

Интеграция приложений и информационных систем

1. Информационные системы предприятия и их подсистемы.

1) системы с эксплуатационным уровнем

системы диалоговой обработки запросов – Transaction Processing Systems (TPS)

2) системы уровня знания

системы работы знания – Knowledge Work System (KWS)

системы автоматизации делопроизводства – Office Automation Systems (OAS)

3) системы уровня управления

управляющие информационные системы – Management Information Systems (MIS)

системы поддержки принятия решений – Decision Support Systems (DSS)

4) системы со стратегическим уровнем

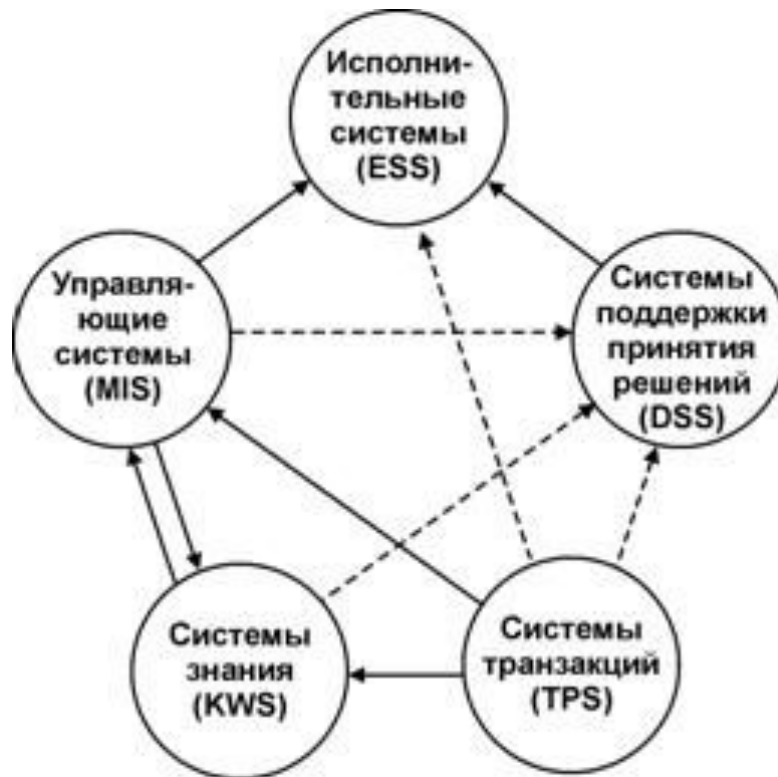
системы поддержки выполнения – Executive Support Systems (ESS)

Взаимосвязь модулей ИС

Модуль **TPS** обслуживает основные производственные и вспомогательные процессы, и обычно это главный источник для других информационных модулей.

ESS - главный получатель данных и внутренних систем из внешней среды.

Связи между **DSS** и совокупностью **TPS**, **KWS**, **MIS** намеренно показаны неопределенными. Иногда **DSS** тесно связана с другими подсистемами. Но это только в том случае, если предприятие отличается высокой степенью автоматизации всех процессов. Обычно подсистема **DSS** изолирована от основных производственных информационных систем и использует их данные и информационные потоки для работы своих аналитических систем.



2. Понятие интеграции.

- **Интеграция ИС** – объединение ИС, связывающее множество документов и отношений в данных системах .
- Под **ИС** понимается множество связанных различными отношениями документов, описывающих некоторые сущности (объекты, факты или понятия).

Стандарт EAI

- EAI (Enterprise application integration) – интеграционная программная структура, объединяющий различного рода приложения, разработанные независимо друг от друга, так, чтобы они работали как одно целое, прозрачно для пользователя

Подходы к интеграции.

- В наши дни чаще всего применяются два подхода: **интеграция по типу «точка-точка»** (point-to-point integration) и **интеграция по шине сервисов** (services bus integration).



3. Проблемы совместимости программных продуктов, ИС.

Информация не найдена

4. Уровни интеграции

Можно выделить 5 уровней интеграции:

- I. **Интеграция бизнес-процессов** – основана на определении, реализации и управлении процессами обмена информацией между различными бизнес-системами.
- II. **Интеграция приложений** – основана на объединении данных или функций одного приложения с другим, благодаря чему обеспечивается интеграция, близкая к реальному времени.
- III. **Интеграция данных** – основана на идентификации и каталогизации данных с целью их дальнейшего использования.
- IV. **Интеграция на основе стандартов** – основана на использовании стандартных форматов данных (например, CORBA, JavaRMI, XML).
- V. **Интеграция платформ** – касается процессов и инструментов, с помощью которых системы могут осуществлять безопасный и оптимальный обмен информацией.

5. Характеристика уровня интеграции бизнес-процессов.

- **Интеграция бизнес-процессов** представляет собой автоматизацию бизнес-процессов организации на основе единой инфраструктуры по созданию и управлению бизнес-процессами. Такая интеграция позволяет объединить в единый бизнес процесс действия, выполняемые в разных прикладных системах. Такая интеграция позволяет:
 - моделировать бизнес-процессы;
 - обеспечить соблюдение правил выполнения бизнес процессов;
 - предоставить пользователем единый интерфейс для выполнения задач в рамках бизнес процессов;
 - обеспечить контроль над выполнением и аудит бизнес процессов;
 - вносить изменение в бизнес процессы в соответствии с требованиями бизнеса;
 - получить данные для анализа выполнения и оптимизации бизнес процессов.

6. Характеристика уровня интеграции приложений.

- **Интеграция приложений по данным** представляет собой организацию взаимодействия приложений посредством передачи данных, между этими приложениями, без модификации или с минимальной модификацией самих приложений.
- При этом данные могут передаваться как в исходном виде, так и с выполнением необходимых преобразований.

7. Характеристика уровня интеграции данных.

- Гарантия качественной интеграции приложений и бизнес-процессов - это **интеграция данных и систем баз данных**.
- На этом уровне в целях интеграции данные должны быть:
 - 1) идентифицированы (то есть указано их местоположение в распределенной системе);
 - 2) каталогизированы;
 - 3) должна быть построена модель метаданных (т.е. описание данных о данных).
- После завершения трех этих этапов данные можно совместно распространять или использовать в системах баз данных.

8. Характеристика стандартов интеграции.

- Среди этих стандартов известны спецификации:
 1. **COM / DCOM** (Component Object Model / Distributed Component Object Model) фирмы Microsoft;
 2. Enterprise Java Beans – **EJB** (основной конкурент DCOM) с протоколом Java Remote Method Invocation (**Java RMI**) фирмы Sun Microsystems;
 3. спецификации компонентов в архитектуре **CORBA**, поддерживаемые консорциумом OMG;
 4. стандарты компонентной разработки Web-приложений, предложенные консорциумом World Wide Web Consortium (W3C) - **XML** (англ. eXtensible Markup Language — расширяемый язык разметки)

- Как правило, средствами интеграции приложений в данной группе средств выступают службы программного обеспечения промежуточного слоя (**middleware**).
- Такие службы иногда называют связующим программным обеспечением.
- Они обеспечивают прозрачную работу приложений в неоднородной сетевой среде, предоставляя им услуги в виде **интерфейсов прикладного программирования (API)**, чтобы обеспечить взаимодействие частей приложений, распределенных по разным узлам корпоративной сети.
- К службам middleware, прежде всего, относятся службы вызова удаленных процедур RPC (Remote Procedure Call), обмена сообщениями и посредники (брокеры) запросов к объектам ORB (Object Request Brokers), мониторы транзакций.

- Благодаря использованию указанных выше стандартов при компонентной разработке приложений, становится возможным широко реализовать на практике преимущества повторного использования компонентов – повышение производительности труда при разработке, простоту применения, единообразию структуры приложений.

9. Характеристика уровня интеграции платформ.

- Чтобы завершить интеграцию систем - базовой архитектуры, аппаратного и программного обеспечения - необходимо интегрировать разнесенные части гетерогенной сети (т.е. имеются разные машинные архитектуры и операционные системы).
- Интеграция платформ касается процессов и инструментов, с помощью которых эти системы могут осуществлять безопасный и оптимальный обмен информацией. В результате, данные могут беспрепятственно передаваться по различным приложениям.

10. Качество программного интерфейса.

- **Индекса качества программного интерфейса** можно измерять в диапазоне от нуля до единицы, от полного отсутствия какого бы то ни было программного интерфейса до наличия исчерпывающе полного (в смысле доступности прикладной функциональности) программного интерфейса.

11. Открытость программного интерфейса.

- **индекс открытости программного интерфейса** - измеряется в пределах от нуля до единицы, от полностью закрытого (ничего не опубликовано), до полностью открытого интерфейса (опубликован интерфейс ко всем прикладным функциям приложения).

12. Интегрируемость программного интерфейса.

- **Индекс интегрируемости приложения** можно определить как индекс качества программного интерфейса, помноженный на индекс открытости программного интерфейса. В результате мы получим числовой показатель, который (в известной степени) характеризует способность приложения быть частью какого-то другого, глобального приложения (сейчас популярен термин композитное приложение).

13. Принцип открытости ИС

- **Открытая система** - исчерпывающий и согласованный набор международных стандартов на информационные технологии и профили функциональных стандартов, которые специфицируют интерфейсы, службы и поддерживающие их форматы, чтобы обеспечить взаимодействие и мобильность программных приложений, данных и персонала

Общие свойства открытых информационных систем:

- 1) расширяемость/масштабируемость;
- 2) мобильность/переносимость;
- 3) взаимодействие;
- 4) стандартизуемость;
- 5) дружелюбность к пользователю

14. Понятие композитного приложения

- **Композитное (составное) приложение** - программное решение для конкретной прикладной проблемы, которое связывает прикладную логику процесса с источниками данных и информационных услуг, хранящихся на гетерогенном множестве базовых информационных систем.
- Обычно композитные приложения ассоциированы с процессами деятельности и могут объединять различные этапы процессов, представляя их пользователю через единый интерфейс.

15. Архитектура ИС.

- **Архитектура информационной системы** — концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.
- **Файл-серверная архитектура**
- **Клиент-серверная архитектура**
 - 1) Тонкий клиент
 - 2) Толстый клиент
- **Трехслойная архитектура**

Файл-серверная архитектура

- Как исполняемые модули, так и данные размещаются в отдельных файлах операционной системы.
- Для хранения данных используется выделенный сервер (отдельный компьютер), который и является файловым сервером. Исполняемые модули хранятся либо на рабочих станциях, либо на файловом сервере.

Клиент-серверная архитектура

- Клиент (исполняемый модуль) запрашивает те или иные сервисы в соответствии с определенным протоколом обмена данными. При этом, в отличие от ситуации с файловым сервером, нет необходимости в использовании прямых путей операционной системы.

Трехслойная архитектура

- Базируется на дальнейшей специализации компонент архитектуры: клиент занимается только организацией интерфейса с пользователем, сервер баз данных - только стандартизованной обработкой данных. Для реализации логики обработки данных архитектура предусматривает отдельный слой - слой бизнес-логики.

16. Классификация ИС по архитектуре.

- По степени распределённости отличают:
 - I. Локальные ИС*, в которых все компоненты (БД, СУБД, клиентские приложения) работают на одном компьютере;
 - II. Распределённые (distributed) ИС*, в которых компоненты распределены по нескольким компьютерам.
- Распределённые ИС, в свою очередь, разделяют на
 - 1. файл-серверные ИС* (ИС с архитектурой «файл-сервер»). База данных находится на файловом сервере, а СУБД и клиентские приложения находятся на рабочих станциях.
 - 2. клиент-серверные ИС* (ИС с архитектурой «клиент-сервер»). База данных и СУБД находятся на сервере, а на рабочих станциях находятся клиентские приложения.
- В свою очередь, клиент-серверные ИС разделяют на *двухзвенные* и *многозвенные*.
- В *двухзвенных (two-tier)* ИС всего два типа «звеньев»: сервер баз данных, на котором находятся БД и СУБД, и рабочие станции, на которых находятся клиентские приложения. Клиентские приложения обращаются к СУБД напрямую.
- В *многозвенных (multi-tier)* ИС добавляются промежуточные «звенья»: серверы приложений (*application servers*). Пользовательские клиентские приложения не обращаются к СУБД напрямую, они взаимодействуют с промежуточными звеньями.

17. Архитектура SOA.

- **SOA** (англ. Service Oriented Architecture) — это прикладная архитектура, в которой все функции определены как независимые сервисы с вызываемыми интерфейсами. Обращение к этим сервисам в определенной последовательности позволяет реализовать тот или иной бизнес-процесс

- **Идея SOA** заключается в создании архитектурной платформы, которая обеспечит быструю консолидацию распределенных компонентов — сервисов — в единое решение для поддержки определенных бизнес-процессов.

Принципы SOA:

- 1) Архитектура, как таковая, не привязана к какой-то определённой технологии,
- 2) Независимость организации системы от используемой вычислительной платформы (платформ),
- 3) Независимость организации системы от применяемых языков программирования,
- 4) Использование сервисов, независимых от конкретных приложений, с единообразными интерфейсами доступа к ним,
- 5) Организация сервисов как слабосвязанных компонентов для построения систем

18. Понятие «тонкого» и «толстого» клиента, примеры.

- В рамках направления "клиент-сервер" существуют два основных "диалекта":
 - «ТОНКИЙ» клиент;
 - «ТОЛСТЫЙ» клиент.

- В системах на основе **тонкого клиента** используется мощный сервер баз данных - высокопроизводительный компьютер и библиотека так называемых хранимых процедур, позволяющих производить вычисления, реализующие основную логику обработки данных, непосредственно на сервере.
- Клиентское приложение, соответственно, предъявляет невысокие требования к аппаратному обеспечению рабочей станции.

- **Толстый** или **Rich-клиент** в архитектуре клиент-сервер — это компьютер, обеспечивающий расширенную функциональность независимо от центрального сервера.
- Как правило, сервер в этом случае является лишь хранилищем данных, а вся работа по обработке и представлению этих данных переносится на машину клиента.

Достоинства

- 1) Толстый клиент обладает широким функционалом в отличие от тонкого.
- 2) Режим многопользовательской работы.
- 3) Предоставляет возможность работы даже при обрывах связи с сервером.
- 4) Имеет возможность подключения к банкам без использования сети Интернет.
- 5) высокое быстродействие.

Недостатки

- 1) Большой размер дистрибутива.
- 2) Много в работе клиента зависит от того, для какой платформы он разрабатывался.
- 3) При работе с ним возникают проблемы с удаленным доступом к данным.
- 4) Довольно сложный процесс установки и настройки.
- 5) Сложность обновления и связанная с ней неактуальность данных.

В настоящее время известны следующие реализации толстого клиента:

- **ArchiMed** - медицинская информационная система
- **R-Keeper Delivery** - автоматизация доставки готовой продукции
- **R-Keeper POS-ITV** - система видеоконтроля
- **R-Keeper Self-Service** - система самообслуживания
- **R-Keeper** - автоматизация ресторанного бизнеса
- **SET Prisma** – система контроля кассовых операций
- **Shelter** – автоматизация гостиниц
- **StoreHouse** - учётная система ресторанного бизнеса
- **Time-Keeper** - система учета рабочего времени
- **Абонемент** - автоматизация фитнес и спа-центров
- **АТОЛ** - рабочее место кассира
- Кассовый программный модуль **Кристалл-УКМ**
- Кассовый программный модуль **Супермаг-УКМ**

Примеры тонких клиентов

- **Thinstation** - дистрибутив GNU/Linux, разработанный специально для создания тонких клиентов.
- **LTSP** (англ. Linux Terminal Server Project) — пакет дополнений для GNU/Linux, позволяющий подключить большое количество низкопроизводительных тонких клиентов к серверу.
- **OpenThinClient**
- **Windows CE**
- **Бездисковая станция**
- **Терминальный доступ**
- **Virtual Network Computing**

19. Понятие информационной услуги (сервиса).

- **Информационная услуга** - услуга, ориентированные на удовлетворение информационных потребностей пользователей путем предоставления информационных продуктов.
- **ИТ-сервис** в корпоративной среде – это ИТ-услуга, которую ИТ-подразделение (департамент, отдел, служба) или внешний провайдер предоставляет бизнес-подразделениям предприятия для поддержки их бизнес-процессов.

- В общем случае ИТ-сервис характеризуется рядом параметров :

- ✓ функциональность;
- ✓ время обслуживания;
- ✓ доступность;
- ✓ надежность;
- ✓ производительность;
- ✓ конфиденциальность;
- ✓ масштаб;
- ✓ затраты.

Понятие web-сайтов и порталов, корпоративные порталы.

- **Веб-сайт** (от англ. website: web — «паутина», «сеть» и site — «место», букв. «место в сети») или просто сайт — в компьютерной сети объединённая под одним адресом (доменным именем или IP-адресом) совокупность документов частного лица или организации. По умолчанию подразумевается, что сайт располагается в сети Интернет.
- **Портал** - это информационный сетевой ресурс, позволяющий получать сконцентрированную информацию по определенной теме и изменять её путем взаимодействия человека и компьютера.
- **Корпоративный информационный портал (КИП)** предназначен для создания единого информационного пространства компании и позволяет интегрировать в единое целое разнородные корпоративные приложения, предоставляя им единый интерфейс доступа.

- Функциональность портала определяется потребностями заказчика. Обычно в рамках КИП осуществляются следующие процессы и оказываются следующие сервисы:
 - 1) обеспечение информационной поддержки сотрудников и клиентов компании;
 - 2) организация коллективной работы и взаимодействия удаленных рабочих групп;
 - 3) управление правами доступа, персонализация предоставляемых данных;
 - 4) управление публикациями (размещением и редактированием информации);
 - 5) организация доступа к приложениям и данным через web-браузер с любого компьютера, подключенного к Интернету.

21. Идентификация пользователей – авторизация, аутентификация.

- **Аутентификация** (англ. Authentication) — проверка принадлежности субъекту доступа предъявленного им идентификатора; подтверждение подлинности.
- **Авторизация** (англ. authorization):
 - 1. Процесс предоставления определенному лицу прав на выполнение некоторых действий.
 - 2. Процесс подтверждения (проверки) прав пользователей на выполнение некоторых действий.

- В информационных технологиях посредством авторизации устанавливаются и реализуются права доступа к ресурсам и системам обработки данных.
- **Авторизация** как правило — следующий шаг системы после **аутентификации**.