Описание интеграционных решений информационной системы и особенности ее использования

к.э.н., доцент кафедры экономическая кибернетика ЮФУ
Тимакин О.А.
бакалавр 4 курса экономфака ЮФУ
направление подготовки «Бизнес-информатика»
профиль «Бизнес-Аналитика» **Радзивон В.**

Интеграция — это не просто соединение модулей информационной системы. При разработке плана интеграции исходят прежде всего из стратегических целей развития предприятия, возможного изменения бизнес-логики, в соответствии с которой выстраиваются бизнес-процессы и осуществляется их информационное сопровождение. Стоимость и сложность проекта интеграции складывается из решения о оптимальной степени интеграции.

Процесс и степень интеграции зависит от модели связи модулей информационной системы компании. Рассмотрим стандартизированную, обобщенную схему взаимодействия модулей. Модуль TPS поддерживает производственные процессы, а так же, является главным источником для других информационных модулей. ESS — аккумулирует данные полученные от внутренних систем и внешней среды. [4]

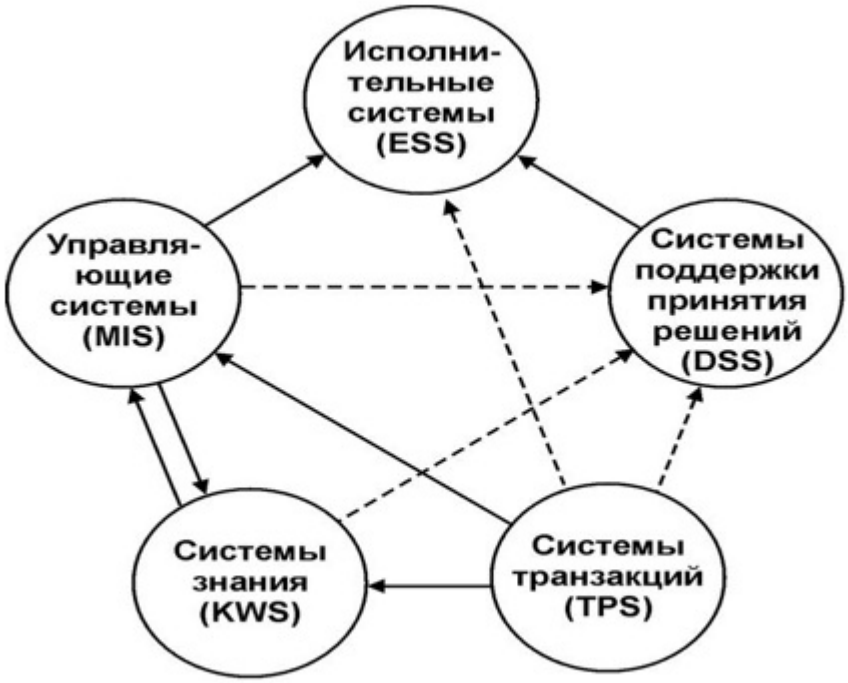

Рисунок 1 – Схема взаимодействия модулей информационной системы

Связь между DSS и модулями TPS, KWS, MIS являются неопределенной, так как чаще всего предприятия характеризуются средним или низким уровнем автоматизации процессов. Как правило, модуль DSS изолирован от основных производственных информационных систем, но использует их информационные потоки для работы своих аналитических систем. [4]

Интеграция может производиться на разных уровнях, в зависимости от факторов, сопровождающих процесс интеграции. Возможна интеграция на уровне форматов, баз данных, программно-аппаратных и сетевых устройств, пользовательских интерфейсов, форм и шаблонов документооборота, программных приложений и т.д.

Подход к разработке и внедрению КИС, основанный на интеграции приложений, позволяет:

- эффективно использовать сделанные ранее инвестиции;
- сократить временные и финансовые затраты на поддержку и развитие информационного пространства компании;
- выбирать самые эффективные для конкретной задачи способы решения, независимость от одной определенной платформы;
- расширение отдельных возможностей существующих информационных систем с уже накопленными в них данными. [1]

Для лучшего понимания разных способов интеграции, необходимо рассмотреть условия их возникновения. Проанализируем факторы, усложняющие интеграцию и приведем варианты минимизации негативного влияния этих факторов (способов интеграции):

- Концептуальная разница в архитектуре — основывается та том, что разработчики разных систем изначально приняли разные решения, предположения и допущения, которые концептуально не стыкуются между собой. Решается введением еще одного слоя абстракции, который концептуально не противоречит обоим подходам.

При этом, есть два варианта реализации:

    - Интеграция на уровне сервисов, когда получившаяся система становится централизованной, а две и более интегрируемых системы превращаются в подсистемы;
    - Интеграция на уровне брокеров, когда мы используем архитектуру брокера (посредника, не являющегося центром), при этом системы остаются независимыми, а брокер обеспечивает прослойку между ними.
- Технологическая разница — когда мы имеем несовместимые форматы обмена данными, протоколы взаимодействия и интерфейсы. Решается, с помощью интеграции на уровне интерфейсов, а именно написанием конвертов, прослоек.
- Несовместимость лицензий, решение может быть в каждом случае индивидуальное, на организационном уровне.

Таким образом можно выделить ряд способов интеграции:

1. Интеграция на уровне брокеров. Преимуществом является то, что этот способ универсален — практически всегда можно создать дополнительный программный модуль, который будут обращаться в обе системы. Однако создание такого модуля процесс сложный и трудоемкий, что влечет за собой высокую стоимость разработки, внедрение и владения.
2. Интеграция на уровне сервисов.

В настоящее время при формировании информационной инфраструктуры предприятия, при проектировании и реализации КИС всё чаще применяется сервис-ориентированная архитектура (Service-Oriented Architecture — SOA). Так называется архитектура, которая стоится из набора однородных слабосвязанных компонентов, называемых сервисами. В основе SOA - использование локализованного множества функций, которые в большинстве случаев, контролируются разными владельцами. В данной архитектуре используются такие базовые понятия: «информационная услуга» и «композитное приложение».

Информационная услуга (сервис) — это отдельная прикладная функция, используемая при разработке приложений, реализующих прикладную логику автоматизируемых процессов как в самой системе, так и для использования в приложениях других автоматизированных систем. [4]

Сервис обычно характеризуется следующими свойствами:

- возможность многократного использования;
- сервис может быть определен одним или несколькими технологически независимыми интерфейсами;
- сервисы слабо связаны между собой и каждый из них может быть вызван посредством коммуникационных протоколов, обеспечивающих возможность взаимодействия.

Использование такого подхода при построении архитектуры сложных интегрированных информационных систем позволяет:

- создать систему композитных приложений, созданных в системе Web-сервисов;
- организовать интеграцию приложений с учетом необходимости автоматизации бизнес-процессов;
- значительно ускорить разработку прикладных приложений и снизить материальные затраты на эти цели.

Обязательным условием использования архитектуры системы на основе SOA является использование единой инфраструктуры описания сервисов, разрешенных протоколов доступа и обмена сообщениями, форматов сообщений.

Данная единая инфраструктура образует так называемую интеграционную шину (Enterprise Service Bus — ESB), которая является одним из центральных компонентов системы. Она устанавливает правила взаимодействия между приложениями интегрируемых систем. Это упрощает управление, поддержку используемых приложений, а также снижает риск фрагментации данных. Основные компоненты архитектуры информационной системы, построенной на основе концепции SOA и ESB представлены на рисунке 2. [4]

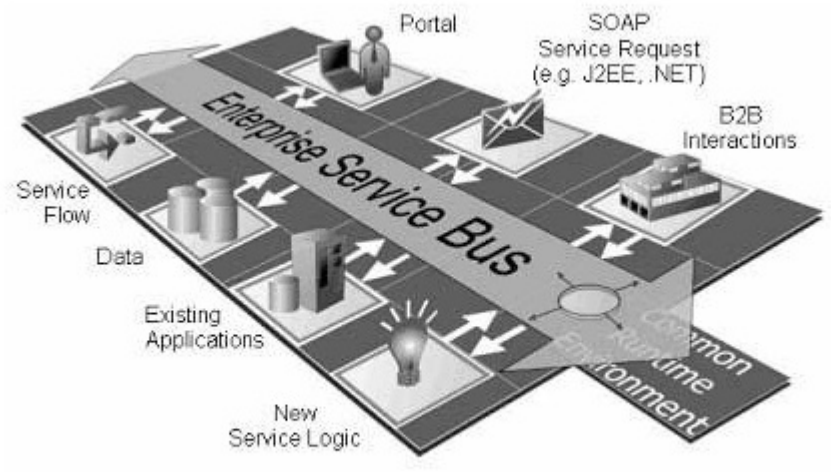

Рисунок 2 – Архитектура SOA

1. Интеграция на уровне физических, программных и пользовательских интерфейсов

   Интеграция происходит с использованием информационных подсистем, реализованных стандартными программными приложениями с открытыми интерфейсами (Open Application Programming Interface). Подобные интерфейсы разрабатываются, например, на базе семейства международных стандартов POSIX.

   POSIX (portable operating system interface for Unix) — набор стандартов, описывающих интерфейсы междуоперационной системой и прикладной программой, библиотеку языка C и набор приложений и их интерфейсов. Стандарт создан для обеспечения совместимости различных UNIX-подобных операционных систем и переносимости прикладных программ на уровне исходного кода, но может быть использован и для не-Unix систем. [5]

2. Интеграция на уровне данных

   Проблемной стороной интеграции на уровне данных, является обилие форматов и типов (неструктурированные, частично-структурированные, жёстко-структурированные) данных, а также значительное накопление их объёмов. Использование разнородных данных приводит создает множество проблем на всех этапах обработки информации, начиная с сбора, заканчивая передачей пользователю для принятия делового решения.

   Для их интеграции в настоящее время обычно используют стандартные интерфейсы и протоколы, например, SQL и JDBC/ODBC, применяют различные инструменты реляционных баз данных (Relational Database — RD), сквозных репозиториев — баз данных с "надстройкой", содержащей информацию об артефактах и объектах проектирования, надмножество словарей метаданных (Transparent Repository — TR) и современных хранилищ и фабрик данных (Data Warehouse, Data Factory — DW, DF). [4]

3. Интеграция на уровне пользователя — это частно возникающая в современных реалиях

ситуация. Так называемая, не автоматизированная интеграция, когда пользователи своими силами и навыками перемещают данные между системами через «копипаст», почту, флешки. Данные методы исключены из рассмотрения в исследовании, но они имеют место в период, пока программные системы не готовы, а скорость развития компании не позволяет ждать.

Наиболее распространены комплексные решения в области интеграции корпоративных информационных систем на базе следующих продуктов:

- IBM Web Sphere (IBM)

  Программное обеспечение WebSphere для сред SOA обеспечивает поддержку динамических взаимосвязанных бизнес-процессов и предоставляет высокоэффективные инфраструктуры приложений для любых бизнес-ситуаций.

- Sonic ESB, Sonic MQ (Progress Software)

  Sonic ESB - это сервисная шина предприятия (ESB), сочетающая в себе обмен сообщениями на основе стандартов, Web-сервисов, XML-преобразования, интеллектуальной маршрутизации для надежного соединения и координации взаимодействия. Sonic ESB использует облегченную и гибкую топологию шины, не имеющую архитектурных ограничений, именно это обеспечивает масштабируемость. [6]

- MS BizTalk (Microsoft);

  Microsoft BizTalk Server — программный продукт компании Microsoft, обеспечивающий возможность автоматизации и управления бизнес-процессами на внутрикорпоративном и межкорпоративном уровне. Дает возможность создания локализованных бизнес-процессов, объединяющих различные приложения внутри предприятия, а также обеспечивающих взаимодействие с партнерами организации через локальную сеть интернет.

Типичные примеры использования BizTalk:

- Синхронизация данных между системами. Готовые данные в определенном формате выкладываются в файлы. BizTalk-процесс проверяя нужный каталог, забирает файлы. Данные из файлов преобразуются во внутренний формат (Xml). Приложения, подписанные на эти данные, получают их. Но данные предварительно преобразовываются в приемлимый для этих приложений формат. Файлы хранятся в BizTalk до тех пор, пока принимающая сторона не примет их.

- Композитный распределенный сервис. Приложение обращается к Web-сервису за данными, он запускает BizTalk процесс, который обращается к другим Web-сервисам за необходимыми данными, после чего консолидирует их и выдает их первому приложению. (Это типичный пример создания композитных Web-сервисов).

Рассмотрим пример использования интегрированной системы для расширения возможностей онлайн ритейла.

Персонализация электронной коммерции Lindt master Swiss chocolatier с IBM WebSphere и CrossView. Данный случай является примером интеграции на уровне сервисов.

Описание использованных для интеграции программ:

CrossView является ведущим поставщиком кросс - канальных торговых решений.

Использование кросс-канальных решений позволяет контролировать и оптимизировать все используемые все каналы продаж сразу, а это способствует улучшению координации цепочки поставок и максимизации возможности сбыта и послепродажного обслуживания. Четкость и слаженность, формирует у клиента доверительное отношение и скорей всего воспользуется вашими услугами снова.

Программное обеспечение WebSphere обеспечивает поддержку динамических взаимосвязанных бизнес-процессов и предоставляет высокоэффективные инфраструктуры приложений для любых бизнес-ситуаций. В данном примере использована функция WebSphere обеспечивающая облачные вычисления.

Облачные вычисления — технология распределённой обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как Интернет-сервис.

Облако – это новая технология использования серверных ресурсов, помогающая задействовать всю доступную мощность процессоров и объем оперативной памяти, разделяя их между различными независимыми задачами.

Одно из главных преимуществ облака, помимо независимости каждого пользователя от остальных, является возможность плавно регулировать объем используемых ресурсов и, соответственно, оплачивать только те ресурсы, которые действительно требуются для решения задачи

IBM Cloud возможно повысить ценность уже имеющегося бизнеса и технологий. Именно облачные технологии обеспечивают интеграцию служб в масштабе всего предприятия и помогают ускорить внедрение инноваций.

Шведская компания одна из первых в этой отрасли начала занимать нишу электронной коммерции посредством запуска облачного веб-магазина на основе IBM и CrossView. Облако позволяет объединить технологии оптовых продаж, технологии ритейла, осуществления экспертизы и эффективно управлять услугами веб-магазина.

Целью Lindt является использование современных технологий для создания более персонализированной связи с клиентами. Что бы дать любителям шоколада более широкий доступ к продукции, вне обычных ритейловых магазинов через которые основная масса продукции поступает потребителям США.

Гибкость IBM WebSphere платформы позволяет Lindt быстро создавать и запускать персонализированные сезонные предложения, в частности на праздники для которых шоколад лучший подарок, такие как День Святого Валентина, Рождество.

Помимо персонализированных промо-акций, Lindt используют свой веб-магазин, чтобы представить и предложить весь диапазон продукции. Покупатели могут смешивать и сочетать продукты, для создания персонализированных подарков.

Использование интегрированной системы для обеспечения работы веб-магазина Lindt переводит их как онлайн ритейлеров на новый уровень. В итоге можно выделить следующие достижения: увеличение конверсии более чем в два раза; увеличение доходов от пользователей с мобильных устройств.

Список использованных источников

1. Обзор терминологии SOA: Часть 1. Сервис, архитектура, управление и бизнес-термины [Режим доступа]: https://www.ibm.com/developerworks/ru/library/ws-soa-term1/
2. Customer story: Markham Stouffville Hospital [Режим доступа]:

https://customers.microsoft.com/Pages/CustomerStory.aspx?recid=13797
3. Lindt masters personalized e-commerce with IBM and CrossView [Режим доступа] http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=AB&infotype=PM&appname=SWGE_UV_LJ_USEN&htmlfid=UVC12377USEN&attachment=U
4. Лекция 5 – Интеграция информационных систем предприятия [Режим доступа]: http://www.intuit.ru/studies/courses/13833/1230/lecture/24065
5. Википедия: POSIX [Режим доступа]: https://ru.wikipedia.org/wiki/POSIX
6. Sonic SOA Suite v7.0 - Передовая реализация интеграционной ESB-платформы промышленного класса [Режим доступа]: http://citforum.ru/seminars/cbd2006/sonic/