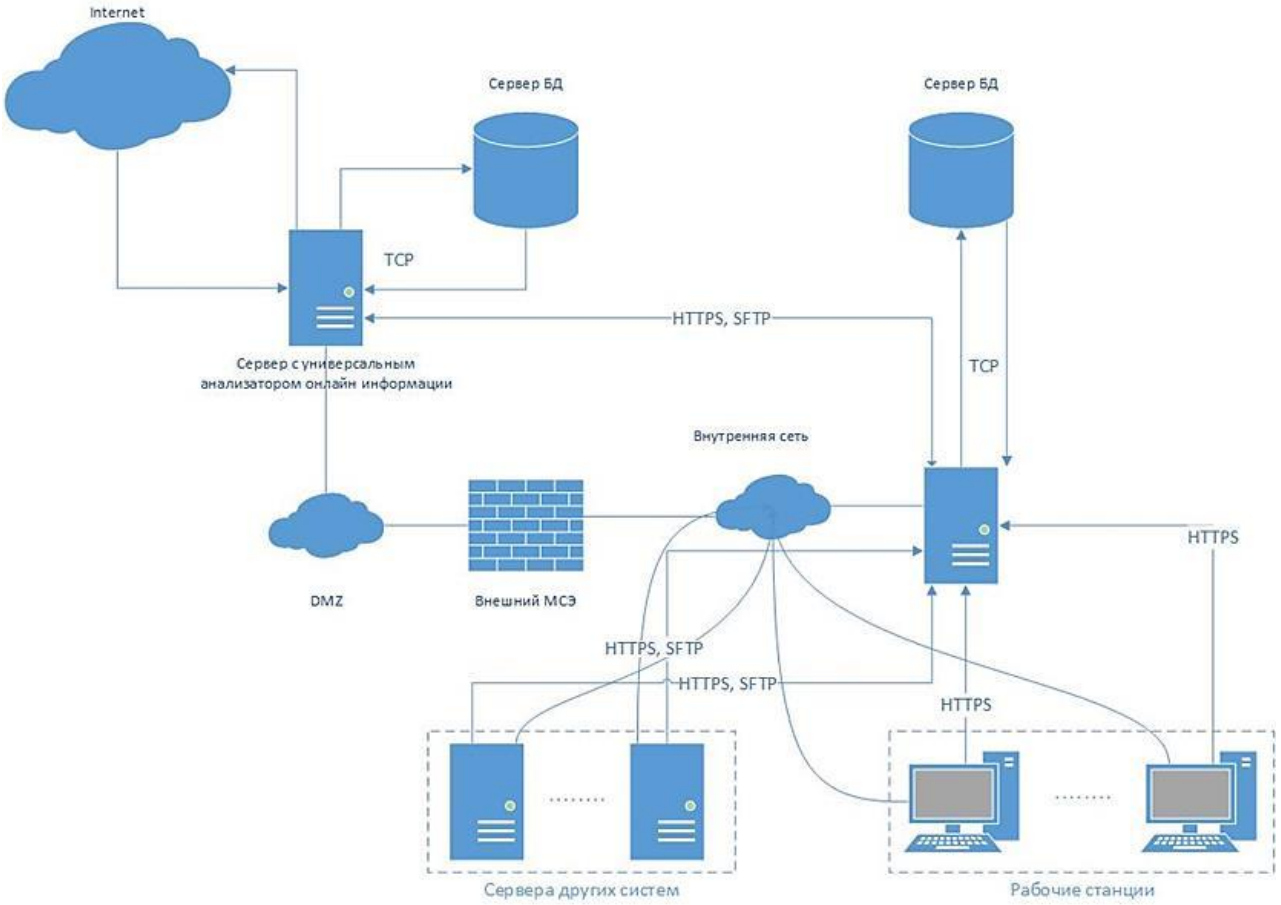

Рис. 1. Архитектура системы сбора данных

В первой части имеются сервер приложений, который взаимодействует с внешними источниками, и сервер БД для хранения полученных данных. Вся часть находится в DMZ (демилитаризированной зоне).

В компьютерной безопасности DMZ, или демилитаризованная зона (иногда называемая периметрической сетью), представляет собой физическую или логическую подсеть, которая содержит и предоставляет внешние службы организации для ненадежной сети. Обычно это более крупная сеть, такая как Интернет. Целью DMZ является добавление дополнительного уровня безопасности в локальную сеть организации (ЛВС): внешний сетевой узел может получить доступ только к тому, что отображается в DMZ, а остальная сеть организации – межсетевой экран. DMZ функционирует как небольшая изолированная сеть, расположенная между Интернетом и частной сетью, и, если ее дизайн эффективен, позволяет организации уделять дополнительное время обнаружению и устранению нарушений до того, как они будут проникать во внутренние сети.

Название происходит от термина «демилитаризованная зона» – область между национальными государствами, в которой военная операция не разрешена.

Сервер с универсальным анализатором передает данные через протокол TCP к серверу БД и имеет выход в сеть Интернет через протоколы https и http.

Во второй части имеется сервер приложений, который взаимодействует с сервером универсального анализатора в первой части архитектуры через протоколы https и sftp.

Сервер приложений во второй части позволяет другим системам через API получать данные, полученные из внешних источников.

С рабочих станций через сервер приложений во второй части имеется доступ к серверу универсального анализатора через интерфейс для задания внешних источников, которые будут подлежать анализу.

Описание программного модуля. Для разработки анализатора был выбран язык PERL, потому что основной особенностью языка считаются его богатые возможности для работы с текстом, в том числе работа с регулярными выражениями, встроенная в синтаксис.

При разработке приложения использовались модули:

- модуль CGI, который предназначен для упрощения создания HTML-документов;
- модуль DBI, необходимый для работы с БД;
- библиотека модулей LWP, которая позволяет в полной мере использовать возможности http(s)-протоколов.

Приложение написано так, что оператор вносит некоторые правила, условия, настройки и шаблоны (которые хранятся в БД приложения) через интерфейс. Затем, в зависимости от настроек, начинается анализ онлайн сервисов, выборка информации и скачивание в БД.

Универсальный анализатор решает все поставленные задачи:

- извлечение частично структурированной текстовой информации с сайтов;
- во время получения информации анализатор способен оценить кодировку получаемых данных и привести к кодировке целевой таблицы для корректного отображения данных.

Данный пункт очень важен для текстовой информации на русском языке.

Кроме того, если кодировка была подобрана неверно либо неверно отображена в источнике, то пользователь сам вручную сможет подобрать кодировку, выбрав читаемый текст, и преобразовать данные для дальнейшей работы с ними;

- разработанный программный модуль способен имитировать браузер пользователя и действия живого человека.

Владельцы некоторых сайтов целенаправленно определяют ботов для последующей подмены контента.

Для решения предоставленной задачи модуль может использовать:

- различные случайные задержки при обращении к сайту. Данный способ помогает предотвратить хаотичные интенсивные запросы, которые приводят к блокировке пользователя сайта, отправляющего эти запросы;
- данные для авторизации на веб-сервисе. Используются, если данные доступны конкретному пользователю. При защищенном соединении происходит обязательная проверка ssl-сертификатов;
- распознавание капчи. Часто на сайтах просят пройти компьютерный тест, используемый для определения, кем является пользователь системы: человеком или компьютером. В универсальном анализаторе за дополнительную плату используется сторонний сервис распознавания капч. Очень полезная функция, когда владельцы информации не хотят отдавать данные роботам;
- использование заданных прокси-серверов. Данный функционал необходим для анонимизации соединения;
- универсальный анализатор способен корректно обработать код состояния HTTP и определить, какие действия ему предпринимать дальше. В приоритете стоят попытки разрешения проблемы без участия пользователя;

Кроме того, робот уведомляет о неизвестных ошибках, возникших в ходе работы, и имеет реестр решений типовых ошибок. Если оператор может обойти эту ошибку, то и робот сможет сделать это по введенным данным от оператора;

- форматно-логический контроль – для снижения числа потенциальных ошибок с получением данных пользователь может задавать определенные шаблоны для полей. Например, ИНН, СНИЛС, Адрес, Паспорт РФ и т. д.;
- проверка адекватности полученных данных, уведомление пользователя о проблемах – при получении явно короткого сообщения, при неуспешной попытке разбиения страницы на структурированный формат пользователь будет уведомлен и будут предложены варианты решения проблемы;

- универсальный анализатор не требует специально обученного человека для его использования. Любой пользователь ПК спокойно справится со всеми настройками и сможет создать задание на выгрузку данных. Данный робот имеет множество настроек, позволяющих пользователю, не заходя в исходный код, использовать его для различных источников информации с разными структурами данных;

- представленный модуль способен анализировать не только html-страницы, а также популярные форматы обмена данными, такие как json, xml и др.;

- у данного анализатора реализован открытый API с основными методами для того, чтобы его можно было использовать без стандартного интерфейса, а также для того, чтобы другие разработчики могли написать свой интерфейс, учитывая особенности их систем;

- помимо прямого получения структурированных данных из БД, их можно получать как дамп базы, таблицы в excel в письме или скачивать напрямую.

На рис. 2 отображена схема извлечения из внешних источников, преобразования и загрузки данных в хранилище антифрод-системой.

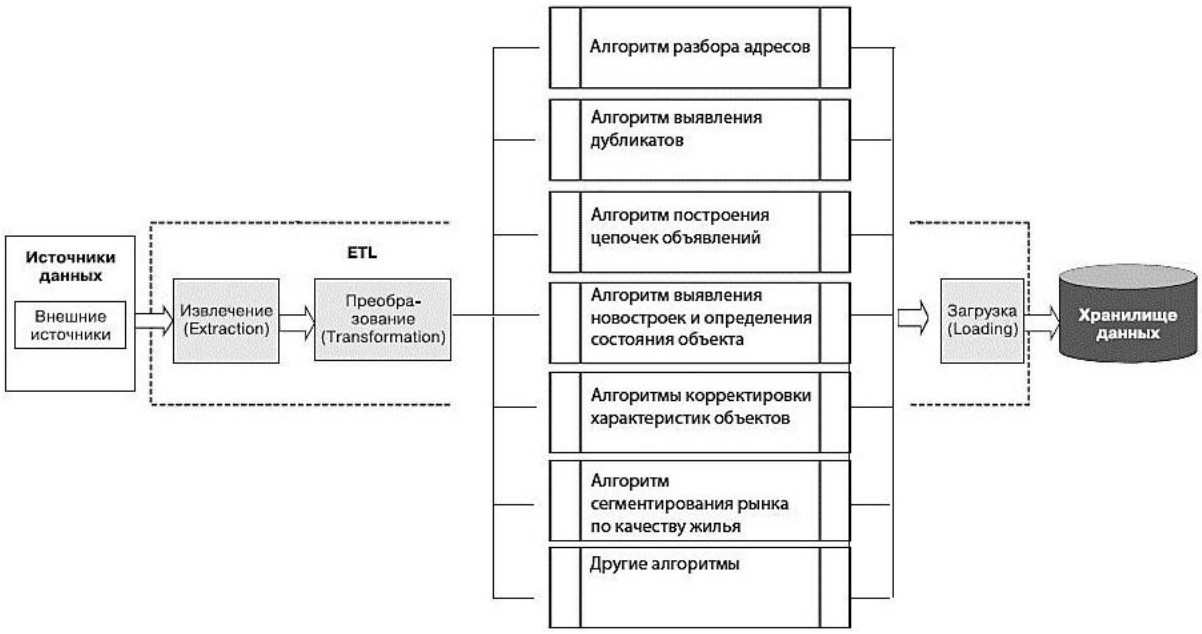

Рис. 2. Схема извлечения данных из внешних источников

Результаты внедрения системы. Представленная разработка позволила повысить эффективность определения мошенников как со стороны клиентов, так и со стороны сотрудников, оформляющих кредит, а также позволила в онлайн-режиме производить оценку принимаемых в залог объектов недвижимости/ипотечных портфелей и ускорила механизм принятия решения о выдаче ипотеки.

До внедрения системы с универсальным анализатором требовалось два сотрудника (оператор и верификатор) на полный рабочий день (работа 7 часов 5 дней в неделю). Все заявки обрабатывались вручную, анализировались сайты недвижимости (около 100 шт.) и подготовленная организацией БД. Производительность данных сотрудников ограничивалась обработкой 200 заявок в день (~1 заявка в 3 мин). В ходе работы возникало 5 % ошибок в связи с ошибками ввода.

После ввода представленной системы производительность повысилась. Теперь требуется всего лишь один сотрудник (администратор) для поддержки (неполный рабочий день). Обработка заявок происходящая в режиме реального времени благодаря максимально свежей БД. Процент ошибки снизился до 0,005 %. Производится автоматический контроль дубликатов и ведение истории.

На примере антифрод-системы, разработанной на основе универсального анализатора, можно увидеть, как универсальный анализатор позволяет автоматизировать процессы, связанные с интеграцией данных.

Заключение. В представленной статье рассмотрены разносторонние проблемы, возникающие при интеграции информационных систем. Также раскрыты подходы к решению поставленной задачи, которые включают в себя планирование архитектуры взаимодействия систем и разработку универсального анализатора данных.

Литература

1. Schmarzo B. Big Data: Understanding How Data Powers Big Business / Bill Schmarzo. Hoboken, NJ : Wiley, 2013. 240 p.

2. Леонов Д. Г. Методы, модели и технологии разработки и интеграции распределенных гетерогенных программно-вычислительных комплексов в транспорте газа. М. : ИЦ РГУ нефти и газа (НИУ) им. И. М. Губкина, 2017. 196 с.

3. Byrne C. Integration and the Path to Becoming a Digital Business. O'Reilly Media, Inc, 2018. 26 p.

4. Костогрызов А. И., Нистратов Г. А. Стандартизация, математическое моделирование, рациональное управление и сертификация в области системной и программной инженерии. 2-е изд. М. : Вооружение, политика, конверсия, 2005. 395 с.

5. Когаловский М. Р. Перспективные технологии информационных систем. М. : ДМК Пресс, 2003. 288 с.

6. Шмаль Г. И., Григорьев Л. И., Кершенбаум В. Я., Леонов Д. Г. Цифровая экономика нефтяного производства. М. : Нефт. хоз-во, 2019. С. 100–104.