

УДК 004.6

СПОСОБЫ И МЕТОДЫ СОЗДАНИЯ ИНТЕГРАЦИЙ ПРИЛОЖЕНИЙ И СИСТЕМ

Вершинин Евгений Владимирович

Кандидат физико-математических наук, доцент и заведующий кафедрой «Системы обработки информации» Калужского филиала федерального государственного бюджетного образовательного учреждения высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», 248000, Россия, г. Калуга, ул. Баженова, д. 2.
vershinin@bmstu.ru

Шилкина Мария Сергеевна

Студент группы ИУК5-11М Калужского филиала федерального государственного бюджетного образовательного учреждения высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», 248000, Россия, г. Калуга, ул. Баженова, д. 2.
shilkinams@student.bmstu.ru

Аннотация

На данный момент времени большинство компаний и предприятий используют разные ИТ-модули и системы для решения разных задач. Но по мере увеличения и усложнения бизнес-процессов, компании сталкиваются с перегруженностью и разрозненностью инструментов и компонентов их системы, которые не могут эффективно и быстро обмениваться данными, и работать как единое целое. В таком случае можно использовать различные интеграции, которые позволяют объединить между собой различные модули, системы, приложения.

В данной статье рассматриваются существующие виды интеграций приложений и систем, методы и способы создания интеграций для объединения таких систем и приложений в единую экосистему, преимущества использования интеграций, а также примеры использования интеграций в зависимости от поставленных целей и задач.

Ключевые слова: интеграция, системная интеграция, интеграция систем, виды интеграций, методы интеграций, способы создания интеграций, взаимодействие систем, передача данных, экосистемы, способы взаимодействия, преимуществ интеграций, примеры интеграций, API, EAI, Common Data Storage, Middleware Data Integration, EDI, веб-хуки.

WAYS AND METHODS FOR CREATING APPLICATION AND SYSTEM INTEGRATIONS

Evgeny V. Vershinin

Ph.D, Associate Professor and Head of the department "Information Processing Systems" of Bauman Moscow State Technical University (Kaluga Branch), 248000, Russian Federation, Kaluga, Bazhenova st., 2.
vershinin@bmstu.ru

Maria S. Shilkina

Student of group IUK5-11M of Bauman Moscow State Technical University (Kaluga Branch), 248000, Russian Federation, Kaluga, Bazhenova st., 2.
shilkinams@student.bmstu.ru

ABSTRACT

At this point in time, most companies and enterprises use different IT modules and systems to solve different problems. But as business processes grow and become more complex, companies are faced with congestion and fragmentation of tools and components of their system that cannot effectively and quickly exchange data and work as a whole. In this case, you can use various integrations that allow you to combine various modules, systems, applications.

This article discusses the existing types of integrations of applications and systems, methods and methods for creating integrations to combine such systems and applications into a single ecosystem, the benefits of using integrations, as well as examples of using integrations depending on the goals and objectives.

Keywords: integration, system integration, systems integration, types of integrations, integration methods, ways of creating integrations, interaction of systems, data transfer, ecosystems, ways of interaction, advantages of integrations, examples of integrations, API, EAI, Common Data Storage, Middleware Data Integration, EDI, webhooks.

Введение. В современном мире различные компании и предприятия не стоят на месте и постоянно развиваются, чтобы успеть за быстро растущим рынком и своевременно удовлетворить потребности клиентов, покупателей и пользователей. Соответственно при расширении и росте бизнеса, у него появляется все больше и больше новых бизнес-задач и бизнес-процессов. При создании приложений и систем для решения поставленных задач разработчики редко хотят начинать с нуля, и в то же время у бизнеса может не быть ресурсов (как времени, так и денег) на создание и разработку с нуля. В таком случае лучше всего использовать уже что-то существующее или связать отдельные компоненты и модули воедино, создав единую экосистему, которая позволит бизнесу работать более гибко, быстро и эффективно. Тема создания различных интеграций на данный момент является актуальной, так как рынок IT- технологий не стоит на месте, и с каждым годом появляются все новые и более эффективные способы и методы создания различных интеграций. В данной статье рассматриваются различные виды интеграций приложений и систем, методы и способы их создания, примеры использования той или иной интеграции.

Что такое интеграция?

Интеграция – это процесс объединения программных и/или аппаратных модулей в единую инфраструктуру, позволяющую всем частям системы работать как единое целое [1]. Простыми словами с помощью интеграций мы можем соединить несколько частей системы в единое целое, при этом частью приложения может быть как отдельное приложение, отдельная система или сервис.

Преимущества системной интеграции

Автоматизация и оптимизация

Благодаря использованию и внедрению системной интеграции все релевантные и коррелированные данные объединяются в единую систему. Это значит, что процесс обработки и выборки данных станет проще и прозрачнее для всех подсистем. Если правильно спроектировать и настроить интеграцию, то можно получить интегрированную систему, которая поможет автоматизировать и оптимизировать бизнес-процессы, благодаря идеально налаженным каналам передачи данных [2].

Доступность, точность и согласованность

Процесс системной интеграции включает в себя сбор данных из нескольких источников и их хранение в одном месте. Это означает, что при использовании системной интеграции нет необходимости синхронизировать данные вручную в нескольких системах. Вместо этого, если одна подсистема вносит какие-либо изменения в данные, они автоматически обновляются и для других систем. Таким образом, одним из основных преимуществ системной интеграции является доступность данных в режиме реального времени. Автоматическое обновление и синхронизация данных также повышают точность, поскольку вероятность доступа пользователей к устаревшим данным сводится к минимуму или полностью исключается. Благодаря доступу к последним данным для всех подсистем все пользователи могут работать вместе, что позволяет им взаимодействовать согласованно. Кроме того, оперативный доступ к данным приводит к более быстрому принятию решений, улучшая общий бизнес-процесс [2].

Эффективность и производительность

Отсутствие и разрозненность в данных между различными частями системы (например, между разными командами компании) может привести к снижению качества работы [3]. Если одна подсистема вносит изменения в свою часть данных, другим подсистемам, возможно, придется «синхронизироваться», то есть физически (вручную) вводить обновленные данные в свои соответствующие хранилища данных.

Интегрированная система помогает достичь более высокой эффективности за счет исключения повторяющегося ручного ввода данных. Поскольку система автоматически обновляет центральную базу данных, подсистемы имеют возможность получать точные данные для их дальнейшей выборки и обработки, тем самым позволяя экономить ресурсы и время.

Масштабируемость

На данный момент времени интегрированные системы в значительной степени зависят от облачного хранилища. Наличие отдельного хранилища или вычислительной платформы для каждой подсистемы требует непомерного количества ресурсов [3]. По мере увеличения объема данных необходимо увеличивать пропускную способность для каждой подсистемы в отдельности. Когда мы имеем интегрированную систему, эта репликация удаляется. Все подсистемы могут использовать общие ресурсы по мере необходимости. Если

объем вычислений или хранения данных в систему растет, мы можем просто увеличить количество облачного ресурса. Таким образом, масштабируемость является одним из ключевых преимуществ интеграции.

Виды интеграций

В общем случае интеграция между частями системы может быть внутренней и внешней (со сторонними сервисами). Под внешней подразумевается интеграция двух отдельных сторонних систем, приложений или сервисов, которые могут разрабатывать разные компании [4]. Под внутренней подразумевается интеграция подсистем в рамках приложения, компании или IT-продукта. Например, если мы хотим интегрировать наш сервис с системой электронной оплаты, или добавить в нашу систему какую-либо функцию, которая уже разработана другой компанией, то нам необходимо настроить внешнюю интеграцию. А если мы хотим настроить взаимодействие между подразделениями компании, то нам необходимо настроить внутреннюю интеграцию.

Внешней интеграцией может быть интеграция «business-to-business» или B2B, которая автоматизирует транзакции и обмен документами между компаниями [5]. Это приводит к более эффективному взаимодействию бизнеса с поставщиками, клиентами и партнерами.

Например, подключение системы закупок розничной торговли к ERP-системе поставщика.

Также внешней интеграцией может быть интеграция со сторонними системами, которая позволяет расширить функциональные возможности уже существующей системы. Такую интеграцию стоит применять, когда бизнесу необходимы новые функции, но средств и времени на полноценную разработку с нуля нет. Примером является интеграция существующего приложения с системами онлайн-платежей (СберPay), социальными сетями (Вконтакте), сервисами потокового онлайн-видео (Rutube) и т. д.

Внутренней интеграцией может быть интеграция корпоративных приложений (EAI - Enterprise Application Integration). Целью данной интеграции является объединение различных подсистем внутри одной бизнес-среды [5]. По мере роста компании внедряют все больше и больше корпоративных приложений для оптимизации своих бизнес-процессов. Эти приложения часто не взаимодействуют между собой и тем самым накапливают огромные объемы данных по отдельности. Интеграция корпоративных приложений (EAI) объединяет все функции в одну бизнес-цепочку и автоматизирует обмен данными в режиме реального времени между различными приложениями.

Например, создание единой экосистемы для бухгалтерского учета, информации о человеческих ресурсах, планирования ресурсов предприятия (ERP) и CRM-систем.

EAI также упрощает глобальную архитектуру экосистемы, обеспечивая взаимодействие многих приложений и работая по различным техническим протоколам. EAI гарантирует высокий уровень безопасности, позволяет создавать общепринятые модели данных и расширяет возможности мониторинга работы систем.

Еще одним из подтипов внутренней интеграции является интеграция с устаревшей системой, то есть интеграция современных приложений в существующие устаревшие системы [6]. Большинство компаний используют устаревшее программное обеспечение для выполнения своих основных бизнес-функций. Его нельзя удалить и заменить более

современными технологиями, поскольку оно имеет решающее значение для каждодневного рабочего процесса компании. Вместо этого устаревшие системы можно модернизировать, установив канал связи с более новыми информационными системами и технологическими решениями. Как пример, подключение устаревшей CRM-системы к хранилищу данных.

Одновременно и внутренней и внешней интеграцией является интеграция обмена данными, которая позволяет проводить полный и прозрачный обмен соответствующей информацией между различными модулями системы, программными обеспечениями, системами и приложениями. Такую интеграцию можно осуществить с помощью единого доступа к данным, общего хранилища данных (Common Data Storage) или общего пользовательского интерфейса (Common User Interface).

Способы и методы создания интеграций

Важной частью настройки интеграции является выбор правильного и эффективного метода создания интеграции. Существует несколько способов и методов создания интеграций. Далее рассмотрим некоторые из них.

Единый доступ к данным

Единый доступ к данным позволяет вам получать доступ к данным из различных источников и представлять их согласованно. Этот метод интеграции фокусируется на отображении данных в согласованном формате для простоты использования, при этом фактически сохраняя исходные данные в их первоначальном назначении [7]. Репликация данных происходит по мере их просмотра, но данные всегда остаются в исходном источнике. Используя такую интеграцию доступа, объектно-ориентированные системы управления могут обеспечить единообразие между различными типами баз данных.

Общее хранилище данных (Common Data Storage)

По мере того, как бизнес-данные становятся все более сложными и объемными, общее хранилище данных является подходом к интеграции, который используют многие предприятия и компании. В принципе, как и при едином доступе к данным, информация преобразовывается перед копированием в хранилище данных, поэтому вашим системам нужно обращаться только к одному источнику данных, а не к сотням [7]. Поскольку данные находятся в одном месте и доступны в любое время, вы можете запускать инструменты аналитики и бизнес-аналитики (BI) когда вам это необходимо. Интегрируя приложения с помощью Common Data Storage, вы получаете определенный тип хранилища, который дает вам свободу управления и защиты данных, используемых несколькими различными программами и приложениями.

Интерфейсы прикладного программирования (API)

API — это набор инструментов программного обеспечения, определений и протоколов, которые помогают приложениям взаимодействовать с другими приложениями [8]. Интерфейсы прикладного программирования (API) обеспечивают наиболее распространенный и простой способ подключения двух систем. Они обеспечивают передачу данных и функций в стандартизированном формате между различными приложениями, веб-службами, системами и модулями. API определяет и автоматизирует, как два или более программных компонента будут обмениваться данными в режиме реального времени, какие запросы могут быть сделаны и как они могут быть выполнены.

Интеграция API позволяет компаниям отправлять автоматические текстовые сообщения, проверять учетные данные для входа, переводить контент и собирать данные о пользователях.

Промежуточное ПО (Middleware Data Integration)

Middleware Data Integration - скрытый программный уровень, который объединяет распределенные системы, приложения, службы и устройства. Он выполняет различные задачи, такие как управление данными, обмен сообщениями, управление API или аутентификация.

Другими словами, это сторонний инструмент, который позволяет соединять два или более приложений. Эти приложения могут быть облачными (например, приложение SaaS) или работать локально (например, устаревшая база данных).

Программное обеспечение промежуточного слоя интеграции может принимать форму инструмента сервисной шины предприятия (ESB), который позволяет приложениям подключаться и взаимодействовать друг с другом через централизованную коммуникационную шину; или интеграционная платформа как услуга (iPaaS), которая представляет собой облачную платформу, которая может соединять приложения через их интерфейсы прикладного программирования или API и устанавливать потоки данных между ними.

Например, существуют различные промежуточные программные продукты, которые устанавливают соединение между веб-сервером и системой баз данных. Это позволяет пользователям запрашивать данные из базы данных с помощью форм, отображаемых в веб-браузере. Взамен веб-сервер возвращает динамические веб-страницы в соответствии с запросами и профилем пользователя.

Веб-хуки (Webhooks)

Также известные как обратные вызовы HTTP, представляют собой сообщения в реальном времени, отправляемые одной системой другой, когда происходит определенное событие [9]. Веб-хук срабатывает, когда в вашем приложении, в CRM, чат-боте или иной системе происходит какое-то событие. Например, пользователь написал комментарий или в систему учета товаров добавили новый товар. Когда происходит такое событие, сервер создает HTTP-вызов и отправляет его на адрес, указанный клиентом для получения веб-хуков [9]. Благодаря этому клиент вовремя получает данные в реальном времени. Также можно настроить веб-хук так, чтобы события в одном приложении вызывали действия в другом. Например, сначала покупатель интернет-магазина создает заказ, затем система отправляет веб-хук в систему продавца, далее система уведомляет продавца и отправляет покупателю счет.

EDI (Electronic Data Interchange)

Электронный обмен данными (EDI) — это обмен деловыми документами в стандартизированном формате в рамках компании. Альтернативой EDI является стандартный электронный формат, который компании используют для замены своих бумажных документов, таких как заказы на покупку. Автоматизация бумажных транзакций помогает компаниям не только экономить время, но и избегать дорогостоящих ошибок, которые могут возникнуть в результате ручной обработки. EDI обычно происходит двумя способами: через VAN (value-added networks), в которой сторонняя сеть отвечает за передачу данных, или напрямую через Интернет.

Способы обмена сообщениями для передачи информации по интеграции

В целом решения по интеграции приложений используют сочетание двух конкретных типов обмена сообщениями для передачи и получения соответствующей информации между источниками данных и приложениями: синхронный обмен и асинхронный обмен.

Синхронный обмен или метод (также называемый запрос-ответ) — это метод взаимодействия, когда приложение отправляет запрос другому приложению и не может продолжать работу с другими функциями, пока не получит ответ [10]. Соединение, сформированное этим методом обмена, остается открытым на время обмена данными и зависит от промежуточного программного обеспечения для предотвращения ненужных задержек из-за ошибок и тайм-аутов. Например, веб-сервер, показывающий содержимое сайта онлайн-покупателям с помощью браузера в режиме реального времени, представляет собой синхронный обмен с использованием API протокола передачи гипертекста (HTTP). Другим примером может быть представитель службы поддержки, нажимающий «позвонить» для определенного контакта в системе управления ресурсами клиентов (CRM), и контактная информация клиента передается в набор средств связи для автоматического набора номера.

Синхронный обмен идеально подходит для последовательной обработки, но его зависимость от подтвержденных ответов может замедлить рабочий процесс, если промежуточное ПО на серверной части неоптимально и часто запрашивает тайм-аут или не работает из-за отсутствия ответа.

Асинхронный обмен

В отличие от своего синхронного, этот метод обмена не зависит от подтвержденного ответа для продолжения операций. Соединения закрываются сразу после отправки запросов, а операции могут выполняться параллельно и в любом порядке [10]. Управление API более надежное, и в результате этот метод более масштабируемый, быстрый и универсальный, чем синхронный обмен. Методы интеграции асинхронного обмена используют несколько различных типов обмена сообщениями:

- «Request-Callback» — это асинхронный вариант Request-Reply. Приложение, отправляющее информационные запросы, не «держит очередь»; вместо этого он устанавливает определенное действие, называемое обратным вызовом, для обработки любого отправленного ответа и продолжения других операций по мере необходимости [10]. Это ускоряет и упрощает автоматизированные задачи, такие как извлечение одним приложением информации из нескольких других приложений, сопоставление и анализ полученной информации (сортировка на основе зависимостей), а затем создание отчета после получения всей необходимой информации.
- «Fire and Forget» — это односторонний метод асинхронного обмена. Запрашивающее приложение отправляет один (или, что более вероятно, несколько) запросов другим приложениям и продолжает работу, не получив подтверждения. Например, если персонал использует решение CRM для ввода новой информации о клиенте, CRM может автоматически обновлять вашу систему ERP новыми контактными данными, и наоборот.
- Маршрутизация сообщений — это вариант «Fire and Forget», который интегрирует промежуточное программное обеспечение для координации запросов от нескольких приложений, обслуживаемых одним принимающим приложением для обработки [10]. Правила маршрутизации и разделения

создаются для обеспечения того, чтобы данные обрабатывались в правильном порядке для выполнения следующего шага в каждом процессе (например, создание отчетов или создание счетов для клиентов с использованием информации, полученной из нескольких разных систем).

- Публикация и подписка – это инверсия метода маршрутизации сообщений. Принимающие приложения создают правила для типов информации, которую они хотят получать, и используют промежуточное программное обеспечение для извлечения необходимой информации из различных источников по мере необходимости. Например, корпорация, использующая централизованное ERP-решение, может получать информацию и наполнять эту систему данными от CRM, программного обеспечения для выставления счетов, пакетов для анализа финансовых и больших данных и т. д. Этот метод позволяет различным приложениям выступать в роли отправителей и получателей, создавая сложную, но мощную сеть приложения с перекрестными соединениями, передающие данные в единый пользовательский интерфейс с максимально возможным качеством и скоростью передачи данных.

Выводы

В статье были рассмотрены существующие и наиболее актуальные виды интеграций систем и приложений, приведены примеры их использования при решении бизнес задач компании, рассмотрены способы и методы создания интеграций в зависимости от поставленных целей и задач бизнеса, а также приведены преимущества использования интеграций. В результате можно сказать, что использование интеграций систем и приложений позволяет наиболее гибко, быстро и эффективно объединить несколько приложений, модулей, компонентов системы в единую экосистему.

Список литературы:

1. Kaseya - System Integration: Purpose, Types, Methods & Benefits – [Электронный ресурс]. – URL: <https://www.kaseya.com/blog/2022/04/07/system-integration/> (дата обращения: 21.11.2022)
2. Думченков, И. А. Обзор методов интеграции информационных систем, их преимуществ и недостатков / И. А. Думченков. – Текст: непосредственный // Молодой ученый. – 2018. – № 23 (209). – С. 176-177. – URL: <https://moluch.ru/archive/209/51296/> (дата обращения: 21.12.2022).
3. Андрей Екс - Роль интеграции ИТ-систем в цифровом преобразовании [Электронный ресурс]. – URL: <https://andreyex.ru/informaciya/rol-integratsii-it-sistem-v-tsifrovom-preobrazovanii/> (дата обращения: 21.11.2022)
4. Monterail Team - Guide to Application Integration for Beginners [Электронный ресурс]. – URL: <https://www.monterail.com/blog/application-integration-beginners-guide> (дата обращения: 21.11.2022)

5. Abe Dearmer - The Five Types of Data Integration [Электронный ресурс]. – URL: <https://www.integrate.io/blog/the-five-types-of-data-integration/> (дата обращения: 21.11.2022)
6. Keith D. Foote - The Fundamentals of Data Integration [Электронный ресурс]. – URL: <https://www.dataversity.net/the-fundamentals-of-data-integration/#> (дата обращения: 21.11.2022)
7. Folio3 Microsoft Dynamics Partner - What is System Integration? | 4 Types of System Integration – Advantages of Each Method Integration [Электронный ресурс]. – URL: <https://dynamics.folio3.com/blog/system-integration/> (дата обращения: 21.11.2022)
8. Кирилл Залата - Описание требований к интеграции (часть 2). API [Электронный ресурс]. – URL: <https://www.artofba.com/post/systems-integration-api> (дата обращения: 21.11.2022)
9. Sadhana Balaji - What are Webhooks and Why You Can't Afford to Ignore Them [Электронный ресурс]. – URL: chargebee.com/blog/what-are-webhooks-explained/ (дата обращения: 21.11.2022)
10. Keith Murphy - What You Need to Know About Application Integration [Электронный ресурс]. – URL: <https://planergy.com/blog/application-integration/> (дата обращения: 21.11.2022)

References:

1. Kaseya - System Integration: Purpose, Types, Methods & Benefits - [Electronic resource]. – URL: <https://www.kaseya.com/blog/2022/04/07/system-integration/> (date of access: 11/21/2022)
2. Dumchenkov, I. A. Overview of information system integration methods, their advantages and disadvantages / I. A. Dumchenkov. – Text: direct // Young scientist. - 2018. - No. 23 (209). – S. 176-177. – URL: <https://moluch.ru/archive/209/51296/> (date of access: 12/21/2022).
3. Andrey Eks - The role of IT systems integration in digital transformation [Electronic resource]. – URL: <https://andreyex.ru/informaciya/rol-integratsii-it-sistem-v-tsifrovom-preobrazovanii/> (date of access: 11/21/2022)
4. Monterail Team - Guide to Application Integration for Beginners [Electronic resource]. – URL: <https://www.monterail.com/blog/application-integration-beginners-guide> (Accessed: 11/21/2022)
5. Abe Dearmer - The Five Types of Data Integration [Electronic resource]. – URL: <https://www.integrate.io/blog/the-five-types-of-data-integration/> (accessed 11/21/2022)
6. Keith D. Foote - The Fundamentals of Data Integration [Electronic resource]. – URL: <https://www.dataversity.net/the-fundamentals-of-data-integration/#> (date of access: 11/21/2022)

7. Folio3 Microsoft Dynamics Partner - What is System Integration? | 4 Types of System Integration - Advantages of Each Method Integration [Electronic resource]. - URL: <https://dynamics.folio3.com/blog/system-integration/> (accessed 11/21/2022)
8. Kirill Zalata - Description of the requirements for integration (part 2). API [Electronic resource]. - URL: <https://www.artofba.com/post/systems-integration-api> (date of access: 11/21/2022)
9. Sadhana Balaji - What are Webhooks and Why You Can't Afford to Ignore Them [Electronic resource]. - URL: chargebee.com/blog/what-are-webhooks-explained/ (accessed 11/21/2022)
10. Keith Murphy - What You Need to Know About Application Integration [Electronic resource]. - URL: <https://planergy.com/blog/application-integration/> (date of access: 11/21/2022)