

Министерство образования и науки Республики Казахстан

ВОСТОЧНО-КАЗАХСТАНСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Д. СЕРИКБАЕВА

И.А. Загайнов

АРХИТЕКТУРА ИНФОРМАЦИОННЫХ СИСТЕМ

Конспект лекций для магистрантов специальностей
6М070300 – «Информационные системы»,
6М070400 – «Вычислительная техника и программное обеспечение».

Усть-Каменогорск
2011

УДК 004.78

Загайнов И.А. Архитектура информационных систем: Конспект лекций для магистрантов специальностей 6М070300 – Информационные системы, 6М070400 – Вычислительная техника и программное обеспечение/ Изд-во ВКГТУ.- Усть-Каменогорск, 2011. –80с.

Конспект мультимедиа лекций (конспект) входит в состав электронного методического пособия по дисциплине.

Конспект содержит текстовую часть всех слайдов мультимедиа лекций в формате MSWord в удобной для распечатки форме.

Конспект лекций используется для:

- внесения пометок при прослушивании лекций;
- записи ответов на обсуждаемые вопросы;
- запись ответов на вопросы тренировочных тестов;
- фиксации последовательности проектирования элементов, и т.д.

В электронном виде конспект удобно использовать для копирования кода примеров скриптов, кода программ обработки событий компонентов.

Утверждено методической комиссией факультета информационных технологий и энергетики.

Протокол № _____ от _____ 2011г.

СОДЕРЖАНИЕ

Введение.....	4
1 Архитектура информационных систем.....	6
2 Системы управления предприятием	12
3 Технологии и методы проектирования.....	25
4 Современные системы управления	33
5 Компоненты инфраструктуры ИС	40
6 Комплекс технологий	46
7 Технологии инфраструктуры	55
8 Классификация систем	60
9 INTERNET/INTRANET/EXTRANET технологии	66
10 Новые тенденции информационных технологий	71
Список литературы	79

ВВЕДЕНИЕ

Краткое описание дисциплины.

Загайнов Иван Александрович – старший преподаватель кафедры «Информационные системы».

Выпускник физического факультета Новосибирского Государственного университета по специальности "Автоматизация физико-технических измерений". На протяжении двадцати лет занимался разработкой и внедрением автоматизированных информационных комплексов на предприятиях города и области ("Проектно-конструкторского бюро АСУ", ОАО "Казахтелеком", СП "Arna-Sprint Data Communications"). Участвовал в открытии телекоммуникационного узла сети передачи данных "KAZNET" в городе Усть-Каменогорске, построении корпоративных сетей на всех крупных предприятиях региона, банках, государственных учреждениях, открытии первого узла доступа к сети Интернет. В течение десяти лет преподает на кафедре информационных систем ВКГТУ им. Д.Серикбаева. Читает курсы «Проектирование информационных систем», «Проектирование приложений ИС». Участвовал в разработке дисциплин, включающих изучение Интернет технологий, Web – технологий, информационных сетей. Комната ГЗ-311, тел. 540-356.

Дисциплина Архитектура информационных систем (AIS 2.2.00) является профилирующей дисциплиной (ПД2), блока по выбору 2.2 Каталога элективных дисциплин специальности.

Читается, как правило, на первом курсе и входит в блок дисциплин для сдачи государственного экзамена.

Современные информационные системы (ИС) поддерживают весь спектр бизнес – процессов предприятия и имеют достаточно сложную инфраструктуру. Компонентами ИС являются программные и аппаратные средства, телекоммуникационная система, математический и лингвистический инструментарий и т.д. В рамках дисциплины слушатели изучают основы проектирования и построения сложных информационных комплексов с использованием различных архитектур, способных обеспечивать работу большого количества пользователей.

В лекциях дисциплины излагается общий теоретический материал, приводятся примеры внедрения ИС в Университете, на предприятиях региона.

Лекции дисциплины читаются в мультимедиа аудитории с демонстрацией основных приемов:

- управления серверным программным обеспечением;
- проектирования и управления серверными модулями;
- разработки интерфейсов пользователей в среде Delphi.

Лекционная программа предусматривает тренировочное тестирование, изучение элементов учебных примеров.

Лабораторный практикум и самостоятельная работа магистрантов с преподавателем (СРМП) проводятся в компьютерных классах Технопарка «Алтай» и учебного вычислительного центра (УВЦ) ВКГТУ. Для выполнения всех заданий лабораторного практикума используется один пример предметной области.

Самостоятельное изучение обеспечено учебными примерами всех компонентов информационной системы, предусматривает работу с электронным учебным материалом на персональном компьютере обучающегося.

Выполнение всех видов работ обеспечивает получение магистрантами навыков работы с технической периодической литературой, поиском дополнительной информации, включая информационные ресурсы сети Интернет. Магистранты учатся самостоятельно осваивать Computer Aided Software Engineering (CASE) - средства разработки ИС.

Знания и навыки, полученные при изучении дисциплины, помогают качественней выполнять магистерские диссертации.

1 АРХИТЕКТУРА ИНФОРМАЦИОННЫХ СИСТЕМ

1.1 Содержание.

Учебный план дисциплины.

Определение информационной системы.

Теоретические основы информационных систем:

- компоненты ИС.
- декомпозиция, подсистемы.
- архитектура ИС.
- структура ИС.

Входной контроль по дисциплинам:

- «Организация вычислительных систем и сетей»;
- «Интернет технологии»;
- «Анализ и моделирование информационных процессов и систем»;
- «Прикладное программирование»;
- «Проектирование информационных систем».

1.2 Целью дисциплины «Архитектура информационных систем» является:

- освоение теоретических основ современных технологий информационных систем и их средств инструментальной поддержки;
- получение практических навыков проектирования функциональных и обеспечивающих компонентов многозвенных информационных систем; подготовка к реализации приложений, разработке архитектуры ИС, темы магистерской диссертации;
- подготовка выполнения плана научно – исследовательских работ и публикации работ магистрантов.

1.3 В результате изучения дисциплины магистранты должны:

- знать качественные и количественные методы анализа систем, методы теоретико-множественного описания систем, основные положения теории управления, методы анализа и синтеза инфраструктуры ИС, основные модели, методы и инструментальные средства, используемые в информационных системах для автоматизации решения интеллектуальных задач;
- уметь выбирать и применять существующие технологии проектирования ИС с учетом сформулированного критерия эффективности, на основе анализа информационных потребностей пользователя, осуществить выбор состава функциональной и обеспечивающих компонентов ИС и состав пользовательского интерфейса, проектировать структуру и типовые подсистемы ИС;

- иметь представление об основных закономерностях функционирования систем и возможностях их системного анализа; о современных методах исследования, оптимизации и проектировании информационных систем обработки информации и управления и их обеспечения; об использовании основных положений теории управления в различных областях науки и техники;

- иметь навыки использования методов теории систем в практике проектирования компонентов систем обработки информации и управления; постановки задачи, использования моделей, методов и средств информационных технологий при создании систем обработки информации и управления; использования языков моделирования для исследования и проектирования.

1.4 Лекции.

Излагается общий теоретический материал, приводятся примеры внедрения ИС на предприятиях региона.

Демонстрируется пример лабораторных работ (телефонный справочник предприятия или тема, предложенная студентами).

Проводится подготовка к рубежному контролю (тестированию).

Методическая помощь в выборе среды проектирования и разработки ИС.

Кто уже работает по специальности?

1.5 Лабораторные.

Выполнение и защита лабораторного практикума по индивидуальным темам (предметным областям).

При выполнении лабораторных работ необходимо использовать лекционный материал, материал учебного примера, операторы методического пособия.

При разработке выбранной предметной области, необходимо использовать как можно больше типов данных, объектов.

Выполняя работу, получить ответы на, приведенные к лабораторной работе, вопросы.

При защите, по заданию преподавателя, производить выполнение указанных операторов, создание объектов, элементов приложений.

При защите лабораторных работ можно пользоваться всеми электронными справочниками. Умение использования справочников оценивается при защите работ.

Курс предусматривает выполнение четырех лабораторных работ.

1.6 Раздел магистерской работы.

Содержит материал по моделированию процессов, инфраструктуре системы.

Защита:

- пояснительная записка;
- демонстрация;
- знание предметной области;
- программные средства.

1.7 Для самостоятельного изучения предлагаются следующие темы:

- изучение свойств и методов компонентов (Help Delphi или MSDN);
- язык гипертекстовой разметки - HTML;
- язык JavaScript;
- проектирование в CASE- среде (MS Visio);
- организация обмена данными в многозвенных ИС.

1.8 Учебный материал.

1. Лекционный материал.
2. Методические материалы лабораторного практикума.
3. Электронные методические пособия.
4. Литература библиотечного фонда.
5. Источники сети Интернет, материалы периодических изданий.
6. Промышленная документация по Delphi, Microsoft Visual Studio.
7. HELP среды разработки приложений Delphi, MSDN.
8. Учебные примеры базы данных и приложений.

1.9 Вопрос.

Какие компоненты относятся к информационной системе?

1. Технические средства (аппаратное обеспечение).
2. Программное обеспечение.
3. Персонал.
4. Бизнес - процессы.
5. Устройства для ручных операций.

1.10 Определение системы.

«Система (system) - это комплекс, состоящий из бизнес-процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям».

1.11 Графическая интерпретация определения системы по ISO/IEC TR 15271.

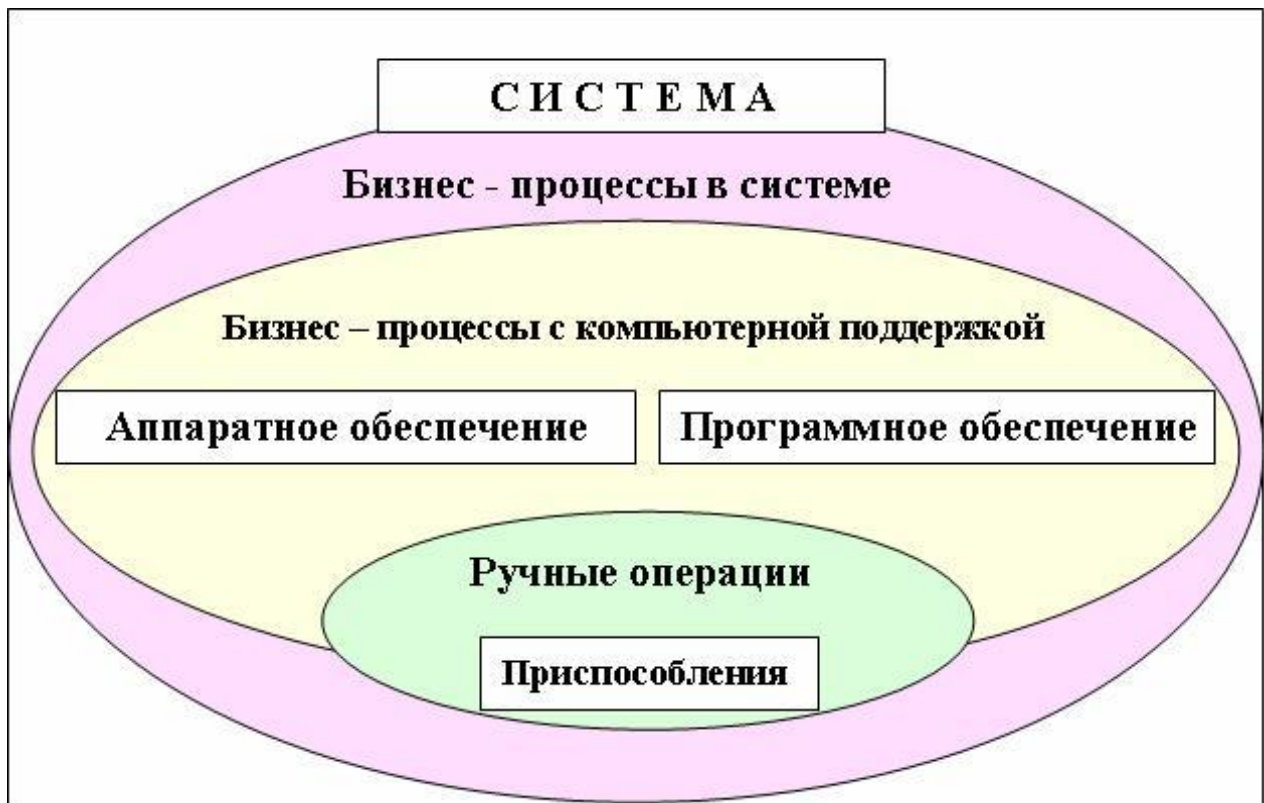


Рисунок 1.1 – Система по ISO/IEC TR 15271

1.12 Компоненты:

1. Аппаратное обеспечение (серверное оборудование, клиентские станции, коммуникационные средства, прикладное оборудование);
2. Программное обеспечение (ПО):
 - серверное ПО (операционные системы, системы администрирования, системы мониторинга, например, сетевой монитор или монитор транзакций);
 - СУБД (Inter Base, Microsoft SQL, MySQL, Oracle);
 - Прикладное ПО (системы управления ресурсами предприятия, системы управления активами и фондами, системы управления взаимоотношениями с клиентами, офисное инфраструктурное обеспечение);
 - ПО, разработанное пользователем;
3. Участники (проектировщики, пользователи);
4. Бизнес – процессы;
5. Прочее обеспечение (математическое, лингвистическое, организационно-нормативное (техническая документация, приказы, должностные инструкции);
6. Все остальное (приспособления).

Подробнее компоненты будут рассматриваться в следующих лекциях.

1.13 Архитектуры информационных систем.

1. Локальная архитектура. Один пользователь, один компьютер, одна среда работы, например:

- база данных;
- BDE;
- клиентское приложение.

Называется, как правило, АРМ – автоматизированное рабочее место.

2. Файл – серверная архитектура. Один файловый сервер, сеть, технология доступа к БД (BDE, IBX, ADO), небольшое количество клиентов, работающих с копиями наборов данных, обладающих одинаковыми правами. Целостность БД обеспечивается клиентскими приложениями.

3. Клиент серверная архитектура. Для управления базами данных используется специализированное программное обеспечение (СУБД), например SQL – сервер «Inter Base».

Уменьшение сетевого трафика (объема передаваемых данных).

Правила поддержки целостности данных переводятся на сервер и одинаковы для всех приложений.

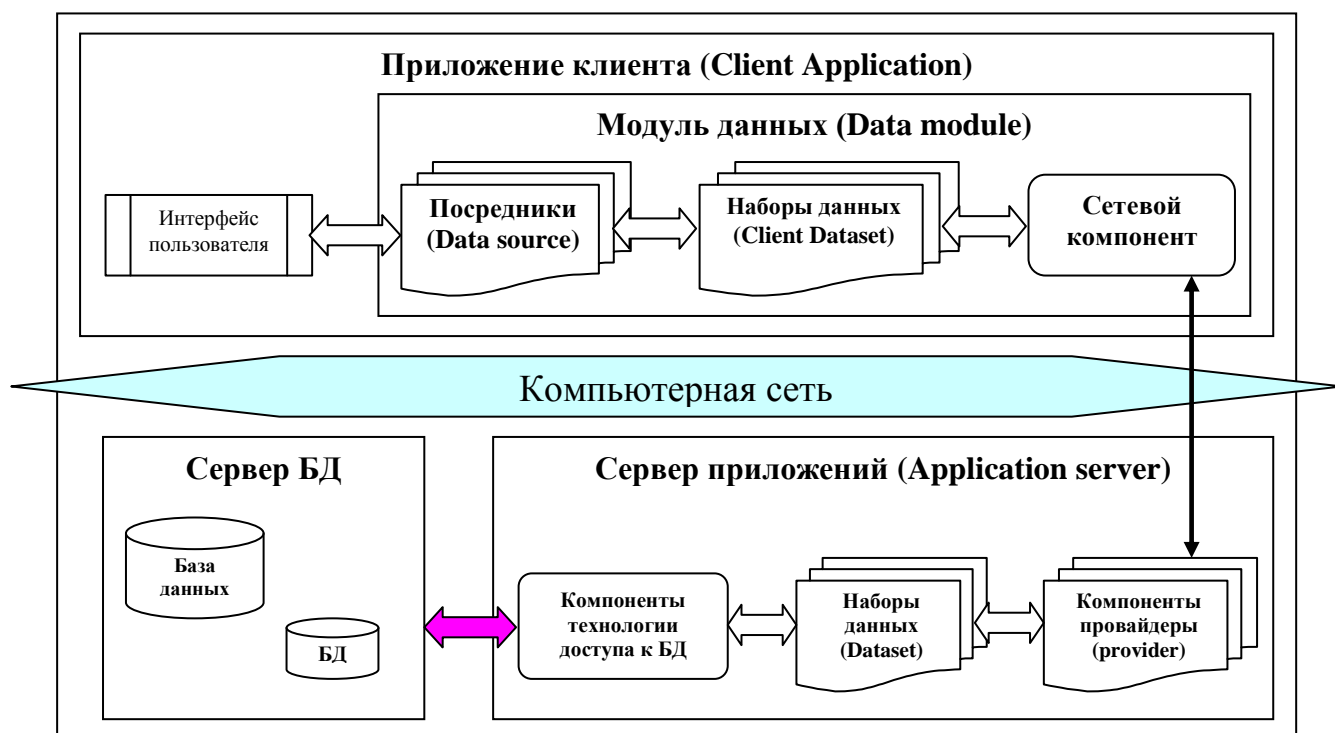


Рисунок 1.2 – Трехзвенная архитектура

1.14 Интернет/Инtranет/Экстранет архитектура.

Использование Интернет как телекоммуникационной среды.

Использование Web – сервера и соответствующих технологий проектирования Web-приложений.

В качестве клиента используется браузер, например, Internet Explorer.

1.15 Список сетевых сервисов:

1. Общее использование вычислительных ресурсов рабочих станций и серверов (процессор, оперативная память) для организации параллельных вычислений.
2. Совместное использование периферийных устройств (принтер, плоттер, сканер, модем, Web - камера).
3. Совместное использование данных и программ в прикладных приложениях, разработанных с применением клиент – серверной или файл - серверной технологии (сетевые программы бухгалтерии, сетевой электронный каталог библиотеки, программы для дистанционного образования).
4. Интернет/Интранет технологии, включая как клиентские сервисы, например, электронную почту или WWW, так и сервисы обслуживания: службу доменных имен или сетевое администрирование.
5. Сетевые услуги реального времени:
 - служба персонального вызова – ICQ;
 - чат – IRC;
 - глобальная служба времени;
 - Интернет – радио;
 - Интернет – видео;
 - Интернет – телефония;
 - электронная коммерция.
6. Сетевые услуги видео-телефонии и видеоконференций.
7. Системы охранного видео-слежения, сигнализации.

1.16 Демонстрация.

Инсталляция серверов.

Инсталляция учебного примера.

Запуск утилит администрирования.

2 СИСТЕМЫ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

Современные системы: системы управления ресурсами предприятий ERP (Enterprise Resource Planning), системы управления активами и фондами EAM (Enterprise Asset Management), системы управления взаимоотношениями с клиентами - CRM (Customer Relationship Management), системы управления цепочками поставок SCM (Supply Chain Management).

CASE среды проектирования.

Классификация методов исследования ИС.

Классификация и характеристика задач, решаемых при исследовании систем.

2.1 Основные системы, используемые в организациях.

Ни одна система не может обеспечить аккумуляцию всей информации для функционирования организации в целом. Рассмотрим классификацию на организационном уровне, приведенную в учебном пособии по курсу «Мастер делового администрирования» (МБА) [1].

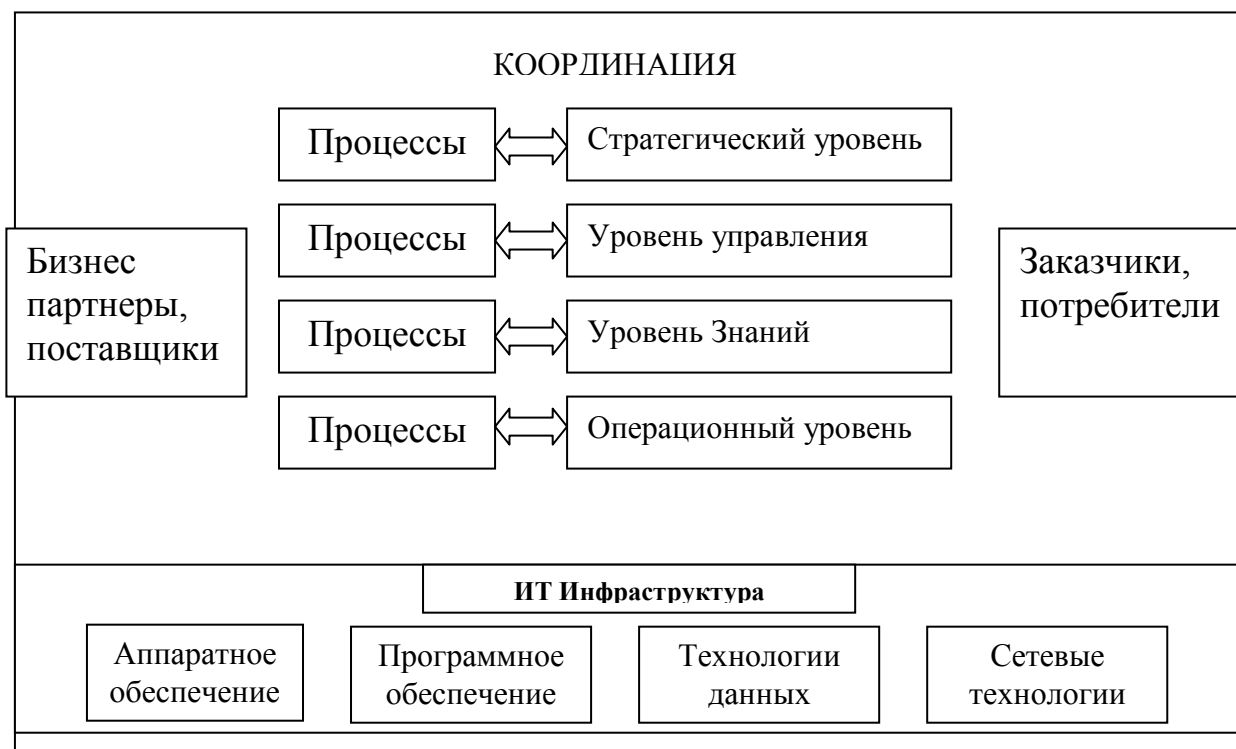


Рисунок 2.1 - Информационная инфраструктура организации

Для работы на различных организационных уровнях предлагается четыре основных типа ИС:

1. Системы стратегического уровня (системы поддержки принятия решений) предназначены для долгосрочного планирования, решения вопросов стратегии сотрудниками верхнего эшелона управления. Основным их предназначением является согласование их с внешней средой. Системы должны помогать находить ответы на такие вопросы:

- каким будет уровень занятости в следующие пять лет?
- какой набор студентов ожидается в ближайшие десять лет?
- в какой области предприятие может рассчитывать на максимальную прибыль?
- какие продукты, услуги будут востребованы в будущем?

2. Системы управленческого уровня используются менеджерами среднего звена, служат для мониторинга, контроля помощи в принятии решений, выполнения административных функций. Эти системы могут предоставлять всю периодическую отчетность, например, по географии проживания контингента учащихся, по данным рубежных и итоговых контролей студентов, по выполнению финансовых планов.

Они могут использоваться при необходимости принятия нестандартных (неструктурированных) решений исходные предпосылки для которых не всегда ясны. Такие системы отвечают на вопросы «что если ...?».

3. Системы уровня знаний представляют собой инженерные, графические, управленческие программные пакеты, осуществляющие обработку информации и подготовку документов. Основное назначение такого программного обеспечения – интеграция новых данных и знаний. Подобные информационные системы являются самыми распространенными во всех функциональных областях.

4. Системы операционного уровня обеспечивают выполнения всех элементарных операций и транзакций внутри организации. Такими операциями могут служить:

- оформление продажи товара;
- регистрация денежных поступлений;
- регистрация успеваемости рубежного контроля студентов;
- оформление движения материалов и т.п.

2.2 Функциональная классификация систем.

Информационные системы также служат для реализации основных бизнес функций, таких как продажи и маркетинг, производство, финансы, бухгалтерия, управление ресурсами (кадрами).

В типичной организации для каждой функциональной области существуют системы различных уровней. Интеграция в одной среде функций различного типа и уровня приводит к разработке интегрированных систем управления предприятием - ERP.

Унификация и стандартизация процессов управления на уровне международных и отраслевых стандартов приводит к возможности многократного использования разрабатываемых систем. Чем больше реальные процессы организации соответствуют стандартам, тем проще внедрять и использовать типовые программные продукты.

ERP системы позволяют интегрировать данные, поступающие в систему из различных функциональных модулей. При этом для использования на верхних уровнях (Рисунок 2.2), как правило, происходит агрегация данных в виде промежуточной отчетности.

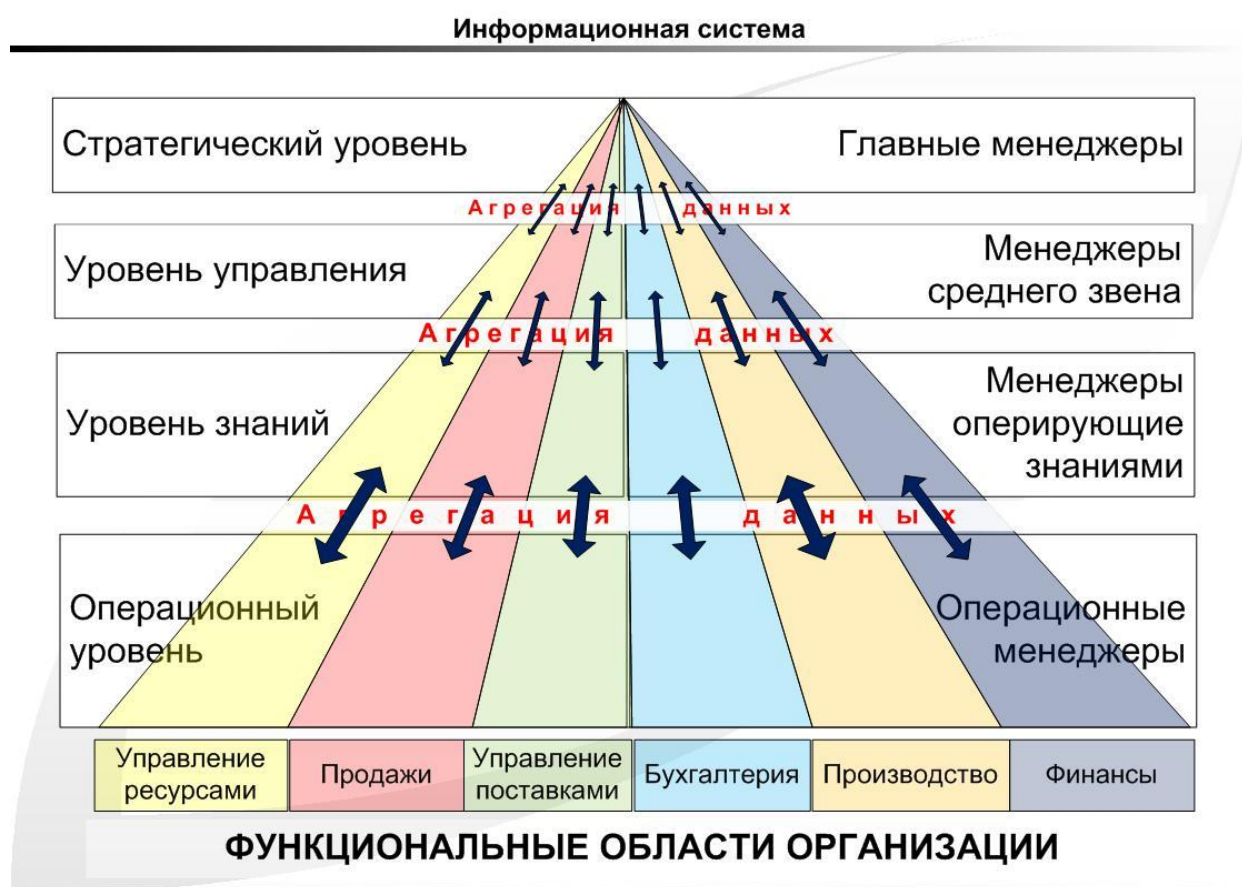


Рисунок 2.2 – Функционально-информационное представление

Интеграция функций и использование единых данных накладывает и ряд специфических ограничений при проектировании, например, разграничение прав доступа (привилегий) пользователей различных уровней к данным.

2.3 Бизнес функция предприятия (БФ) – функциональный базис для всех технологических и административно – хозяйственных процедур.

Существуют три основных свойства бизнес-функции:

- нормируемость (формальные единицы измерения или система координат);

- исчисляемость (масштабируемость);
- возможность количественной оценки.

В пункте 3.1.2 методического указания к курсовому проектированию [2] - пример описания деятельности ВКГТУ (фрагмент устава университета).

2.4 Бизнес-процесс – это описание технологии достижения результата в определенном функциональном базисе. Также это формализованное описание заданных управляемых процедур, включая как выполненные этим набором функции, так и используемые им данные. Состав и взаимоотношения затрагиваемых им организационных подразделений и единиц.

Из этих определений можно сделать вывод, что бизнес-процесс является составной частью бизнес - функции. Им описываются более конкретные задачи проекта (ИС). Множество процессов, объединенных одной функцией, решают множество задач, что обеспечивает достижение единой цели, стоящей перед ИС.

Для моделирования процессов чаще всего используются специальные среды проектирования CASE (Computer Aided Software Engineering).

2.5 Вопрос.

Какие модели проходили в специальных дисциплинах?

2.6 Для обеспечения целостности данных и согласованности процессов в ИС необходимо соблюдать некоторые ограничения, обеспечивающие механизм управления процессами и операциями над данными.

Бизнес - правила (БП) – это механизмы управления БД и предназначены для поддержания БД в целостном состоянии, а также для выполнения других действий, например, накапливания статистики работы с БД. Они также определяют механизмы, согласно которым при изменении одних данных изменяются и связанные с ними данные в той же или других таблицах БД.

Идеология архитектуры «клиент-сервер» требует переноса максимально возможного числа БП на сервер.

2.7 К преимуществам такого подхода относятся:

- гарантия целостности БД, поскольку БП сосредоточены в едином месте (в базе данных);
- автоматическое применение БП, определенных на сервере БД, для любых приложений;
- отсутствие различных реализаций БП в разнотипных клиентских приложениях, работающих с БД;
- быстрое срабатывание БП, поскольку они реализуются на сервере и, следовательно, нет необходимости посылать данные клиенту, увеличивая при этом сетевой трафик;

- доступность изменений, внесенных в БП на сервере, для всех клиентских приложений, работающих с настоящей БД, и отсутствие необходимости повторного распространения измененных приложений клиентов среди пользователей.

2.8 К недостаткам хранения бизнес - правил на сервере можно отнести:

- отсутствие у клиентских приложений возможности реагировать на некоторые ошибочные ситуации, возникающие на сервере при реализации БП (например, игнорирование приложениями, написанными на Delphi, ошибок при выполнении хранимых процедур на сервере);

- ограниченность возможностей SQL и языка хранимых процедур и триггеров для реализации всех возникающих потребностей определения БП.

На практике в клиентских приложениях реализуют лишь такие бизнес - правила, которые трудно или невозможно реализовать с применением средств сервера. Все остальные БП переносятся на сервер.

Примеры реализации бизнес-правил приведены в методических указаниях к лабораторным работам [3].

В первую очередь бизнес - правила реализуют следующие ограничения:

- задание допустимого диапазона значений;
- задание значения по умолчанию;
- требование уникальности значения;
- запрет пустого значения;
- ограничение ссылочной целостности.

Бизнес - правила можно реализовать на физическом и программном уровне. В первом случае эти правила задаются при создании таблиц и входят в структуру БД.

На программном уровне бизнес - правила можно реализовать на сервере или в приложении. Для реализации бизнес - правил на сервере обычно используются триггеры или хранимые процедуры.

2.9 Computer Aided Software Engineering (CASE) технологии представляют собой совокупность методов проектирования программного обеспечения, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех стадиях разработки и сопровождения систем и разрабатывать приложения в соответствии с информационными потребностями пользователей.

Большинство существующих CASE-средств основано на методах структурного или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств.

На сегодняшний день практически все ведущие компании - разработчики технологий и программных продуктов (IBM Rational Software, Oracle, Borland, Computer Associates и др.) располагают развитыми CASE технологиями.

2.10 Методическую основу ТС ПО корпорации Oracle (www.oracle.com) составляет метод Oracle (Oracle Method) - комплекс методов, охватывающий большинство процессов жизненного цикла (ЖЦ) программного обеспечения.

В состав комплекса входят:

CDM (Custom Development Method) - разработка прикладного ПО;

PJM (Project Management Method) - управление проектом;

AIM (Application Implementation Method) - внедрение прикладного ПО;

BPR (Business Process Reengineering) - реинжиниринг бизнес-процессов;

OCM (Organizational Change Management) - управление изменениями.

По существу, CDM является методическим руководством по разработке прикладного ПО с использованием инструментального комплекса Oracle Developer Suite, а сам процесс проектирования и разработки тесно связан с Oracle Designer и Oracle Forms.

Комплекс Oracle Developer Suite содержит набор интегрированных средств разработки для быстрого создания приложений. Он включает средства моделирования, программирования на Java, разработки компонентов, бизнес - анализа и составления отчетов. Все эти средства используют общие ресурсы, что позволяет совместно работать над одним проектом группе разработчиков. Oracle Developer Suite интегрирован с Oracle Database и Oracle Application Server, образуя единую платформу для создания и установки приложений.

Oracle Developer Suite включает в себя:

- Oracle Designer - средство моделирования и генерации приложений;

- Oracle Forms - средство быстрой разработки приложений;

- Oracle Reports - визуальное средство разработки отчетов;

- Oracle JDeveloper - средство визуального программирования на языке Java;

- Oracle Discoverer - средство для разработки аналитических приложений;

- Oracle Warehouse Builder - система для построения хранилищ данных;

- Oracle Portal - средство разработки информационного портала организации.

2.11 Технология Rational Unified Process (IBM Rational Software) - одна из наиболее совершенных технологий, претендующих на роль мирового корпоративного стандарта - Rational Unified Process (RUP).

RUP представляет собой программный продукт, разработанный компанией Rational Software (www.rational.com), которая в настоящее время входит в состав IBM.

В состав Rational Suite, кроме самой технологии RUP как продукта, входят следующие компоненты:

- Rational Rose - средство визуального моделирования (анализа и проектирования), использующее язык UML;
- Rational XDE - средство анализа и проектирования, интегрируемое с платформами MS Visual Studio .NET и IBM WebSphere Studio Application Developer;
- Rational Requisite Pro - средство управления требованиями, предназначенное для организации совместной работы группы разработчиков;
- Rational Rapid Developer - средство быстрой разработки приложений на платформе Java 2 Enterprise Edition;
- Rational ClearCase - средство управления конфигурацией ПО;
- Rational SoDA - средство автоматической генерации проектной документации;
- Rational ClearQuest - средство для управления изменениями и отслеживания дефектов в проекте на основе средств e-mail и Web;
- Rational Quantify - средство количественного определения узких мест, влияющих на общую эффективность работы программы;
- Rational Purify - средство для локализации трудно обнаруживаемых ошибок времени выполнения программы;
- Rational PureCoverage - средство идентификации участков кода, пропущенных при тестировании;
- Rational TestManager - средство планирования функционального и нагрузочного тестирования;
- Rational Robot - средство записи и воспроизведения тестовых сценариев;
- Rational TestFactory - средство тестирования надежности;
- Rational Quality Architect - средство генерации кода для тестирования.

Одно из основных инструментальных средств комплекса Rational Rose представляет собой семейство объектно-ориентированных CASE-средств и предназначено для автоматизации процессов анализа и проектирования, а также для генерации кодов на различных языках и выпуска проектной документации. В основе работы Rational Rose лежит построение диаграмм и спецификаций UML, определяющих архитектуру системы, ее статические и динамические аспекты.

2.12 Компания Borland (www.borland.com) в результате развития собственных разработок и приобретения целого ряда компаний представила интегрированный комплекс инструментальных средств, реализующих управление полным жизненным циклом приложений (Application Life Cycle Management, ALM).

В соответствии с технологией Borland процесс создания ПО включает в себя пять основных этапов:

- определение требований;

- анализ и проектирование;
- разработка;
- тестирование и профилирование;
- развертывание.

2.13 Компания Computer Associates (www.ca.com) предлагает комплексы инструментальных средств поддержки различных процессов ЖЦ ПО:

1. AllFusion Modeling Suite - интегрированный комплекс CASE-средств, включающий следующие продукты:

- AllFusion Process Modeler (BPwin) - функциональное моделирование;
- AllFusion ERwin Data Modeler (ERwin) - моделирование данных;
- AllFusion Component Modeler (Paradigm Plus) - объектно-ориентированный анализ и проектирование с использованием UML и возможностью генерации кода;
- AllFusion Model Manager (Model Mart) - организация совместной работы команды разработчиков;
- AllFusion Data Model Validator (ERwin Examiner) - проверка структуры и качества моделей данных.

2. AllFusion Change Management Suite - комплекс средств управления конфигурацией и изменениями.

3. AllFusion Process Management Suite - средства управления процессами и проектами для различных типов приложений.

BPwin - средство моделирования бизнес-процессов, реализующее метод IDEF0, а также поддерживающее диаграммы потоков данных и IDEF3. В процессе моделирования BPwin позволяет переключиться с нотации IDEF0 на любой ветви модели на нотацию IDEF3 или DFD и создать смешанную модель. BPwin поддерживает функционально-стоимостной анализ (ABC).

2.14 Microsoft Office Visio помогает создавать имеющие профессиональный вид схемы, служащие пониманию, документированию и анализу сведений, данных, систем и процессов.

В комплекте поставки продукта в любой редакции (Standard, Professional) есть набор шаблонов моделей для наиболее распространенных нотаций, таких как диаграммы потоков данных, диаграммы цепочки добавленного качества, диаграммы типа Event-driven Process Chain, IDEF0, SwimLane, а также шаблоны для моделирования оргструктур компаний.

Технически Visio представляет собой настольное приложение, манипулирующее отдельными файлами (документами). Документ Visio включает одну или несколько диаграмм, расположенных на одной либо ряде страниц. Каждый документ содержит набор символов (соответствующих объектам моделей) и коннекторов (соответствующих связям), при этом у символов, помимо имен, могут быть дополнительные атрибуты, определяемые пользователем в процессе моделирования.

При необходимости набор символов, входящих в комплект поставки продукта, может быть расширен за счет символов, создаваемых пользователями. Глобальных ограничений на правила и возможности создания связей между определенными типами символов нет.

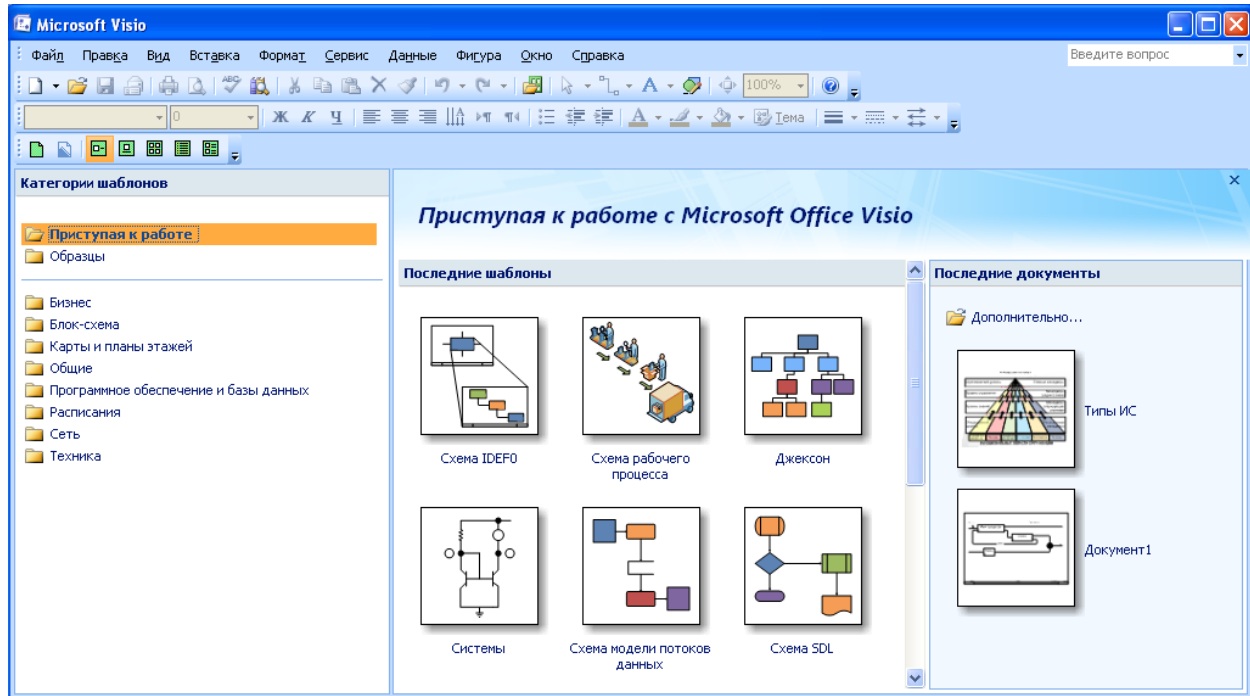


Рисунок 2.3 – Шаблоны диаграмм Microsoft Visio

В отличие от промышленных средств проектирования Visio не поддерживает функциональную связь между листами диаграмм, не осуществляет проверку целостности и полноты данных и связей.

Средства интеграции с приложениями Microsoft Office, позволяют осуществлять вставку диаграмм Office Visio 2007 в документы Microsoft Office в качестве иллюстраций, создание диаграмм Visio непосредственно в этих приложениях.

2.15 Методология функционального моделирования SADT (Structured Analysis and Design Technique) - одна из самых известных методологий анализа и проектирования систем, введенная в 1973 г. Дугласом Россом (Ross). На ее основе разработана, в частности, известная методология IDEF0 (Icam DEFinition).

Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Основные элементы этой методологии основываются на следующих концепциях:

- графическое представление блочного моделирования. Графика блоков и дуг SADT-диаграммы отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него. Взаимодействие блоков друг с другом описываются посредством интерфейсных дуг, выражающих "ограничения", которые в свою очередь определяют, когда и каким образом функции выполняются и управляются;

- строгость и точность. Выполнение правил SADT требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика.

Правила SADT включают:

- ограничение количества блоков на каждом уровне декомпозиции (правило 3-6 блоков);
- связность диаграмм (номера блоков);
- уникальность меток и наименований (отсутствие повторяющихся имен);
- синтаксические правила для графики (блоков и дуг);
- разделение входов и управлений (правило определения роли данных).
- отделение организации от функции, т.е. исключение влияния организационной структуры на функциональную модель.

2.16 Приведенные на рисунках примеры диаграмм, описаны в статье Максима Сычевского «Использование BPwin в консалтинговых проектах», КомпьютерПресс №1 за 2002 год.



Рисунок 2.4 – Диаграмма процесса верхнего уровня

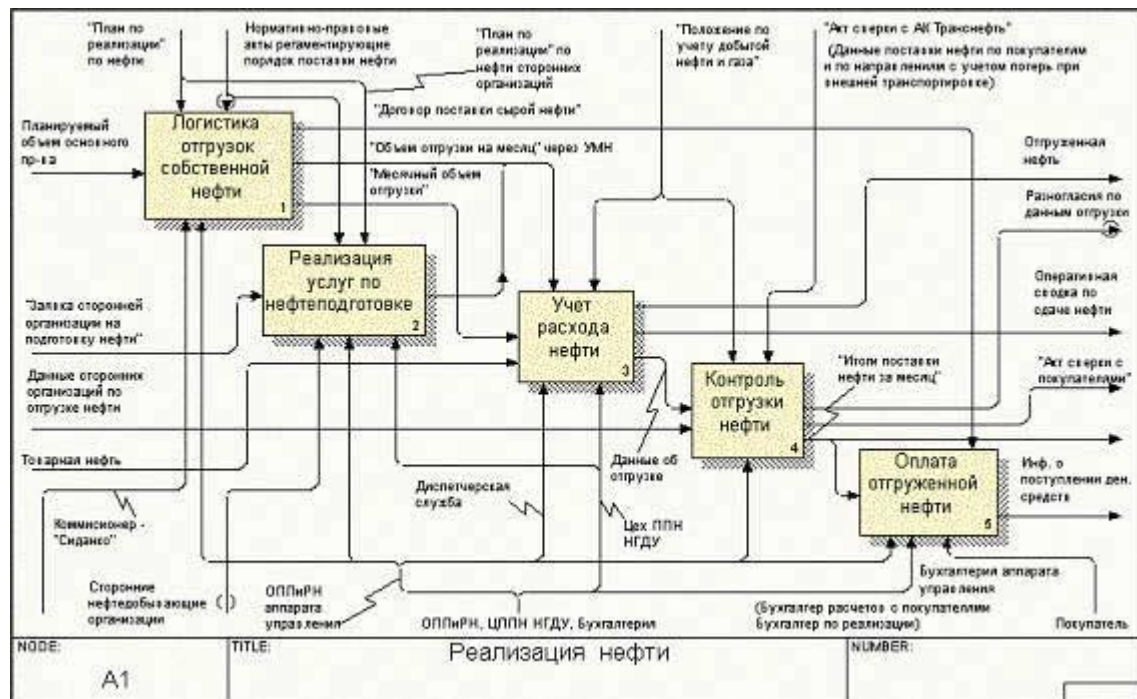


Рисунок 2.5 – Декомпозиция процесса

2.17 Для детализации нижнего уровня модели используются диаграммы потоков данных - DFD (Data Flow Diagrams).



Рисунок 2.6 – Пример диаграммы потоков данных

Описание функциональных блоков диаграмм потоков данных приведено в методических указаниях к лабораторному практикуму.

2.18 Для построения небольшой модели используются два, три уровня структурных диаграмм с последующим переходом в диаграммы потоков данных, расширяемых миниспецификациями.

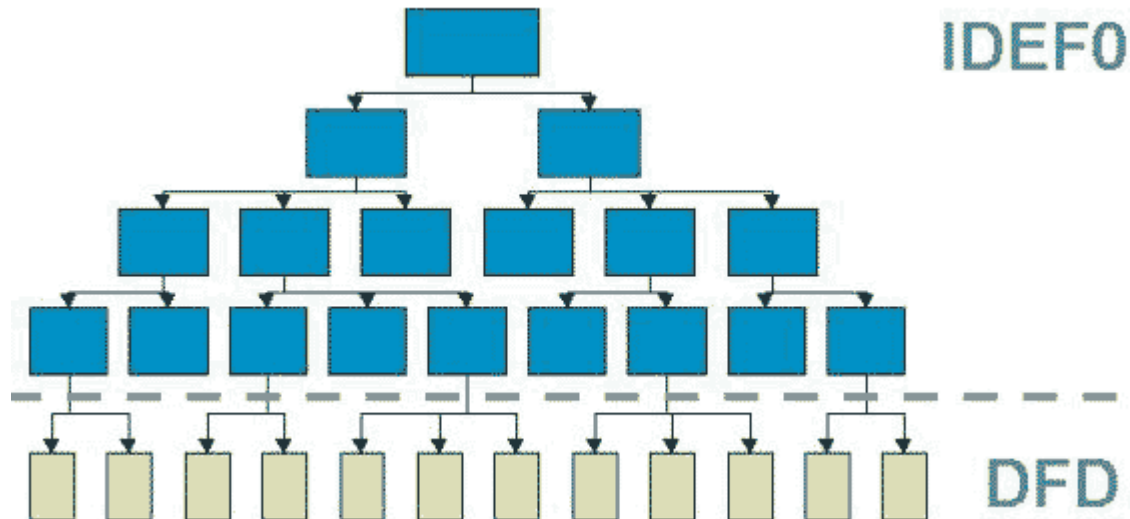


Рисунок 2.7 – Иерархическая структура

2.19 Сравнительный анализ SADT-моделей и потоковых моделей

Как уже отмечалось, практически во всех методах структурного анализа используются три группы средств моделирования:

- диаграммы, иллюстрирующие функции, которые система должна выполнять, и связи между этими функциями - для этой цели чаще всего используются DFD или SADT (IDEF0);
- диаграммы, моделирующие данные и их взаимосвязи Entity-Relationship Diagrams (ERD), диаграммы «сущность-связь»;
- диаграммы, моделирующие поведение системы - State Transition Diagrams (STD), диаграммы переходов состояний.

Таким образом, наиболее существенное различие между разновидностями структурного анализа заключается в методах и средствах функционального моделирования. С этой точки зрения все разновидности структурного системного анализа могут быть разбиты на две группы - применяющие методы и технологию DFD (в различных нотациях) и использующие SADT-методологию.

Соотношение применения этих двух разновидностей структурного анализа в существующих CASE-средствах составляет по материалам CASE Consulting Group 90% для DFD и 10% для SADT.

2.20 Демонстрация.

Работа со справочным материалом.

Работа учебного примера.

Подключение и регистрация сервера.

Работа в среде MS Visio 2007.

3 ТЕХНОЛОГИИ И МЕТОДЫ ПРОЕКТИРОВАНИЯ

Технология и методы проектирования.

Основные стадии и этапы технологической схемы проектирования ИС.

Декомпозиция, принципы иерархии.

Внешняя среда.

Методологические принципы проектирования ИС:

- концептуальное проектирование;
- логическое проектирование;
- физическое проектирование.

3.1 Методологические принципы проектирования ИС (11 номер КП за 2001г.).

Концептуальное моделирование - для определения направления развития предприятия.

Логическое моделирование - для описания деятельности предприятия CASE-средствами.

Физическое моделирование - для формализации деятельности предприятия средствами ERP-системы (то есть для создания нормативной модели предприятия).

3.2 Концептуальная модель является отраслевой моделью и, как правило, разрабатывается для предприятия внешним консультантом (обычно на основе эталонных моделей, предлагаемых поставщиками ERP-систем). В ней определяются основные направления развития предприятия через графическое представление передовой мировой практики (заключенной в стандарты ISO и ERP) и через определение несоответствий деятельности предприятия данной практике. Концептуальная модель подразумевает унификацию основных процессов предприятия в соответствии со стандартами ISO 9001:2000 и ERP.

Эталонная модель с использованием ISO 9000 переводится в IDEF0-модель, отображающую:

- декомпозицию процессов предприятия - верхний уровень иерархии процессов соответствует элементам и подэлементам стандарта ISO 9001:2000, а нижние уровни раскрываются с использованием ERP-стандарта;
- проектирование графического «скелета» документации системы менеджмента качества (СМК) предприятия;
- определение ключевых пользователей процессов и бизнес - функций;
- определение на базе ERP-системы основных модулей информационной системы предприятия, обеспечивающих выполнение процессов;
- определение связей процессов по входам/выходам.

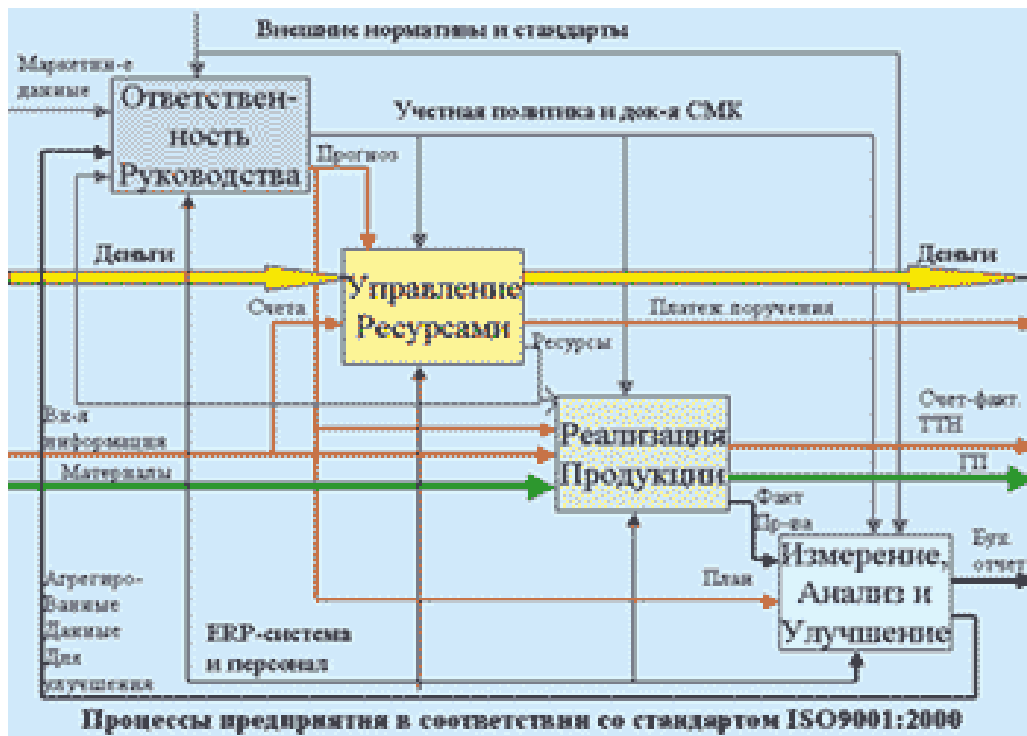


Рисунок 3.1 – Пример IDEF0-модели управления предприятием

3.3 Второй уровень бизнес - моделирования логический, необходим для уточнения основных выводов, следующих из концептуальной модели.

Цель логического моделирования - построить интегрированную модель деятельности предприятия, являющейся связующим звеном между бизнес - методиками и ERP-системой.

Логическая модель описывает деятельность предприятия, посредством объектно-ориентированного проектирования (опираясь на методологию бизнес - моделирования RUP и нотацию UML) или структурного проектирования.

Логическая модель позволяет спланировать, как нужно реорганизовать текущие способы выполнения процессов предприятия в желаемые - вплоть до каждого рабочего места.

Модель помогает детально ответить на следующие вопросы:

- кто и где исполняет бизнес - функции (организационный аспект деятельности);
- что перемещается в материальных и в связанных с ними информационных потоках (элементный аспект деятельности предприятия);
- как предприятие выполняет бизнес - функции (функциональный аспект);
- когда предприятие осуществляет бизнес - функции (динамический аспект);
- какие информационные технологии необходимы для поддержания бизнес - функций на предприятии.

3.4 Вопрос.

Зачем производится декомпозиция сложных функций?

3.5 Функциональная декомпозиция.

Для выполнения процесса декомпозиции сложной БФ используется структурный метод, в основе которого лежат три основных принципа:

1. Первым шагом упрощения сложной системы является ее разбиение на атомарные, с точки зрения пользователя, функции, при этом такое разбиение должно удовлетворять следующим критериям:

- каждая подсистема должна реализовывать единственную функцию системы, атомарную с точки зрения пользователя;
- функция каждой подсистемы должна быть легко понимаема независимо от сложности ее реализации;
- связь между подсистемами должна вводиться только при наличии связи между соответствующими функциями системы;
- связи между подсистемами (интерфейсы подсистем) должны быть простыми, насколько это возможно, для обеспечения независимости между ними.

2. Второй важной идеей, лежащей в основе структурных методов, является идея иерархии.

Для «понимаемости» сложной системы недостаточно разбиения ее на части, необходимо эти части организовать определенным образом, а именно в виде иерархических структур (рисунок 2.6).

Любая организация имеет директора, заместителей по направлениям, иерархию руководителей подразделений, рядовых служащих (организационно – штатная структура предприятия).

3. Последним моментом использования структурных методов является использование различных графических нотаций, диаграммы которых служат для облегчения понимания сложных систем. Известно, что “одна картинка стоит тысячи слов”.

Существует наиболее устоявшийся перечень атрибутов, которые модель бизнес-процессов должна описывать на изобразительном уровне, а именно:

- воздействия, инициирующие каждый шаг бизнес - процесса;
- исполнители каждого шага (это могут быть как люди, так и программы и механизмы);
- воздействия, регламентирующие данный шаг (законодательные акты, рыночные условия и т. п.);
- результат, получаемый на выходе конкретного шага бизнес - процесса.

3.6 Модель учебного примера, реализующая единственную функцию предприятия – установление контактов.

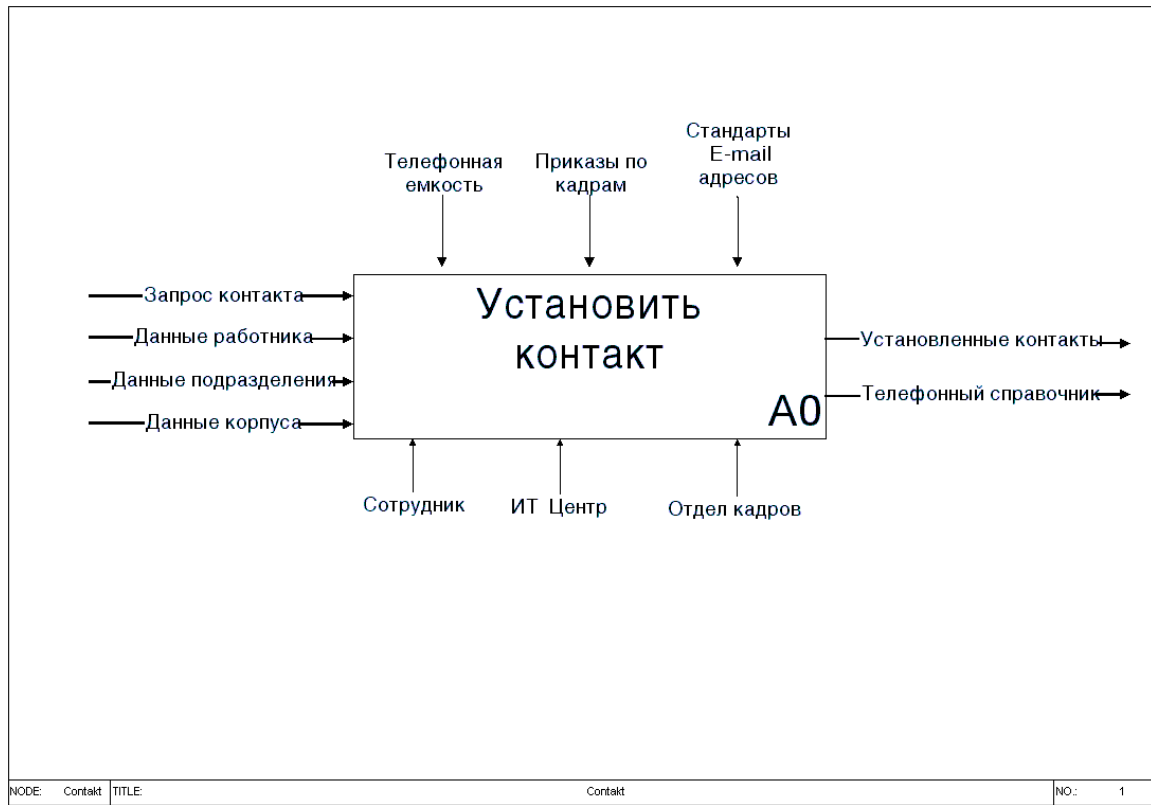


Рисунок 3.2 – Контекстная диаграмма процесса

3.7 При анализе задачи могут рассматриваться следующие аспекты:

- частота издания телефонного справочника предприятия;
- его объем и себестоимость;
- количество исправлений – изменений за определенный срок;
- количество устанавливаемых, например, ежедневно контактов;
- среднее время установления контакта (нормирование).

Влияние рассмотренных факторов на деятельность предприятия, его основные показатели.

Решение: разработать электронный справочник включающий:

- серверную БД;
- клиентскую часть, включая локальную БД;
- подсистему автоматического обновления;
- подсистему администрирования;
- подсистему подготовки печатного издания;
- модуль аналитических расчетов, по количеству обращений, установленных контактов, прогнозирование увеличения адресов электронной почты и списков рассылок;
- web – интерфейс доступа.

3.8 Модели процессов проектирования.

Проектировщик ИС (программист) не является автором концептуальной, логической и физической модели будущей системы. Как руководитель диссертации сделает Вам постановку задачи, такое качество продукта и получит.

Концептуальная модель строится бизнес – аналитиками (внешний консалтинг) с участием руководства предприятия.

Логическая модель - привлечение специалистов знакомых с методологиями моделирования, с работой CASE – средств (объектное, структурное).

Физическое моделирование, системотехники и администраторы, учет особенностей архитектуры ИС, существующей инфраструктуры (архитектура и топология подразделений и телекоммуникаций) предприятия, используемое программное и аппаратное обеспечение (SW + HW), наличие персонала.

Участие системотехников, программистов на всех этапах анализа, залог успешной реализации проекта информатизации бизнес – процессов.

Решение на реализацию того или иного проекта должно инициироваться самым верхним уровнем управления предприятием.

Любой проект, даже не большой по объему и стоимости, реализуется большой группой участников (руководители разного звена, снабженцы, системные и сетевые администраторы и т.д.).

3.9 Основные этапы.

Традиционно выделяются следующие основные этапы жизненного цикла (ЖЦ) программного обеспечения:

- анализ требований,
- проектирование,
- кодирование (программирование),
- тестирование и отладка,
- эксплуатация и сопровождение.

3.10 Классические модели проектирования информационных систем:

1. Каскадная модель. Переход на следующий этап означает полное завершение работ на предыдущем этапе.

2. Поэтапная модель «Водопад», с промежуточным контролем. Разработка системы ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют уменьшить трудоемкость процесса разработки по сравнению с каскадной моделью. Время жизни каждого из этапов может растягиваться на весь период разработки.

3. Спиральная модель. Особое внимание уделяется начальным этапам разработки - выработке стратегии, анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования). Каждый

виток спирали предполагает создание некой версии продукта или какого-либо его компонента, при этом уточняются характеристики и цели проекта, определяется его качество, и планируются работы следующего витка спирали.

3.11 Перечень основных стандартов системной инженерии.

SPC (Software Productivity Consortium) выделяет тот минимум стандартов на процессы проектирования, который рекомендуется взять за основу. В их число включены ISO/IEC 12207, ISO/IEC 15288 CD2, ISO 15504 (SPICE), EIA/ANSI 632, EIA/IS 731 (SECM), TickIT.

Назначение следующих нормативных документов (НД):

- ISO/IEC 12207, Information technology - Software life cycle processes. 1995.

- ISO/IEC TR 15271, Information technology - Guide for ISO/IEC 12207. 1998. (Стандарт ISO/IEC 12207 оказал революционизирующее влияние на многие другие НД, в том числе на стандарты моделей системного проектирования: процессы жизненного цикла систем, модель зрелости процессов.)

- EIA/ANSI 632, Processes for Engineering a System. 1999. (Этот стандарт не только заменил ряд популярных более старых американских стандартов, но был использован как вклад американской группы в создание ISO/IEC 15288.)

- EIA/IS-731, System Engineering Capability Model (SECM). 1999. Part 1, SECM Model. Part 2, SECM Appraisal Method. (В области стандартов на уровни зрелости процессов аналогично тому, как модель SW CMM переросла в модель и стандарт SPICE, модель SE CMM переросла в модель и стандарт SECM.)

- ISO/IEC 15288 CD2, Life Cycle Management - System Life Cycle Processes. 2000.

3.12 Для обеспечения преемственности полезно добавить в эту группу стандарты ГОСТ 34 (не гармонизированные с новыми, но применимые и полезные из-за совместимости по многим базовым понятиям, по сути многих работ, по опыту применения и др.).

Существенно, что два «потока» стандартов - на SE (system engineering) и на SW (software engineering), развивавшихся параллельно, четко стыкованы посредством указанных документов.

3.13 Организация проектной деятельности регламентируется стандартами уровня предприятия и проекта. В стандартах явно предусмотрены работы по постановке проектной деятельности и управлению ею. В ISO/IEC 15288 такими являются:

1. Процесс «Управление предприятием» (внедрение стандарта на основе стратегии предприятия);

2. Процесс «Управление процессами ЖЦС»;
3. Процесс «Управление ресурсами для 1 и 2».

3.14 Для процесса «Управление предприятием» предусмотрены следующие цель и результаты.

Цель процесса: определить, документировать и поддерживать правила и процедуры, относящиеся к организации ведения дел на предприятии в части, имеющей отношение к данному стандарту.

Результаты процесса:

1. Стратегические и тактические планы и цели, которые определяют формирование правил и процедур для внедрения требований данного Международного Стандарта;
2. Правила и процедуры для «управления ЖЦС», включая управление качеством, гарантии и контроль, в соответствии с ISO 9001;
3. Роли, ответственность и права (власть), способствующие эффективному управлению ЖЦС.

3.15 Задачи стандартов определить процессы, которые должны в практике работы предприятия отвечать, в частности, на такие вопросы:

- откуда берутся (должны браться) проекты и проектные программы?
- как получить стандарты, регламенты, инструкции, которые рационально использовать именно на данном предприятии (стандарты предприятия)?
- как управлять процессами проектирования на предприятии?
- как получить набор стандартов конкретного проекта?
- как обеспечить стыковку различных проектов?
- что является определяющим критерием для проверки правильности выполнения проекта?

Узнать про используемые стандарты предприятия, можно, поинтересовавшись о последних результатах внедрения ИС.

3.16 Заключение.

В первом модуле рассмотрели:

1. модель бизнеса (предприятия), с использованием математических моделей ...;
2. модель проектирования, как процесса;
3. модели процессов, например, линейное планирование, ...;
4. модели данных (иерархическая, файловая, реляционная, объектная, объектно-реляционная, ...);
5. Архитектура системы ...

3.17 Вопросы по 1 лабораторной работе.

Определение и основные свойства бизнес - функции.

Методологии проектирования информационных систем.

Определение бизнес – процесса, примеры бизнес - процессов.

Принципы декомпозиции сложных систем.

Что является источником бизнес – процессов?

Особенности реализации ограничений в ИС с различной архитектурой.

Типы диаграмм используемых в различных нотациях.

3.18 Задания СРМП.

1. Входной контроль по дисциплинам;
2. Защита модели выбранного бизнес – процесса;
3. Защита разработанных бизнес – правил;
4. Ответить на контрольные вопросы первого модуля;
5. Защитить отчет по первой лабораторной работе или защита раздела магистерской диссертации;
6. Разработать пример вопроса тестового задания по теме раздела.

3.19 Задания СРМ.

1. Изучить методические указания к первой лабораторной работе;
2. Ответить на примеры тестовых заданий;
3. Выбор предметной области для выполнения лабораторных работ;
4. Установка на персональном компьютере серверного программного обеспечения;
5. Проверка работоспособности среды управления сервером.

3.20 Демонстрация.

Построение диаграмм в среде MS Visio 2007.

Анализ основных ошибок описания процессов.

4 СОВРЕМЕННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

Примеры внедрения систем.

Средства разработки приложений.

Программные технологии в Delphi 7.

Компоненты для разработки Web – приложений.

4.1 Рассмотрим в качестве примера информационно – телекоммуникационную структуру образовательной среды ВКГТУ.

Ниже приведен пример схемы, построенной в среде MS «Visio2002».

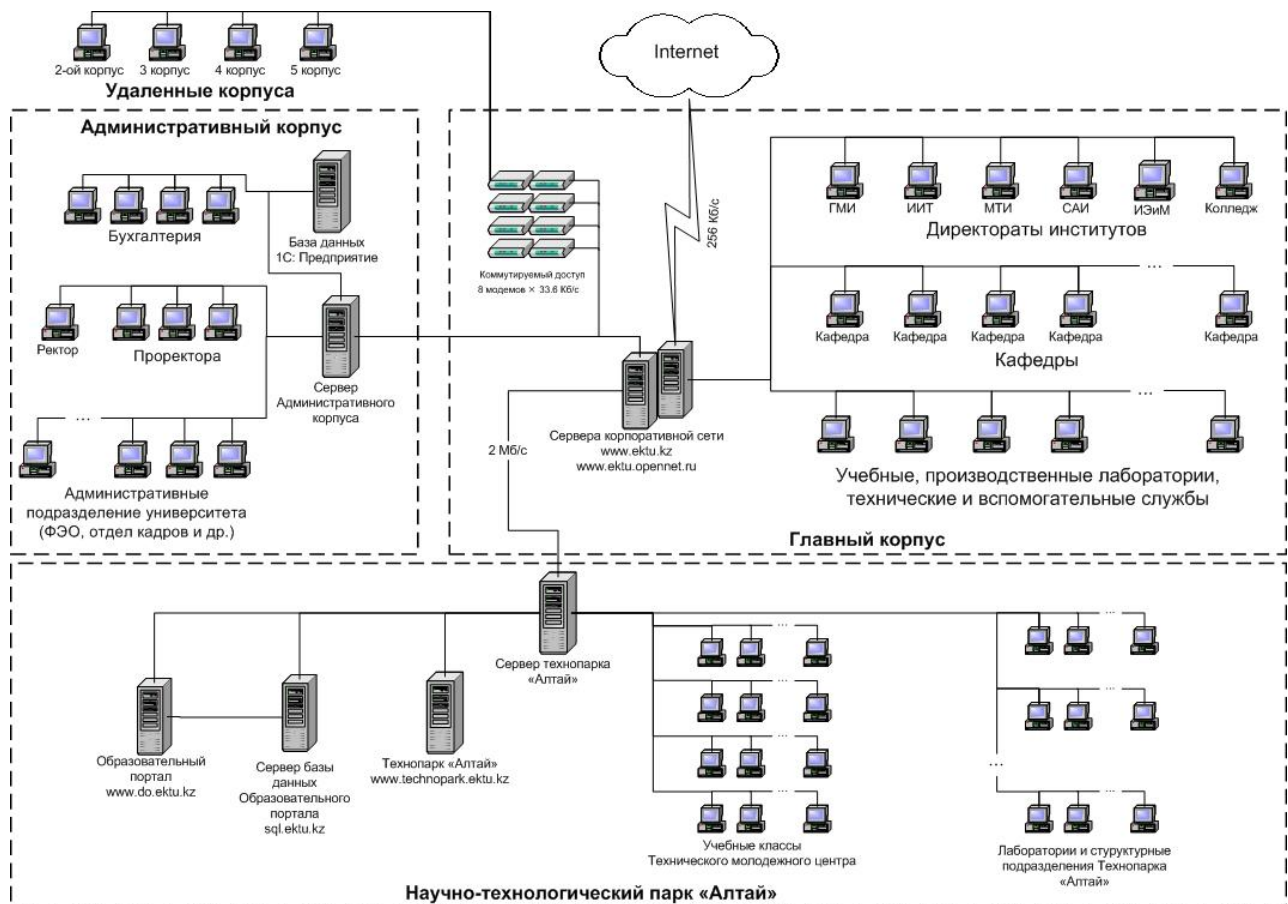


Рисунок 4.1 – Информационно – телекоммуникационная структура ВКГТУ

При дальнейшей детализации схема может отображать сетевую топологию и инфраструктуру серверных приложений (контроллер домена, службу разрешения имен, службу управления адресами).

4.2 На всех рабочих станциях имеется возможность реализации любой пользовательской функций электронной среды.

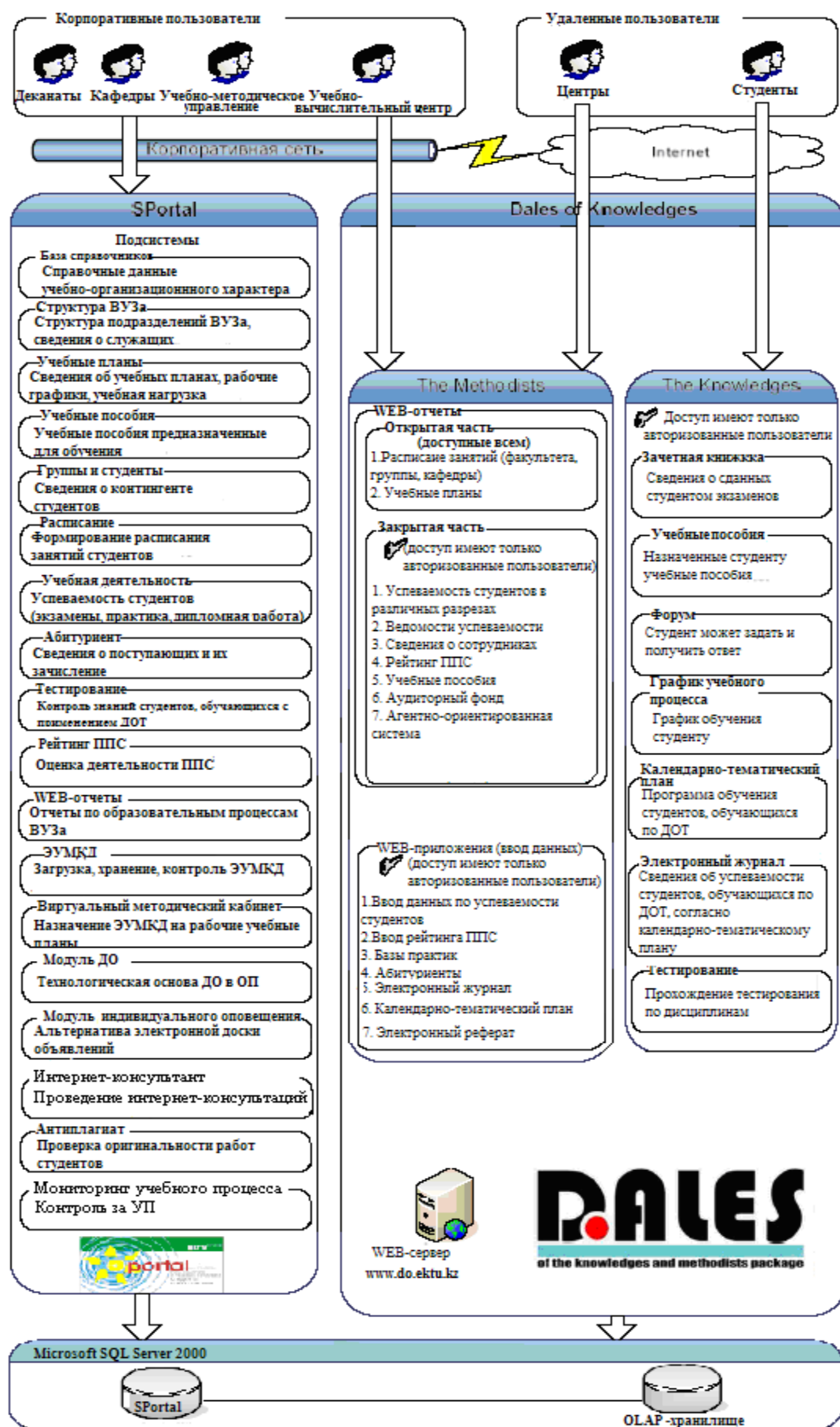


Рисунок 4.2 – Структура образовательной IT-системы ВКГТУ

В составе сети введена в действие система оперативного обмена информацией (сообщения, файлы, голосовой обмен), что позволило внедрить наиболее востребованные процедуры электронного документооборота.

Образовательное пространство современного высшего учебного заведения должно формироваться как открытая система с интеллектуально развитым интерфейсом взаимодействия всех участников довузовского, вузовского и послевузовского образовательного процесса.

В состав информационной системы вуза должны входить следующие подсистемы:

4.3 Подсистемы управления учебным процессом:

- «Приемная комиссия»;
- «Электронный деканат»;
- «Электронная кафедра»;
- «Расписание»;
- «Успеваемость»;
- «Тестирование».

4.4 Подсистемы дистанционного образования:

- «Управление контентом дистанционного образования»;
- «Управлением контингентом студентов и учебными достижениями»;
- «Организация контроля знаний».

4.5 Подсистемы управления персоналом:

- «Отдел кадров»;
- «Штатное расписание»;
- «Воинский учет».

4.6 Подсистемы управления финансово-хозяйственными процессами:

- «Бухгалтерский учет»;
- «Финансово-экономический анализ и прогнозирование»;
- «Учет материально-технических средств»;
- «Расчет заработной платы».

4.7 Дополнительные:

- «Подсистема электронного документооборота».
- «Подсистема управления научно-инновационной деятельностью».
- «Подсистема управления библиотекой».

4.8 Программные технологии для разработки Web – приложений:

CGI (Common Gateway Interface);
 PHP (Personal Home Pages);
 Perl;

ASP (Active Server Page), продолжение ASP.NET;
 JSP (JavaServer Pages);
 JavaScript;
 XML (Extensible Markup Language);
 CSS (Cascading Style Sheets).

4.9 Обзор средств разработки приложений приведен по статье в 6 номере журнала «КомпьютерПресс» за 2009 год.

4.10 Eclipse — это не только инструмент, но и, что более важно, набор технологий, изменивший рынок средств разработки приложений. Технология Eclipse сегодня используется во многих известных средствах разработки.

Развитием Eclipse управляет некоммерческая организация Eclipse Foundation. Ее целью является создание бесплатной платформы, которую производители ПО могли бы использовать для разработки коммерческих продуктов.

4.11 IBM's Rational Application Developer (RAD) for Websphere — это среда разработки и набор инструментов, основанные на платформе Eclipse и оптимизированные для создания приложений, выполняемых под управлением сервера приложений IBM WebSphere.

С помощью RAD можно создавать приложения для платформ J2SE, J2EE, порталов, Web-приложения, Web-службы, SOA-приложения, используя встроенные визуальные средства быстрого проектирования, разработки, тестирования и развертывания.

Последняя версия продукта — RAD v7.5 — позволяет создавать приложения, использующие стандарты Java EE5 (EJB 3.0, JPA, JSF 1.2, JAX-WS 2.0, JSP 2.1, Servlet 2.5), поддерживает генерацию кода JavaScript, создание AJAX-приложений, а также позволяет применять адаптеры WebSphere для бизнес-приложений компаний SAP и Oracle.

RAD является составной частью портфеля продуктов семейства IBM Rational для поддержки всего жизненного цикла разработки приложений и отлично интегрирован с такими инструментами, как средство проектирования приложений Rational Software Architect.

Стоимость RAD довольно высока — она составляет более 4 тыс. долл. за одну лицензию с годовой технической поддержкой.

4.12 MyEclipse — это основанная на платформе Eclipse и ее расширениях среда разработки приложений, которая создана компанией Genuitec, специализирующейся на обслуживании компаний — разработчиков приложений. Сегодня набор расширений, входящих в состав MyEclipse, включает средства поддержки J2SE, J2EE, баз данных, UML, технологии Java Server Faces, AJAX и генерации отчетов.

Среда MyEclipse в первую очередь предназначена для разработчиков крупных компаний, в основном использующих технологии J2EE и AJAX. Помимо этого существует редакция MyEclipse Blue Edition, оптимизированная для создания приложений, выполняемых в среде сервера приложений IBM WebSphere.

Стоимость MyEclipse невысока — в зависимости от редакции она составляет от 30 до 150 долл. за годовую подписку.

4.13 Средство разработки Delphi, созданное компанией Borland и принадлежащее сегодня компании Embarcadero, было создано еще в середине 90-х годов, когда средства визуальной разработки приложений только начали появляться. Создавая Delphi, компания Borland сумела объединить удобство и простоту визуального проектирования пользовательского интерфейса с объектно-ориентированным языком программирования и высокопроизводительным компилятором.

Библиотека повторно используемых компонентов Visual Component Library (VCL), входящая в комплект поставки продукта и используемая при создании приложений, может быть пополнена компонентами сторонних разработчиков. Последняя версия Delphi включает средства поддержки нескольких языков программирования (Delphi, C, C++) для платформ Windows, .NET и Mono, средства моделирования приложений, поддерживает ряд современных технологий разработки веб-приложений, таких как AJAX.

Стоимость Delphi в зависимости от редакции составляет от 900 до 2000 долл.

4.14 JDeveloper, являющийся бесплатным инструментом, появился более десяти лет назад, когда компания Oracle лицензировала у Borland компилятор Java. В 2001 году этот продукт был полностью переписан и на данный момент представляет собой не только средство разработки, но и платформу, используемую во всех продуктах Oracle, как в бизнес-приложениях, так и в инфраструктурных продуктах, и включающую средства администрирования баз данных, инструменты для поддержки SOA-приложений и бизнес-процессов. JDeveloper также поставляется совместно с платформой Oracle Application Framework.

JDeveloper в первую очередь предназначен для разработчиков, создающих приложения для крупных компаний и участвующих в проектах внедрения бизнес-приложений, принадлежащих корпорации Oracle (EBusiness Suite, Siebel, Peoplesoft).

4.15 Adobe Creative Suite — наследник известного набора инструментов Macromedia Studio, состоящего из Dreamweaver 8, Flash 8, Flash 8 Video Converter, Fireworks 8, Contribute 3 и FlashPaper.

Adobe Creative Suite происходит от дизайнерских инструментов и предназначен для создания привлекательных динамических веб-сайтов,

поэтому включает инструменты для создания графики, анимации, обработки видео- и аудиоданных. Именно поэтому разработчики считают Adobe Creative Suite (а точнее, средства программирования, включенные в соответствующие инструменты) наилучшим средством для создания веб-приложений.

4.16 Microsoft Visual Studio представляет собой среду разработки, ориентированную на платформу Microsoft .NET и включающую инструменты Visual C++, Visual Basic, Visual C# и Visual J#. С помощью Visual Studio можно создавать широкий спектр Windows- и Web-приложений, Web-служб и приложений для мобильных устройств.

Последняя версия продукта, Visual Studio 2008, поддерживает несколько версий .NET Framework, разработку AJAX-приложений, язык LINQ (language integrated query) для упрощения создания приложений с базами данных.

Visual Studio предназначена для различных категорий разработчиков — от любителей до авторов корпоративных бизнес-приложений. Стоимость различных редакций этого продукта варьируется от 300 до 800 долл.

4.17 Средство разработки Java-приложений NetBeans было создано в середине 90-х годов и приобретено в 1999 году компанией Sun Microsystems, которая в 2000-м перевела этот продукт в категорию ПО с открытым кодом.

Сегодня среда NetBeans представляет собой платформу с открытым кодом, поддерживающую все типы Java-приложений (J2SE, J2EE, мобильные и веб-приложения) и включающую средства контроля версий и рефакторинга, а также средства поддержки языков, отличных от Java, таких как Ruby, JRuby, JavaScript, PHP, и технологии AJAX.

NetBeans доступен для свободной загрузки на соответствующем сайте и поддерживается сообществом разработчиков. Однако доступна и платная техническая поддержка продукта от компании Sun Microsystems.

4.18 Sun Studio — это среда разработки компании Sun для языков C, C++ и Fortran, основанная на платформе NetBeans, поддерживающая платформы Solaris, OpenSolaris и Linux и рассчитанная на оптимизацию производительности обработки данных с помощью многоядерных процессоров Sun SPARC, а также двух- и четырехъядерных процессоров Intel и AMD.

Среда Sun Studio доступна для свободной загрузки, но не является продуктом с открытым кодом.

Sun Studio включает полный набор инструментов, требующихся для разработки приложений: среду разработки, оптимизирующие компиляторы с автоматическим распараллеливанием вычислений, высокопроизводительные библиотеки.

4.19 Все технологии реализуют стандарты Web – интерфейса (использование службы WWW для получения сервисных услуг), определяющего взаимодействие браузера, Web – сервера и внешних модулей.

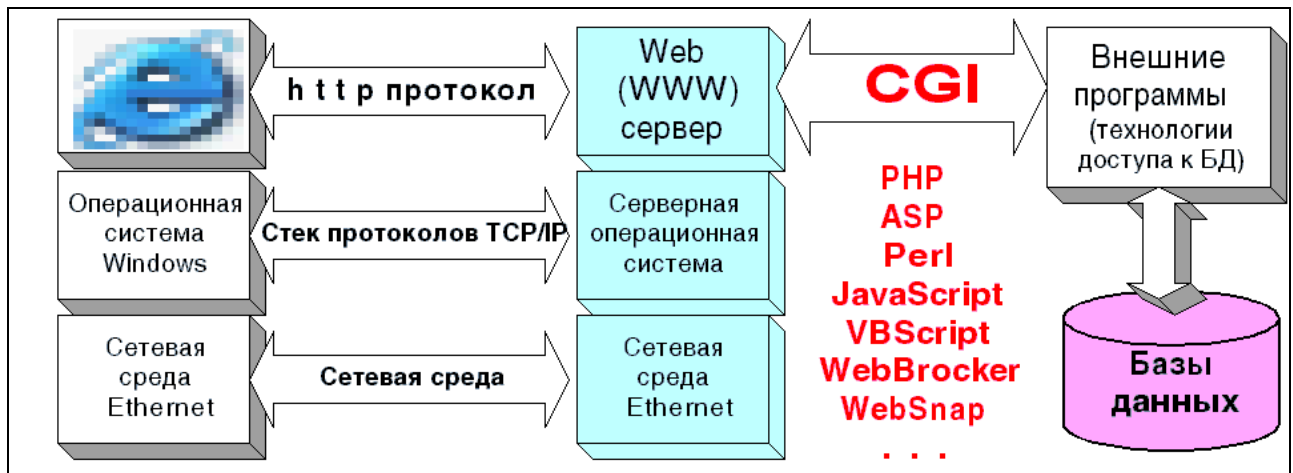


Рисунок 4.3 – Структура Web - приложения

4.20 CGI (Common Gateway Interface).

Спецификация CGI описывает формат и правила обмена данными между ПО Web - сервера и внешней запускаемой программой.

4.21 Технологии разработки Web – приложений в Delphi 7.

CGI – консольное приложение;

WebBroker;

WebSnap;

IntraWeb.

4.22 Демонстрация.

Проектирование CGI – скрипта.

Работа учебных скриптов.

Компоненты технологии WebBroker.

Функции управления.

5 КОМПОНЕНТЫ ИНФРАСТРУКТУРЫ ИС

Инфраструктура ИС, ее компоненты.

Базовые информационные системы.

Архитектура и структура программных технологий.

Кодирование и декодирование информации.

Переход от анализа и моделирования, к кодированию (программированию).

5.1 Информационная архитектура/инфраструктура (Information Architecture) форма, которую информационные технологии принимают в отдельной фирме, направленная на достижение поставленных этой фирмой целей и выполнения специфических функций [1].

Для анализа связности компонентов информационной инфраструктуры используются описательные (дескриптивные), коррелятивные и казуальные методы, «Успехи современного естествознания» №3 2010 Анализ инфраструктуры информационной системы Цветков В.Я., Булгаков С.В. Москва, Россия.

5.2 На рисунке 2.1 рассмотрена информационная инфраструктура организации, включающая ИТ - инфраструктуру.

5.3 ИТ - инфраструктура ИС.

1. Аппаратное обеспечение:

- сервер (ы);
- клиентские станции;
- коммуникационные средства;
- офисное оборудование.

2. Программное обеспечение:

- серверное ПО (операционные системы, системы администрирования, системы мониторинга, например, сетевой монитор или монитор транзакций;
- СУБД (Inter Base, Microsoft SQL, MySQL, Oracle);
- прикладное ПО (Enterprise Resource Planning (EPR) - Система управления ресурсами предприятия, системы управления активами и фондами - Enterprise Asset Management (EAM), системы управления взаимоотношениями с клиентами - Customer Relationship Management (CRM), системы управления цепочками поставок - Supply Chain Management (SCM);

- прикладные системы (офисные, графические, прочие);
- разработанное пользователем.

3. Специализированные средства:

- среды моделирования;

- среды разработки;
- средства управления и мониторинга.

4. ИТ – технологии (сетевые, данных):

- стандарты;
- протоколы;
- спецификации;
- языки.

5.4 Примеры прикладного программного обеспечения:

R/3(SAP), Triton (Baan), Oracle Application;

«Галактика»;

«Scala»;

«1С Предприятие»;

«КОМПАС».

5.5 Базовые компоненты системы управления предприятием ERP – стандарта:

- рынок и маркетинг;
- нормативное управление (отчетность);
- финансовое планирование;
- оперативное планирование;
- склад (хранение и перемещение);
- управление ресурсами (персоналом);
- контроль качества;
- сбыт (продажи);
- снабжение (закупки);
- бухучет;
- финансовый учет;
- доставка;
- гарантийное обслуживание;
- техобслуживание и ремонт;
- управление технологическими процессами (АСУ ТП).

5.6 Интерфейсы систем – межкомпонентное взаимодействие.

В общем случае интерфейс (interface) – общая граница между двумя функциональными объектами (компонентами или уровнями) системы.

Стандарт определяет службы в терминах функциональных характеристик и поведения, наблюдаемого на интерфейсе.

Стандарт является договором в том смысле, что он документирует взаимное обязательство между пользователем и поставщиком служб, а также гарантирует стабильное, документально оформленное, четкое выполнение данного обязательства. (ISO/IEC 14252).

5.7 Планирование архитектуры системы. В качестве упражнения построить из перечисленных компонентов модель архитектуры.

1. Уровни сетевой модели OSI (прикладной, представительный или сессий, транспортный, сетевой, передачи данных, физический);

2. Представление и хранение данных (языки программирования, языки построения запросов, средства доступа, файловая система, организация памяти);

3. Обработка данных (утилиты и библиотеки, прикладные программы, функции операционной системы, ядро операционной системы, системы команд процессора, системные шины);

4. Интерфейсы (генераторы форм и отчетов, программные оболочки, технологии оконного интерфейса, драйверы, периферийные устройства).

5.8 Web - сервера.

- Apache от ASF (Apache Software Foundation);

- Internet Information Server (IIS) от Microsoft;

- iPlanet server (бывший Netscape Enterprise server) от Netscape и Sun.

5.9 Среда разработки Web – приложения.

Web-сервер SmallHTTPServer. Каталог размещения по умолчанию - c:/shhttps/.

Каталог базы данных Web-сервера - c:/shhttps/www/.

Каталог расположения CGI-скриптов - c:/shhttps/www/cgi-bin/, создается самостоятельно.

Каталог для размещения собственных проектов c:/shhttps/project/, создается самостоятельно.

Запуск Web-сервера в работу производится вручную, помещение в «автозапуск» не рекомендуется.

После занятия сервер необходимо выгрузить. Клик правой кнопкой мыши на иконке сервера в панели задач с выбором функции “Exit”.

Операционная и программная среда разработки Delphi для создания приложений WebBroker, WebSnap и IntraWeb.

5.10 Пример.

Кто посмотрел Web-интерфейс управления Shttp-сервером?

Демонстрация примера управления.

5.11 Сравнение архитектур технологий WebBroker и WebSnap, демонстрирующее направления развития, приведено в таблице.

Таблица 5.1 Основные отличия архитектур.

Web Broker	WebSnap
Поддерживается для обратной совместимости	Несмотря на то, что поддерживаются все компоненты архитектуры WebBroker, осуществляющие генерацию HTML-документов, компоненты WebModule и WebDispatcher и т.д. имеют новую реализацию
Приложение может содержать только один WebModule	Приложение может быть разделено на блоки несколькими контейнерами WebModule, что позволяет вести параллельную разработку без возрастания числа конфликтов
Приложение может содержать только один компонент WebDispatcher	Может существовать несколько компонентов-диспетчеров, обрабатывающих различные типы запросов
Нет поддержки языка сценариев для серверной части приложения	Поддержка сценариев на JScript или VBscript, позволяющая отделить логику генерации HTML от бизнес - логики
Нет поддержки контекста (сессии)	Поддержка сессий позволяет хранить информацию о конечном пользователе, которая может быть полезна для реализации таких возможностей, как авторизация и отключение пользователей
Каждый запрос должен быть явно обработан компонентом WebActionItem	Компоненты-диспетчеры автоматически отвечают на различные запросы в соответствии с заложенной бизнес - логикой
Только некоторые компоненты позволяют осуществлять предварительный просмотр генерируемых страниц, оставляя большую часть разработки невизуальной	Дизайнер WebSnap позволяет создавать Web-страницы визуально и просматривать результаты во время разработки. Предварительный просмотр доступен для всех компонентов

5.12 Технология разработки Web-интерфейса.

Как правило, набор библиотек интерпретаторов, имеющих достаточно сложную структуру, ориентированных на конкретную аппаратную платформу сервера и программное обеспечение Web-сервера.

Если вы разработаете скрипт PHP и попытаетесь запустить его на сервере Apache, не установив сам пакет интерпретатора скриптов и не указав в настройке сервера путь к нему, то ни какого взаимодействия со скриптом не получится.

Без внешнего модуля интерпретации может работать только классический CGI – скрипт, включающий в себе все функциональные библиотеки.

Обязательно помнить о соответствии версий программных продуктов и поддерживаемых кодировках.

5.13 Технологии доступа к базам данных, реализуемые средой Delphi.

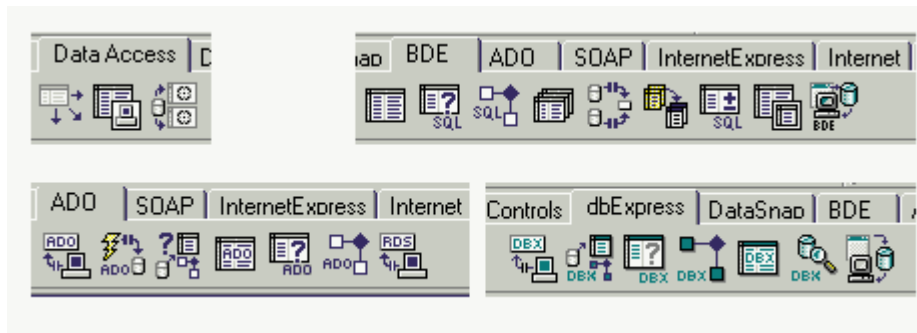


Рисунок 5.1 – Меню компонентов технологий доступа

5.14 Компоненты технологии – WebBroker (вкладка Internet). (Классы с приставкой T).

WebModule
WebDispatcher
WebActionItem
WebRequest
WebResponse
PageProducer
TableProducer
QueryTableProducer

5.15 Реализация Web – модуля, выбор из репозитория проектов, приводит к автоматическому созданию экземпляров классов;

TWebModule;
TWebDispatcher;
TWebRequest;
TWebResponse.

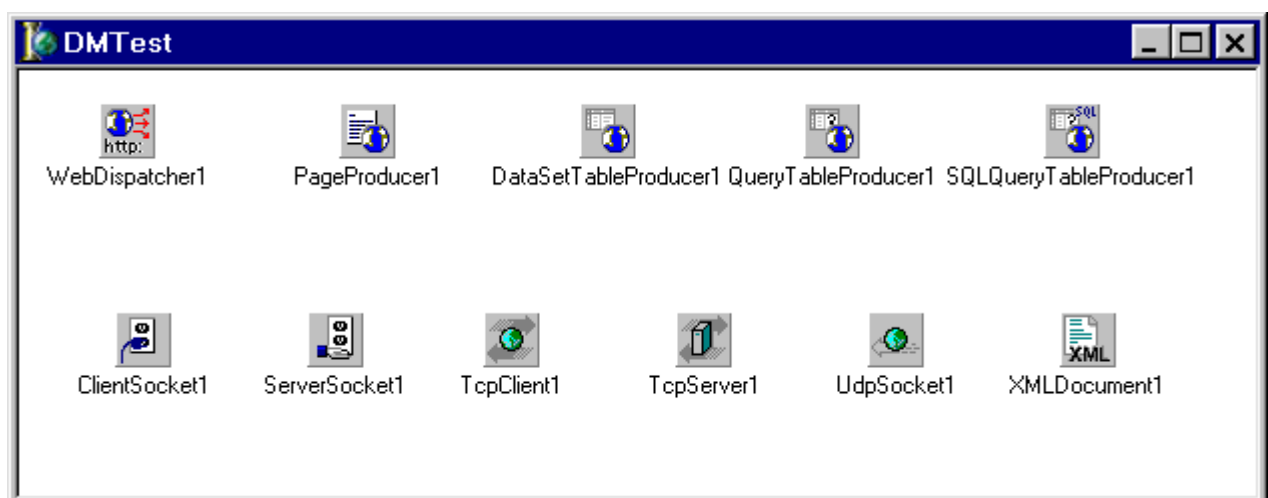


Рисунок 5.2 – Окно модуля с пиктограммами компонентов

Среда Delphi имеет открытую структуру кода, которую Вы всегда можете изучать в качестве пособия.

5.16 Основные составляющие технологий.

Первый фундаментальный механизм. Каждый Web – модуль должен содержать набор действий, определяющих варианты поведения модуля при получении того или иного запроса.

Второй – использование шаблонов (прозрачные теги).

Третий – использование JavaScript для интерпретации действий пользователя на стороне клиента.

Четвертый – появление серверных скриптов, интерпретируемых специальными модулями.

5.17 Типы данных.

При передаче данных серверному модулю происходит их кодирование:

- символы основного набора латиницы не кодируются;
- русские и т.п. символы заменяются шестнадцатеричным представлением вида %FF.

5.18 Демонстрация.

Создание модуля.

Проектирование действия.

Вызов модуля.

Пример тестового скрипта.

6 КОМПЛЕКС ТЕХНОЛОГИЙ

Комплекс технологий проектирования ИС.

Основные функции администрирования.

Методы получения данных из клиентского приложения.

Тренировочный тест к рубежному контролю.

6.1 Технологии для разработки Web – приложений:

CGI;

JavaScript;

ASP (Active Server Page), ASP.NET;

PHP;

Perl;

Редакторы гипертекста и скриптов;

Среды проектирования Web – приложений.

Современные среды проектирования (Delphi, Microsoft Visual Studio) включают: средства редактирования гипертекста, проектирования скриптов (JavaScript), поддержку серверных скриптов, средства предварительного просмотра страниц через браузер.

6.2 Основой всех современных технологий является интерпретация серверного модуля.

Для увеличения производительности Web - приложений, которые выполняются в отдельном адресном пространстве (CGI), создаются приложения в виде библиотек, загружаемых в адресное пространство Web - сервера и при необходимости остающихся там для обработки последующих запросов.

Подобные приложения для Microsoft Internet Information Service носят название ISAPI (Internet Server Application Program Interface).

А такие библиотеки для Web - сервера Apache называются Apache DSO (Dynamic Shared Objects).

6.3 Очередным шагом в развитии технологий создания Web - приложений стало появление средств, позволяющих отделить задачи Web - дизайна от задач, связанных с реализацией функциональности приложений.

Основная идея заключается в создании Web - страниц с внедренными в них фрагментами кода на скриптовых языках.

Внедренный фрагмент кода замещается результатом его выполнения, а полученная таким образом динамическая страница передается в пользовательский браузер.

6.4 Одной из наиболее популярных сегодня технологий, реализующих идею создания Web - страниц с фрагментами кода, является ASP.NET - ключевая в архитектуре Microsoft.NET Framework.

Основное отличие этой технологии от ASP в плане архитектуры приложений заключается в том, что код, присутствующий на Web - странице, не интерпретируется, а компилируется и кэшируется, что способствует повышению производительности приложений.

ASP.NET позволяет создавать так называемые серверные компоненты, возвращающие в браузер HTML-код с фрагментами кода на языке JavaScript способные предоставить более удобный пользовательский интерфейс, нежели статические страницы.

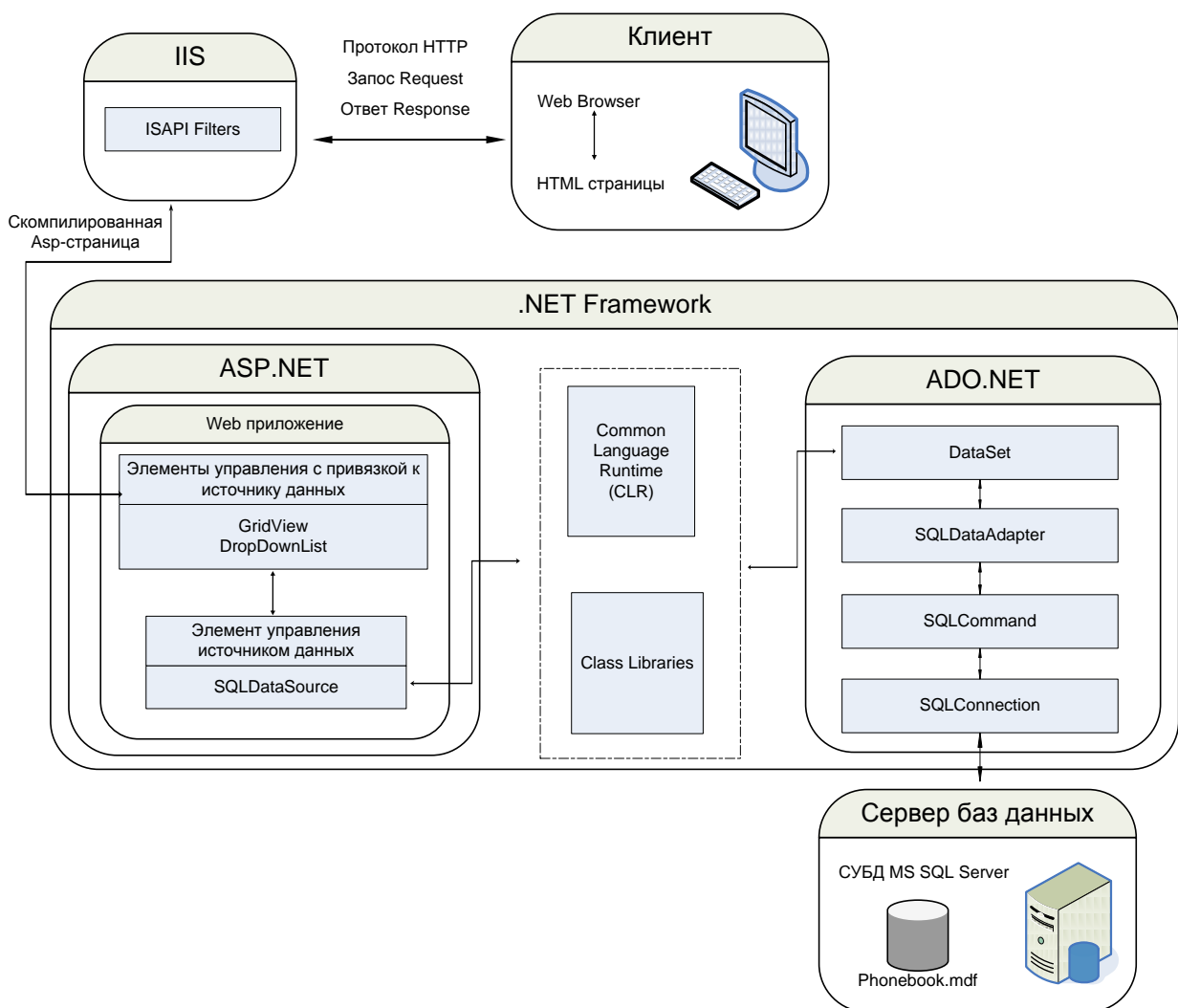


Рисунок 6.1 – Архитектура, основанная на применении ASP.NET

6.5 Важными особенностями серверных компонентов ASP.NET являются возможность обработки на сервере событий, возникающих в клиентском приложении, и возможность генерировать HTML-, WML- и

СНТМL-код в зависимости от типа клиента и поддерживаемых им языков разметки и протоколов передачи данных.

6.6 Технология ASP.NET 2.0, представляющая собой дальнейшее развитие технологии ASP.NET.

ASP.NET 2.0 обеспечивает доступ к расширенному набору готовых блоков, шаблонов страниц, прикладных интерфейсов, а также к средствам персонализации, к инструментам для поддержки локализации приложений, к утилитам развертывания, позволяющим распространять приложения без предоставления исходного кода.

Технология позволяет создавать Web - приложения, близкие в плане удобства и пользовательского интерфейса к Windows-приложениям.

6.2 Серверное приложение WebBroker является кроссплатформенным. Модуль может быть скомпилирован и как dll библиотека ISAPI и как модуль Web - сервера Apache.

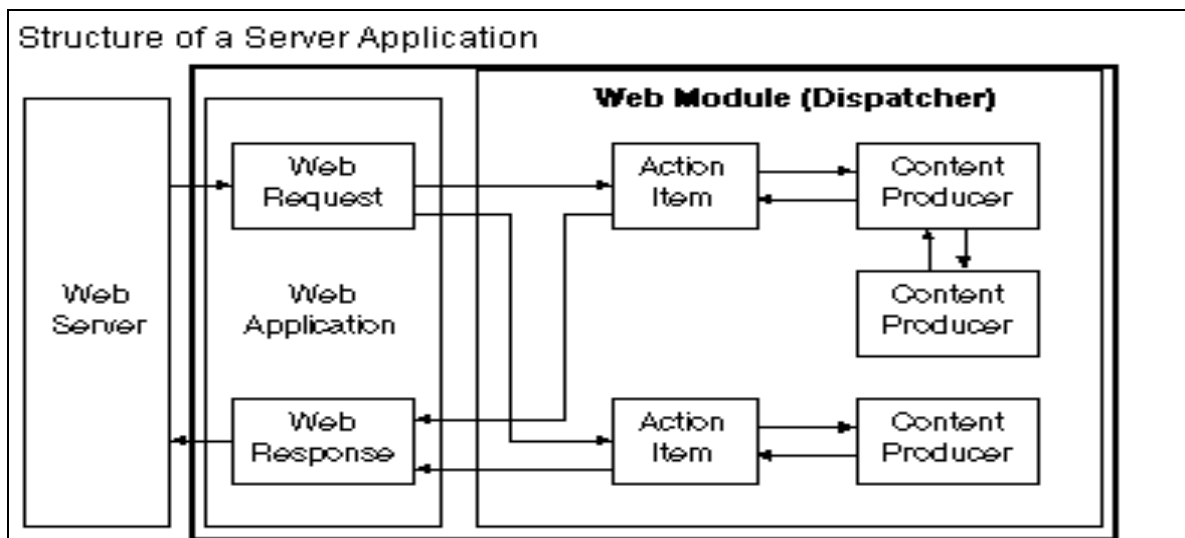


Рисунок 6.2 – Структура серверного приложения WebBroker

Особенностью приложения является диспетчеризация коллекции действий, использование шаблонов с динамическими тегами.

6.7 Web-диспетчер (TWebDispatcher) создаваемый при запуске модуля, управляет обработкой HTTP-запросов, для генерации ответов клиентам. Когда объект TWEBAPPLICATION получает HTTP-запрос, диспетчер передает запрос соответствующему действию или нескольким действиям для формирования ответа.

6.8 Каждое действие (Action items) вводится как элемент массива всех действий (ActionItems) данного модуля и описывается следующими свойствами:

Default – определяет, вызывается ли данное действие по умолчанию;

Enabled – определяет включено/отключено действие;

MethodType – определяет тип HTTP – запроса для данного действия;

Name – содержит название действия;

PathInfo – содержит часть строки URL, определяющей вызов этого действия; В качестве примера -

<FORM method="post" action="http://127.0.0.1/cgi-bin/test.exe/**registr**">

Producer – содержит имя объекта, отвечающего за генерацию ответа;

On Action – название процедуры (обработчика события), вызываемой данным действием.

Действия могут полностью отвечать на запрос или подготавливать только часть ответа с последующей передачей обработки другим действиям.

6.9 Программная генерация действия.

Код создания действия, например, в обработчике события BeforeDispatch можно написать:

```
var
```

```
WAI : TWebActionItem;
```

```
WAI.Create(nil);
```

```
WAI.MethodType:=mtPost;
```

```
WAI.PathInfo:='/start';
```

```
WAI.Producer:=PageProducer1; //Нельзя к двум действиям.
```

```
WAI.OnAction:=WebModule1.webActionItem3Action;
```

```
WAI.Enabled:=true;
```

Управлять свойствами действий в момент работы диспетчера нельзя!!!

6.10 Обращение к данным в модуле.

В отличие от «классического» CGI, реализованного консольным приложением, реализованные в среде Delphi алгоритмы, не нуждаются в прямом доступе к переменным окружения.

Данные клиентского приложения, передаваемые в Web – модуль, доступны при помощи объекта Request класса TCGIRequest. В полях объекта собраны воедино все данные пользователя, включая и данные Web – формы.

6.11 Вопрос.

Когда происходит перекодировка символов кириллицы?

6.12 Свойства TWebRequest:

Accept – тип MIME – данных, которые готов принять клиент в ответном потоке;

Authorization – содержит информацию об аутентификации;
 Cachecontrol – описание способа кэширования информации клиентом;
 Connection – содержит тип соединения, которое поддерживает браузер клиента;
 Content – содержит тело запроса при передаче данных методом POST;
 ContentLength – число символов в запросе;
 ContentType – содержит тип данных, отправленных в запросе;
 Cookie – поля cookie – файлов;
 Data – содержит дату и время отправки запроса;
 Method – используемый метод запроса;
 PathInfo – часть URL, содержащую адрес модуля, обрабатывающего запрос;
 Query – строка запроса (часть URL после знака ?) при передаче данных методом GET;
 RemoteAddr – IP адрес клиента;
 UserAgent – определяет модель браузера и операционную систему;
 ContentFields – Tstrings – массив, содержащий поля POST – запроса;
 CookieFields – Tstrings – массив полей cookie, переданных вместе с запросом;
 QueryFields – Tstrings – массив полей GET – запроса.
 Для методов POST и GET необходимо знать отличия и учитывать их при разработке модуля.

6.13 Пример программы, обратить внимание на строки в коде программы.

```

// s:= Request.ContentFields.Text;
// ShowMessage(s);

// s:= Request.PathInfo;
// ShowMessage('IEPathInfo='+s);
  
```

Просмотр параметров клиента производится аналогичным способом:

```

//s:= Request.RemoteAddr;
//ShowMessage(s);
  
```

Данные о типе клиентского приложения используются сервером для формирования соответствующего заголовка.

6.14 Пример обращение к значениям переменных:

```

var
  su, n, z :string;

begin
  su:=Request.ContentFields.Values['surname'];
  
```

```
n:= Request.ContentFields.Values['name'];
z:= Request.ContentFields.Values['num_z'];
... ..
```

Вспомнить вопрос, – «Какие типы данных можно передать с HTML – формы на сервер?».

6.15 Управление данными в таблицах БД.

Запись данных в серверную БД производится при помощи SQL - оператора INSERT.

Синтаксис:

```
INSERT INTO <object> [(col [, col ...])]
{ VALUES (<val> [, <val> ...]) | <select_expr> };
<object> = tablename | viewname
<val> = {
<constant> | <expr> | <function> | NULL | USER
} [COLLATE collation]
```

Удаление записей - SQL – оператор DELETE.

Синтаксис:

```
DELETE FROM table [WHERE <search_condition>];
```

<search_condition> - Условия поиска, которые определяют строки для удаления. Если это предложение не используется, оператор воздействует на все строки в определенной таблице.

Изменение записей - SQL - оператор UPDATE изучить самостоятельно.

Перечисленные операторы управления данными могут вызываться при помощи компонентов TQuery или при помощи механизма хранимых процедур.

6.16 Компонент TIBQuery.

Поместить компонент в модуль данных или создавать на стадии прогона программы. Настройка основных свойств:

```
try
IBQuery1 := TIBQuery.create(nil);
IBQuery1.Database := etest;
IBQuery1.SQL.Add('insert into IS2003 ');
IBQuery1.SQL.Add('values (1, :SURNAME, :NAME, :NUM_Z)');
IBQuery1.Params[0].AsString := su;
IBQuery1.Params[1].AsString := n;
IBQuery1.Params[2].AsInteger := StrToInt(z);
```

```

IBQuery1.ExecSQL;
IBQuery1.Destroy;
IBTransaction1.Commit;
except
// ShowMessage('Ошибка регистрации');
  Response.Content := PageProducer1.Content;
end; //try

```

6.17 Прямое задание ответа (TWebResponse).

Необходимо определить алгоритм поведения программы при неудачной попытке записи данных в БД. Одним из способов является отправка клиенту заранее подготовленной Web – страницы с соответствующим сообщением.

В нашем примере клиенту будет отправлена повторно форма регистрации, с пустыми значениями полей. Это реализуется оператором:

```
Response.Content := PageProducer1.Content.
```

6.18 Вопрос.

Тип данных свойства Content объекта Response?

6.19 Технология разработки Web-приложений WebSnap представляет собой законченную платформу разработки Web-приложений в Delphi за счет использования самых современных средств быстрой разработки приложений (RAD). Инструменты, входящие в состав WebSnap, например отладочный Web-сервер, упрощают отладку и тестирование приложений. Предусматривается возможность интеграции приложений WebSnap в корпоративные Web-сайты и порталы, разработанные с использованием таких известных средств, как DreamWeaver и FrontPage, а также поддержка серверных сценариев на JavaScript.

6.20 Примеры среды Delphi расположены в каталоге:

```
C:\Program Files\Borland\Delphi7\Demos\WebSnap.
```

Каждый пример имеет файл описания. Инсталляция производится только после инсталляции пакета для отладчика. Может потребоваться загрузка дополнительных библиотек.

Запуск из меню «Tools/Web App Debugger».

6.21 Состав репозитория.

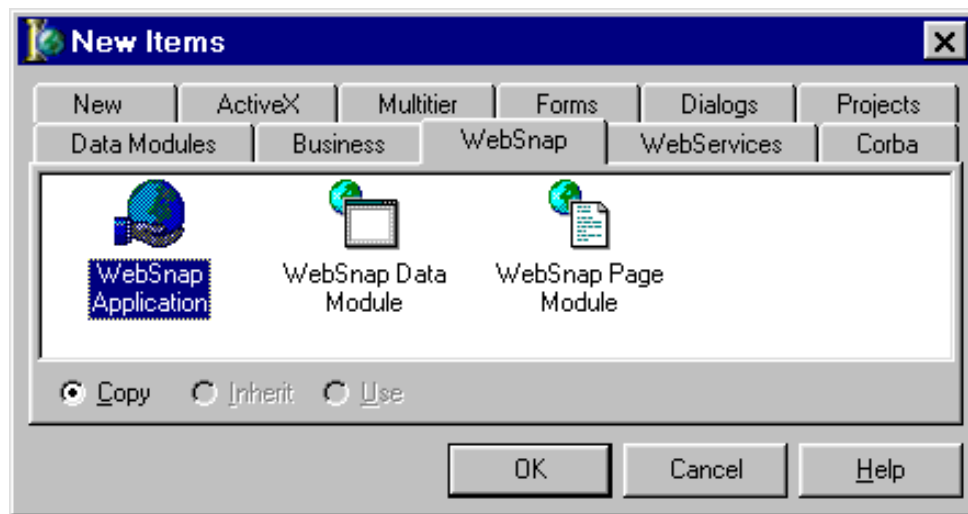


Рисунок 6.3 – Репозиторий технологии WebSnap

6.22 Шаблон Web - приложения.

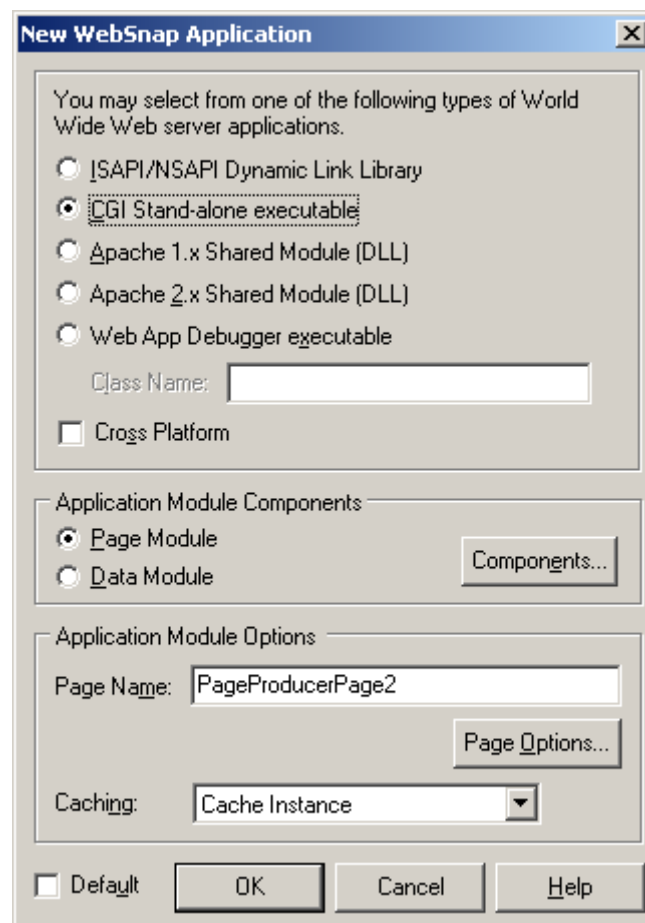


Рисунок 6.4 – Установка параметров

6.23 Вопросы по 2 лабораторной работе.

Методологии проектирования информационных систем.

Технологии серверных модулей.

Архитектура и структура серверных приложений.

Принципы навигации в сложных проектах.

Особенности реализации обработки ошибок.

Типы данных, используемых в сетевом взаимодействии.

6.24 Контрольные задания для СРМП:

1. Защита выбора типов данных, определения пользователей;
2. Выбор и защита уровней доступа пользователей;
3. Ответить на контрольные вопросы второго модуля;
4. Провести отладку кода серверного модуля;
5. Защитить отчет по второй лабораторной работе;
6. Тест первого рубежного контроля.

6.25 Контрольные задания для СРМ:

1. Изучить конспект 4,5,6 лекций;
2. Изучить методические указания ко второй лабораторной работе;
3. Ответить на примеры тестовых заданий ко второму модулю;
4. Выбор и настройка компонентов базовых технологии;
5. Тестирование межпрограммного взаимодействия;
6. Проверка идентификации пользователей;
7. Изучить работу и код программ учебного примера.

6.24 Демонстрация.

Создание тестовой формы.

Извлечение данных клиентской формы.

Отладка клиент – серверного взаимодействия.

Обработка ошибок обращения.

Демонстрация отладки кода оператора.

7 ТЕХНОЛОГИИ ИНФРАСТРУКТУРЫ

Полный комплекс технологий для построения ИТ – инфраструктуры и управления:

- семейство базовых программных технологий;
- решения для коллективной работы;
- полнофункциональный комплекс бизнес приложений и интеграционное решение для управления данными.

Формирование серверного ответа:

- диспетчеризация ответов серверным – модулем;
- ошибки формирования ответов;
- стандартные коды ошибок.

7.1 Решения оптимизации инфраструктуры.

Для обеспечения поддержки быстроменяющихся потребностей бизнеса необходима формализация ИТ – услуг по определенным правилам и методикам.

Предлагаемые решения:

1. Исследование существующей структуры ИТ-подразделения.
2. Формализация списка услуг, предоставляемых ИТ-подразделением.
3. Разработка соглашений о качестве предоставляемых услуг:
 - разработка ключевых критериев качества (параметров доступности) для каждой услуги;
 - определение значений для каждого параметра;
 - создание соглашений о качестве для каждой услуги (SLA).

7.2 Автоматизация процессов управления ИТ-услугами:

- выбор и поставка программных продуктов для управления ИТ-услугами;
- установка и настройка программного обеспечения.

7.3 Внедрение единого центра обслуживания (HelpDesk, ServiceDesk):

- разработка структуры;
- разработка методик работы;
- разработка должностных инструкций;
- разработка правил взаимодействия потребителей услуг с центром обслуживания.

7.4 Контроль и управление:

1. обучение персонала;
2. создание проектной документации;

3. снижение издержек по содержанию ИТ - инфраструктуры, минимизация потерь, связанных со сбоями автоматизированных систем предприятия.

4. оценка значения ИТ - подразделения для компании и эффективность его работы. Реальный контроль за выполнением услуг, заказанных ИТ - подразделению, на основе соглашений о качестве предоставления услуг.

7.5 Средства создания порталов.

Портал представляет собой Web-сайт, предназначенный для определенной аудитории (например, для клиентов и сотрудников компании), осуществляющий анализ, обработку и доставку информации, и предоставляющий доступ к различным сервисам и приложениям на основе авторизации пользователей.

Условно порталы можно разделить на корпоративные (ориентированные на сотрудников и клиентов одной компании), вертикальные (выпускаемые для разных отраслей рынка и обслуживающие пользователей, которые работают на этом рынке или потребляют его услуги) и горизонтальные (предназначенные для широкой аудитории).

7.6 Основные разработки:

WebLogic Portal Server (BEA);
 WebSphere Portal Server (IBM);
 Oracle 9i Application Server Portal (Oracle);
 Enterprise Portal (PeopleSoft);
 Plumtree Corporate Portal (Plumtree Software);
 SAP Enterprise Portal (SAP);
 Sun Java System Portal Server (Sun Microsystems);
 Vignette Application Portal 7 (Vignette).

7.7 Средства управления информационным наполнением Web-сайтов и Web-порталов (Content Management Systems, CMS):

Documentum (EMC);
 P8 (FileNet);
 Hummingbird Document Management (Hummingbird);
 Content Manager (IBM);
 TeamSite (InterWoven);
 Livelink Enterprise Suite (OpenText);
 Universal Content Management (Stellent);
 Vignette Content Suite 7 (Vignette).

7.8 Средства создания приложений электронной коммерции.

Приложения электронной коммерции реализуют схемы «предприятие—потребитель» (B2C, business-to-consumer), или «предприятие—предприятие» (B2B, business-to-business):

WebSphere Commerce (IBM);
 Commerce Server 2002 (Microsoft);
 E-Business Suite (Oracle).

Обычно в основе всех указанных категорий продуктов лежат серверные СУБД и серверы приложений, основанные либо на технологии J2EE, либо на технологиях COM/COM+/.NET.

Не редкость для таких продуктов и наличие средств интеграции с ERP- и CRM-системами.

7.9 Основы Web-служб.

В основе Web-служб лежат несколько простых принципов:

1. возможные для вызова команды описываются на языке WSDL (Web Service Description Language);
2. непосредственная активизация команд происходит в виде отправки SOAP (Simple Object Access Protocol) - сообщений по адресу, где располагается Web-служба (используется стандартный протокол HTTP);
3. для поиска Web-служб существуют глобальные или локальные (внутренние) каталоги, поддерживающие стандартные службы обнаружения UDDI (Universal Description, Discovery and Integration Protocol).

Все современные средства разработки ведущих производителей поддерживают создание Web-служб, а программные платформы (будь то серверные операционные системы или серверы приложений) обеспечивают выполнение Web-служб.

7.10 В Web-службах везде используется язык XML. Он служит, в частности, для описания сообщений, которыми могут обмениваться Web-службы и их потребители.

SOAP-сообщение — это XML-документ, состоящий из трех базовых элементов: <Envelope>, <Header> и <Body>.

Язык WSDL соответствует спецификации XML и позволяет создавать XML-документы, описывающие методы Web-служб, параметры методов, способы их вызова и т.п. Для того чтобы воспользоваться специализированными Web-службами в рамках механизмов обнаружения UDDI, следует составить SOAP-сообщения и интерпретировать возвращаемые XML-документы.

7.11 Web-службы дополнились и собственной архитектурой (Web Services Architecture, WSA) и обширным набором дополнительных стандартов и протоколов для решения таких задач, как гарантированная передача сообщений, обеспечение адресации, безопасность, передача вложенных данных.

Вопросы, связанные с обеспечением полной совместимости Web-служб, реализованных на различных платформах, выделились в особую категорию. Появилась даже специальная организация — Web Services-Interoperability Organization (WS-I), отвечающая за создание профилей, описывающих принципы совместимых Web-служб. В настоящее время уже разработан базовый профиль (WS-I Basic Profile) для SOAP 1.1 и WSDL 1.0.

7.12 Архитектура модуля WebSnap включает:

1. WebSnap Application, контейнер создаваемого приложения;
2. Web Page Module – контейнер шаблонов генераторов страниц;
3. Web Data Module – модуль данных приложения.

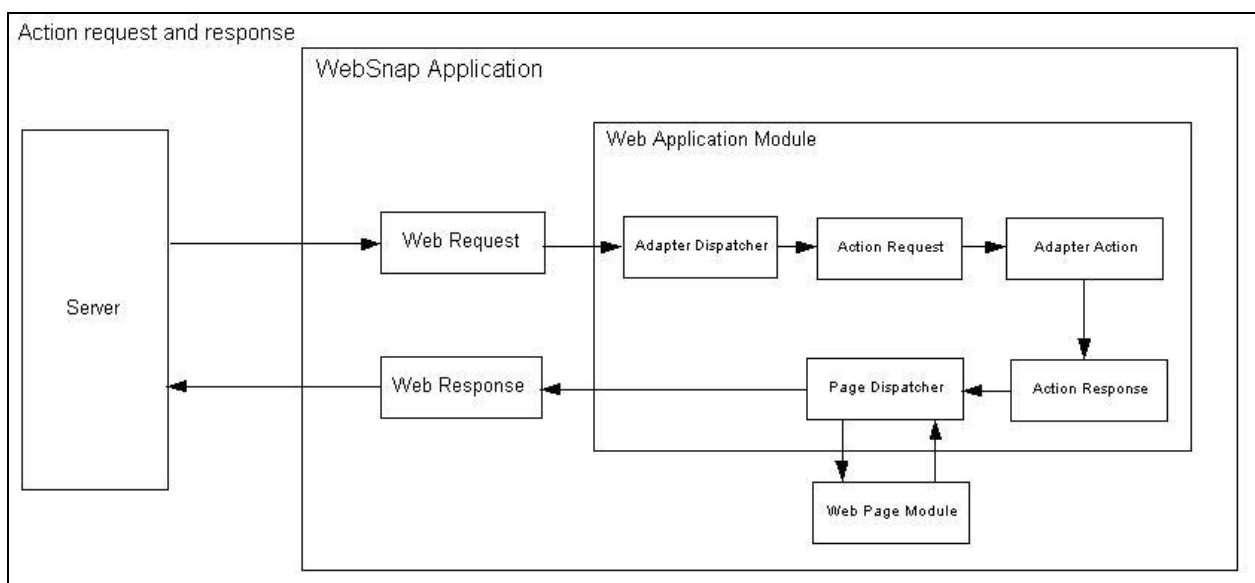


Рисунок 7.1 – Структура серверного приложения WebSnap

Структура модуля включает компоненты Adapter Dispatcher для автоматической обработки ввода данных HTML – формы, и PageDispatcher для анализа пользовательских запросов для переадресации их в модули обработки.

7.13 Технология интерпретации серверных скриптов позволяет автоматизировать все основные функции управления данными за счет интегрирования в HTML – документ фрагментов кода JavaScript для генерирования всех необходимых событий. Обеспечивая при этом автоматическую идентификацию этих событий серверным модулем.

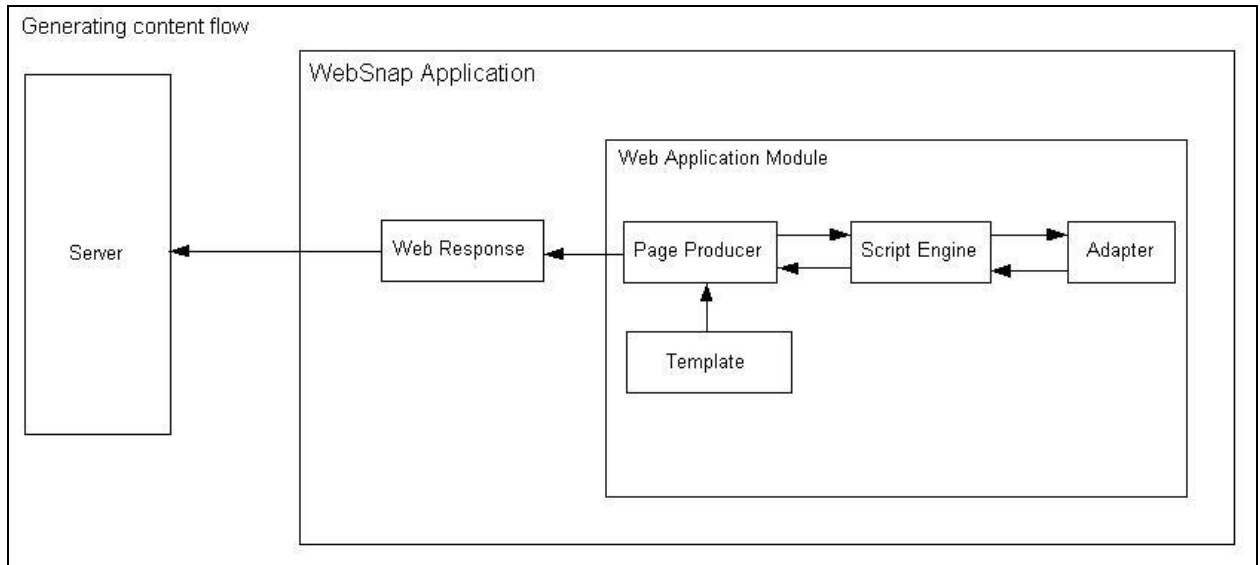


Рисунок 7.2 – Структура генерации кода JavaScript.

7.14 Технология обеспечивает наборы шаблонов для выполнения стандартных действий над данными.

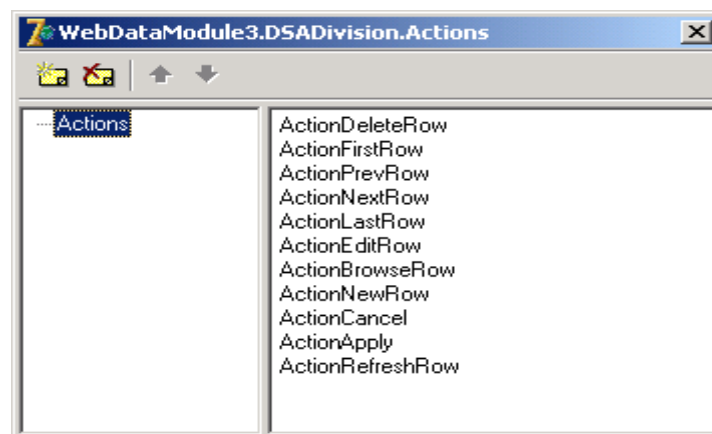


Рисунок 7.3 - Список выполнимых действий

7.15 Демонстрация.

Демонстрация создания Web – приложения.

Работа учебного примера.

Реализация меню проекта.

Реализация стандартных наборов действий.

8 КЛАССИФИКАЦИЯ СИСТЕМ

Функционально-информационное представление.

Классификация.

Примеры промышленных информационных систем.

Полнофункциональный комплекс бизнес приложений и интеграционное решение для управления данными.

Формирование ответа на основе наборов данных (НД), использование шаблонов.

Компоненты Delphi (IntraWeb) для формирования ответа.

8.1 Функции ERP систем.

Финансовое управление и составление отчетности.

Обеспечение производственного процесса (планирование, управление).

Управление продажами.

Управление закупками.

Управление хранением и перемещением.

Управление техобслуживанием и ремонтом.

Управление персоналом.

На рисунке 2.2 показано функционально-информационное представление. Такой функциональный элемент как управление продажами, может включать в себя множество процессов.

8.2 Вопрос.

Какие типовые конфигурации реализованы в системе «1С Предприятие»?

8.3 Классификация современных систем.

1. Системы управления ресурсами предприятий ERP (Enterprise Resource Planning) и MRP (Manufacturing Resource Planning);

2. Системы управления активами и фондами EAM (Enterprise Asset Management);

3. Системы управления взаимоотношениями с клиентами - CRM (Customer Relationship Management);

4. Системы управления цепочками поставок SCM (Supply Chain Management).

8.5 Специфика систем управления определяется [4]:

- отраслевой принадлежностью, корпоративными стандартами;
- видами производства (серийное, сборочное, непрерывное, дискретное);

- типами организации производственного процесса (производство на склад, производство на заказ, циклическое производство).

Любая специфика требует от системы;

- наличия средств разработки, позволяющих наращивать функциональность системы;
- возможности интеграции с другими системами.

8.6 Специфика ЕАМ – систем, заключается в управлении ориентированном не на продукт, а на процесс. Используется в отраслях с большим капитальным вложением (горнодобывающая, распределительные сети, эксплуатация инженерных коммуникаций). Основными процессами в этих отраслях являются: техническое обслуживание и ремонт, управление материалами и финансами.

8.7 Основная функциональность CRM – систем заключается в автоматизации:

- продаж;
- маркетинга;
- обслуживания клиентов.

8.8 Примеры:

R/3(SAP);
Oracle Application;
Галактика;
Scala;
1С Предприятие.

8.9 Переход к IntraWeb.

Дальнейшее развитие технологий WebBroker и WebSnap реализовано в Delphi 7 компанией AtoZedSoftware. Новая технология получила название IntraWeb благодаря архитектурной особенности, позволяющей устанавливать приложение, как на сервере, так и на локальной машине клиента.

Доступ к приложению осуществляется через браузер, например, Internet Explorer.

При этом IntraWeb - приложение генерируя HTML - страницы, предоставляет пользователю различные возможности, доступ к данным в БД и взаимодействие с серверными модулями.

С помощью технологии IntraWeb можно создавать как самостоятельные приложения, так и страницы, предназначенные для рассмотренных ранее технологий WebBroker и WebSnap.

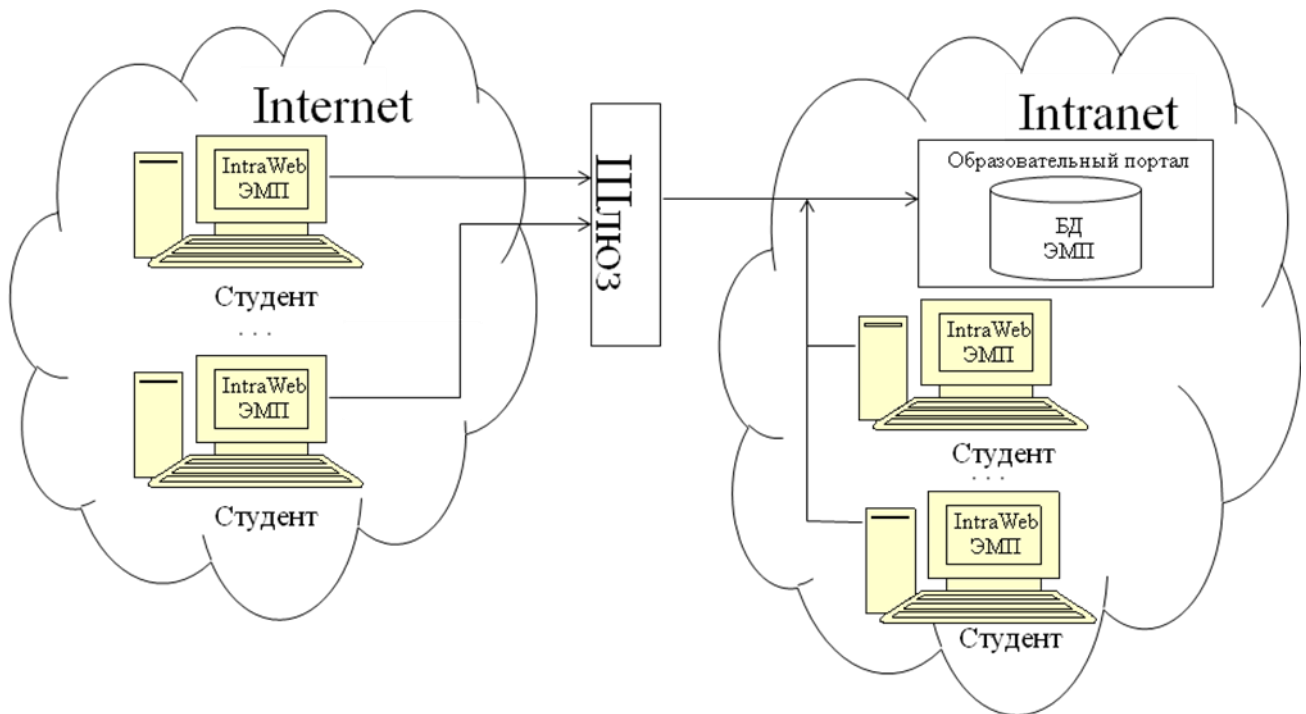


Рисунок 8.1 – IntraWeb архитектура

8.10 Основные особенности архитектуры IntraWeb:

- она позволяет проектировать приложения серверов Web так же, как проектируется обычное приложение Delphi что существенно облегчает разработку, так как позволяет использовать все преимущества визуального проектирования;
- внешний вид формы совпадает с внешним видом страницы, происходит автоматическая конвертация программного кода в код HTML и JavaScript;
- автоматически генерируется система обработчиков (действий), реализуемых клиентскими и серверными скриптами.

8.11 Компоненты, предназначенные для IntraWeb, расположены в Delphi 7 на страницах библиотеки IW Standard, IW Data, IW Client Side, IW Control.

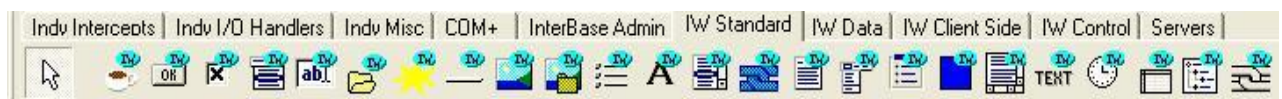


Рисунок 8.2 – Компоненты «IW Standart» IntraWeb

Большинство этих компонентов имеют имена, совпадающие с их аналогами в VCL и CLX. Только перед именем добавляются символы «IW». Пиктограммы компонентов также похожи на аналоги VCL

8.12 IntraWeb-приложения предназначены для следующих типов серверов:

- ISAPI/NSAPI;
- Apache versions 1 and 2;
- CGI (page mode only);
- Windows services.

8.13 В страничном режиме, приложения могут создаваться в форматах ISAPI, Apache DSO, NSAPI, CGI, или Win-CGI. В режиме приложения – только в форматах ISAPI, NSAPI и Apache DSO.

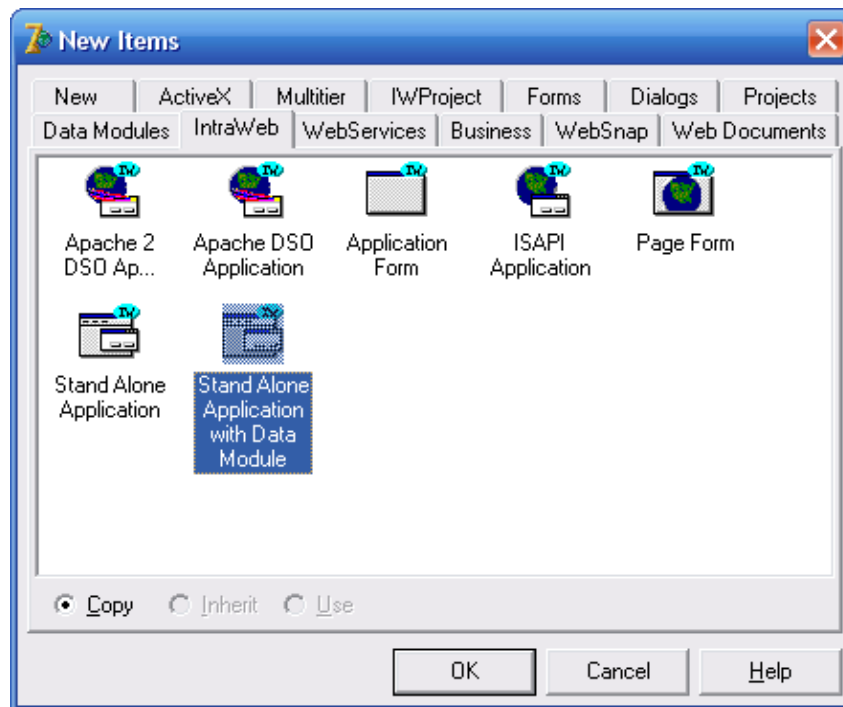


Рисунок 8.3 – Вкладка IntraWeb репозитория Delphi

8.14 Встроенный сервер IntraWeb обеспечивает автоматический просмотр работы проекта.

При работе отслеживается количество активных сессий, возможно настройка отображения заголовков запросов пользователей.

Проблему корректировки кода сервера ни кто не решил!!!

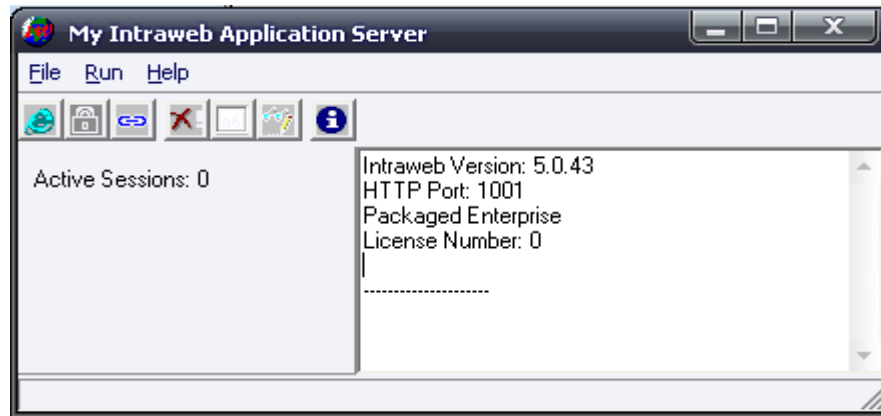


Рисунок 8.4 - Окно сервера IntraWeb приложения

8.15 Выбор списка компонентов для реализации проекта!

8.16 План лабораторной работы «Проектирование серверных модулей».

Организация вывода информации из базы данных в ответ на запрос клиента.

Проверка интерактивного взаимодействия клиента с серверным – модулем.

Проектирование механизма навигации по проекту.

Публикация проекта в Интранет.

Тестирование сетевого взаимодействия.

8.17 Контрольные задания для СРМП:

1. Защита выбора механизма навигации;
2. Выбор и защита компонентов отображения данных;
3. Ответить на контрольные вопросы третьего модуля;
4. Провести отладку кода динамического формирования управляющих элементов;
5. Защитить отчет по третьей лабораторной работе;
6. Защитить раздел методического обеспечения дисциплины.

8.18 Контрольные задания для СРМ:

1. Изучить конспект 7,8 лекций;
2. Изучить методические указания к третьей лабораторной работе;
3. Ответить на примеры тестовых заданий к третьему модулю;
4. Выбор и настройка компонентов механизма навигации;
5. Выбор и настройка компонентов механизма генерации данных;
6. Тестирование передачи данных пользователю;
7. Разработать раздел методического обеспечения дисциплины;
8. Изучить работу и код программ учебного примера.

8.19 Демонстрация.

Инсталляция и работа учебного примера.

Работа оболочки электронного методического пособия.

Проектирование IntraWeb – приложения.

Механизмы навигации в проекте.

9 INTERNET/INTRANET/EXTRANET ТЕХНОЛОГИИ

Технологии разработки Internet/Intranet/Extranet приложений, реализуемые в среде Delphi.

Обзор компонентов Delphi для проектирования Интернет – приложений.

Компонент Браузера и Web - сервера.

Компоненты для разработки FTP – модулей.

Компоненты для работы с электронной почтой.

Тренировочный тест к рубежному контролю.

Примеры среды Delphi.

9.1 Интернет/Инtranет технологии:

- электронная почта - E-mail (SMTP, IMAP, POP, X-400);
- сервис FTP (FTP, TFTP);
- всемирная паутина - WWW (HTTP, URL, CGI, ISAPI, ASP, JavaScript, Perl, HTML, CSS);
- сервис новостей – News (NNTP);
- поисковые системы (Wais, Gofer);
- эмулятор сетевого терминала – Telnet;
- служба контроллера домена;
- служба доменных имен – DNS;
- сервис удаленного доступа – RAS;
- пограничный сервер – Proxy;
- сервис коллективной работы (порталы);
- сетевое администрирование.

9.2 Среда Delphi.

Компоненты сетевого взаимодействия технологии Indy.

Помимо базовых служб и протоколов Интернет существует широкий набор дополнительных сервисов, возможности которых часто используются разработчиками. К тому же далеко не всегда возможность отображения информации с помощью браузера является приемлемым решением для Интернет - приложений. В этом случае разумно использовать Интернет-инфраструктуру для обмена данными, а отображение информации обеспечить за счет более сложных клиентских приложений, разработанных на Delphi [6].

Для решения такого класса задач в состав Delphi включена библиотека Internet Direct (Indy) компании Nevrona Designs (<http://www.nevrona.com/Indy/>). Данная библиотека, разработанная специально для Borland Delphi, насчитывает уже восемь версий, последняя из которых вошла в состав новой версии Delphi. Набор компонентов разделен

на три группы: клиентские (Indy Client), серверные (Indy Servers) и вспомогательные (Indy Misc).

9.3 Большинство компонентов Indy Client и Indy Servers представляют собой пары, соответствующие клиентским и серверным частям протоколов и служб (за исключением отдельных, в основном серверных, компонентов типа TunnelMaster и TunnelSlave), и позволяют использовать такие протоколы, как TCP/IP, UDP, NNTP, SMTP, FTP, HTTP, а также службы ECHO, FINGER, WHOIS и т.д.



Рисунок 9.1 – Палитра компонентов Indy Client

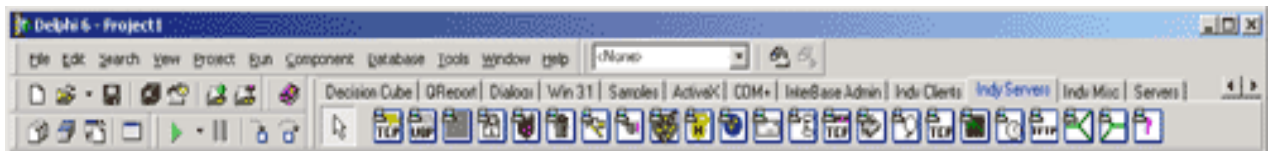


Рисунок 9.2 – Палитра компонентов Indy Servers

Клиентские компоненты Indy написаны с использованием сокетов. Сокет со стороны клиента требует соединения с сервером. Если связь установлена, клиент и сервер могут начинать обмен сообщениями. Эти сообщения носят различный характер, но обычно обмен происходит по определенному протоколу (например, HTTP).

9.4 TIdTCPClient и TIdTCPServer. Эти компоненты используются для поддержки одного из основных сетевых протоколов — TCP (Transmission Control Protocol), а также являются базовыми классами для компонентов TIdSMTP и TIdFTP. Класс TIdTCPServer обладает свойством ThreadMgr, по умолчанию равным nil. Если ThreadMgr равно nil, когда TIdTCPServer активизирован, класс TIdThreadMgrDeafault будет создан неявно. В противном случае используется установленный менеджер процессов.

9.5 TIdFTP. Компонент включает полную поддержку протокола передачи файлов — FTP (File Transfer Protocol). Поддерживается пассивная и активная передача данных, а также такие операции, как GET и PUT, удаление директорий, получение квот, размеров файлов и каталогов. В своей работе TIdFTP использует класс TIdSimpleServer. Когда выполняется передача файла по протоколу FTP, вторичное соединение по протоколу TCP открыто для передачи данных и закрывается, когда данные были переданы. Такое

соединение называется «канал передачи данных», уникальный для каждого передаваемого файла.

9.6 TIdHTTP и TIdHTTPServer. Компоненты используются для обеспечения сетевого протокола HTTP (поддерживаются версии 1.0 и 1.1, включая операции GET, POST и HEAD). Кроме того, обеспечивается поддержка аутентификации и применения прокси-серверов. Серверный компонент используется для предоставления услуг другому Web-серверу, поддерживающему данный протокол. TIdHTTPServer облегчает реализацию таких функций, как cookies, управление состояниями и др.

9.7 TIdPOP3. Клиентский компонент, предназначенный для обеспечения протокола POP (Post Office Protocol), включая поддержку MIME-кодирования и декодирования, а также передачу многобайтных символов.

9.8 TIdSMTP. Клиентский компонент, предназначенный для применения в приложениях протокола SMTP (Simple Mail Transfer Protocol), обеспечения поддержки аутентификации, MIME-кодирования и декодирования, а также для поддержки многобайтных символов.

9.9 TIdMessage. Компонент используется в комбинации с другими компонентами, чтобы должным образом расшифровать или кодировать сообщения. Это могут быть POP-, SMTP- и NNTP-компоненты. Класс поддерживает MIME-шифрование и расшифровку, многобайтные символы и кодировку ISO.

9.10 Использование компонента TWebBrowser.

Вкладка «Internet» представляющая компоненты технологии WebBroker, содержит компонент браузера – TWebBrowser. Используя элемент управления ActiveX WebBrowser, входящий в состав Microsoft Internet Explorer, компонент позволит программе работать с данными в формате HTML.

Отправлять запросы на сервера по любым типам URI, получать ответы и отображать содержимое в окне браузера. Все последние версии Windows содержат TWebBrowser в своем составе и без него практически неработоспособны.

9.11 Базовые операции.

Для того чтобы использовать TWebBrowser в своей программе, следует разместить его на форме. Затем, для отображения в нем страницы HTML, необходимо вызвать его метод Navigate:

```

procedure TForm1.Button1Click(Sender: TObject);
var
  Flags, TargetFrameName, PostData, Headers: OleVariant;
begin
  WebBrowser1.Navigate('http://www.borland.com', Flags,
    TargetFrameName, PostData, Headers);
end;

```

9.12 Параметры, передаваемые в метод Navigate.

Первым параметром является строка с URL, указывающим адрес, из которого должна осуществляться загрузка. Поддерживаются все протоколы, доступные в IE, например,

file:// — загрузка файла,
 res:// — загрузка из ресурса.

Остальные параметры не являются обязательными и служат для передачи дополнительной информации.

Подробнее работа с компонентом описана в статье Анатолия Тенцера, журнал «КомпьютерПресс 6'2001».

9.13 Доступ к документной модели TWebBrowser.

Иерархическое объектное представление HTML-объектов называется DOM (Document Object Model).

В Internet Explorer реализовано расширение HTML, называемое Dynamic HTML (DHTML). Эта модель представляет все элементы HTML-документа в виде набора коллекций объектов, доступных для изменения.

Скрипты, встроенные в страницы и приложения и имеющие доступ к этим коллекциям, могут находить и изменять их элементы, а также добавлять новые, причем изменения будут немедленно отражены в окне TWebBrowser.

9.14 Возможности.

Наиболее интересным является параметр PostData, позволяющий передать на Web-сервер данные, полученные в результате заполнения формы, если этот сервер использует HTTP метод POST.

Предоставление пользователю возможности работать с полученными данными, как и многие другие функции, доступны через метод ExecWB, предоставляющий простой способ обращения к интерфейсу IOleCommandTarget.

Можно настраивать интерфейс в зависимости от поддерживаемых текущей версией TWebBrowser возможностей.

Если требуется более тонкая настройка компонента, то необходимо реализовать интерфейс IDocHostUIHandler, позволяющий программисту взять под контроль поведение TWebBrowser.

9.15 Демонстрация.

Создание приложения с использованием компонента браузера.

Проектирование Web-сервера.

Демонстрация настройки учебного FTP-сервера.

Примеры среды Delphi.

10 НОВЫЕ ТЕНДЕНЦИИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Проверка базовых знаний, полученных при изучении предшествующих модулей.

Новые тенденции в информационных технологиях, интеллектуализация ИС, интеграция систем.

Вычислительная техника, программное обеспечение, телекоммуникационные технологии.

10.1 Основные направления развития науки.

Раздел интеграции и интеллектуализации подготовлен по учебному изданию «Интеллектуальные информационные технологии» Башмакова А.И., Башмакова И.А. [5].

Интеллектуальные информационные технологии базируются на результатах исследований многих областей науки. В первую очередь следует назвать дискретную математику, математическую логику, кибернетику, математическую лингвистику, искусственный интеллект (ИИ), психологию, системотехнику и др. Кроме того, в них используются все последние достижения технологий программирования, Интернет.

В качестве примера взаимного проникновения интеллектуальных информационных технологий и современных технологий программирования можно указать платформу Microsoft.NET, в создание которой корпорация вложила свыше 2 млрд. долл. и над развитием которой работают более 5 тыс. специалистов.

10.2 Программы решения отдельных интеллектуальных задач.

1. Программы компьютерного доказательства теорем;
2. Игровые программы;
3. Распознающие и узнающие программы;
4. Программы для семантического анализа и обработки естественно-языковой информации: машинный поиск в базах данных естественно-языковых документов; машинный перевод; автоматическое реферирование; автоматическая классификация документов; генерация (синтез) текста; генерация (синтез) речи;
5. Программы, моделирующие поведение;
6. Программы для анализа и синтеза музыкальных произведений.

10.3 Методы автоматического распознавания образов и их реализация в системах оптического чтения текстов (OCR-системах — Optical Character Recognition) — одна из самых плодотворных технологий ИИ.

Ведущие российские OCR-системы:

- Fine Reader, Fine Reader Рукопись и Form Reader фирмы ABBYY Software House (<http://www.abbyy.ru>), позволяющие распознавать как печатные, так и рукописные многоязычные тексты;

- CuneiForm (<http://www.cuneiform.ru>) фирмы Cognitive Technologies;

- Cognitive Forms фирмы Cognitive Technologies (<http://www.cognitive.ru>), предназначена для массового ввода структурированных документов (например, налоговых деклараций, бухгалтерских форм, платежных документов).

10.4 Работа со знаниями.

1. Методы и средства представления знаний: модели знаний; логические модели; продукционные модели; фреймы; семантические сети; онтологии; объектно-ориентированные модели;

2. Системы представления знаний и базы знаний, методы и средства извлечения знаний из различных источников (экспертов, документов), согласование и интеграция знаний

3. Методы обработки знаний: поиск знаний, верификация, систематизация, обработка нечетких знаний, Аргументация и объяснение на основе знаний.

10.5 Одной из фундаментальных моделей представления знаний, выраженных в текстовом виде является гипертекст (ГТ).

Основные понятия гипертекстовой информационной технологии:

1. Текст разбивается на фрагменты, представляющие его семантические единицы (сеты). Между ними устанавливаются связи, которые могут наделяться именами;

2. В отличие от обычного текста, который читается последовательно (в порядке, определенном его автором), ГТ можно читать, двигаясь по разным траекториям, образованным связанными сетями;

3. Активируемые переходы выбираются читателем (пользователем). Имена (типы) связей облегчают решение задачи выбора перехода.

В графовой модели ГТ вершины соответствуют вычлененным фрагментам текста, а ребра — возможным переходам между ними. Каждый путь на графе представляет отдельную линию прочтения текста.

10.6 Гипертекстовая информационная технология (ГИТ) — технология обработки семантической информации, основанная на использовании ГТ. Она относится к проблематике ИИ, так как ее содержанием является представление, поиск и обработка семантической информации, выраженной в текстах.

Области применения ГИТ весьма разнообразны:

- информационные ресурсы и технологии Интернет;
- гипертекстовые информационные модели экономических систем;
- базы данных с гипертекстовой организацией;

- электронные записные книжки;
- электронные картотеки, словари, энциклопедии, справочники;
- обучающие системы;
- экспертные системы;
- организация пользовательского интерфейса.

10.7 Интеллектуальное программирование.

1. Языки для интеллектуальных систем (языки логического программирования, объектно-ориентированные языки, языки представления знаний, языки семантической разметки);
2. Автоматический синтез программ (дедуктивные и индуктивные методы);
3. Инструментальные средства;
4. Интеллектуальные интерфейсы;
5. Мультиагентные технологии.

10.8 Интеллектуальные автоматизированные системы:

1. Нейропакеты;
2. Экспертные системы;
3. Интеллектуальные АСУ;
4. Интеллектуальные САПР;
5. Интеллектуальные роботы;
6. Системы управления знаниями;
7. Системы виртуальной реальности.

10.9 Развитие систем и моделей метаданных.

Метаданные (metadata) — это информация о документе, понимаемая ЭВМ, т. е. обладающая свойством внутренней интерпретируемости. В общем случае метаданные представляют собой информацию, характеризующую какую-либо другую информацию.

Система метаданных выступает в качестве центрального звена любой ИС. Как и в технологиях БД, для метаданных определяются два уровня представления:

- инфологический, фиксируемый схемой метаданных, которая отражает состав и структуру элементов данных (полей) в экземпляре метаданных, их семантику, типы значений (включая словари и классификаторы) и ограничения целостности;
- даталогический, фиксируемый форматом метаданных, который отражает способ представления (кодирования) информации.

Одной из наиболее перспективных моделей метаданных на сегодняшний день является модель RDF (Resource Description Framework), разработанная консорциумом W3C. Она определяет основные принципы представления и обработки метаданных и обеспечивает функциональную совместимость web-приложений, обменивающихся такой информацией. В

RDF использованы принципы объектно-ориентированного моделирования, элементы языков HTML, SGML и XML.

Синтаксис метаданных в RDF описывается на основе языка XML. Несмотря на то, что RDF была разработана в расчете на XML-платформу, она не зависит от XML. Данная модель позволяет представлять семантическую структуру XML-документов и выражать смысл этих и иных ресурсов WWW.

10.10 Системы метаданных:

- «Дублинское ядро» (инвариантный к ПО набор наиболее общих полей описания информационных ресурсов (ИР), введенный для обеспечения глобальной интероперабельности приложений, работающих с метаданными);
- MARC - предназначена для описания библиотечных ресурсов (как на бумажных, так и на электронных носителях);
- GILS - предназначена для описания любых видов ИР, расширяющая MARC и базирующаяся на протоколе Z39.50;
- ONIX - предназначена для описания товаров в системах электронной коммерции;
- LOM - предназначена для описания образовательных ИР;
- IAFA/WHOIS - предназначена для описания сетевых ИР;
- UDDI - предназначена для описания web-сервисов;
- EAD — предназначена для описания архивных материалов;
- GEM — расширение «Дублинского ядра» для описания образовательных ИР;
- МЕКОФ — международный коммуникативный формат, выступающий в качестве альтернативы MARC.

10.11 Перечень стандартов и спецификаций, составляющих ядро платформы XML. Полные описания опубликованы на сайте W3C (<http://www.w3.org>).

Таблица 10.1 Стандарты и спецификации ядра платформы XML.

Группа стандартов	Краткое название стандарта (спецификации)	Назначение
Идентификация ИР	URI	Uniform Resource Identifier - унифицированный идентификатор ресурса. Обобщенный формат идентификатора ИР. Его разновидностями являются URL и URN
	URL	Uniform Resource Locator - унифицированный указатель ресурса. Формат идентификатора ИР на основе его адреса в сети
	URN	Uniform Resource Name - унифицированное имя ресурса. Формат идентификатора, не зависящего от сетевого адреса ИР
Представление научной информации	MathML	Mathematical Markup Language - язык математической разметки. Является приложением XML и позволяет описывать математические выражения
	CML	Chemical Markup Language - язык химической разметки. Является приложением XML и позволяет описывать химические формулы и уравнения

Таблица 10.2 Стандарты и спецификации ядра платформы XML.

Группа стандартов	Краткое название стандарта (спецификации)	Назначение
Фундаментальные стандарты платформы XML	XML	Спецификация языка XML
	XML Information Set	Спецификация базовых типов данных
	Namespaces in XML	Спецификация механизмов использования пространств имен
Идентификация элементов данных в XML-документах и описание связей между ними	XPointer	XML Pointer Language — язык указателей XML. Определяет правила построения и использования указателей
	XLink	XML Linking Language — язык связывания XML. Определяет правила задания и интерпретации ссылок
	XInclude	XML Inclusions. Определяет механизм включения одного XML-документа в другой
Описание представления и правил трансформации XML-документов	XSL	Extensible Stylesheet Language — расширяемый язык стилей. Служит для описания представления XML-документов
	XSLT	Extensible Stylesheet Language for Transformation - расширяемый язык стилей для трансформации. Служит для описания правил преобразования XML-документов
	CSS	Cascading Style Sheet — язык описания каскадных таблиц стилей. Служит для описания представления XML-документов
Модели документов и представление метаданных	DTD	Document Type Definition — определение типа документа. Язык описания модели XML-документа
	XML Schema	Язык описания модели XML-документа
	RDF	Модель и язык представления метаданных, являющийся приложением XML
	RDF Schema	Метамодель и язык описания моделей метаданных, базирующихся на RDF
Языки запросов	XQuery (XQL)	XML Query Language — язык запросов для XML-данных
	XPath	XML Path Language — язык описания путей для доступа к фрагментам XML-документов. Используется в XPointer, XSLT и XQuery
Передача XML-данных	XMLP	XML Protocol — протокол передачи XML-данных. Определяет механизмы пакетного обмена XML-данными и способы удаленного вызова процедур в WWW
	SOAP	Simple Object Application Protocol — упрощенный прикладной протокол передачи XML-данных
	XForms	XML Forms. Определяет механизм передачи содержимого web-форм (например, запросов и результатов их выполнения) для XML-данных
Интерфейсы прикладного программирования	DOM	Document Object Model — объектная модель документа. Определяет объектную модель ИР (XML-документа) и API для его обработки
	SAX	Simple API for XML. API для синтаксического анализа XML документов, основанный на событиях
Обеспечение совместимости с web-1	XHTML	Extensible HyperText Markup Language - расширяемый HTML. Определяет модель XML-документа, соответствующую HTML 4, благодаря чему HTML 4 может интерпретироваться как приложение XML
	XML Base	Определяет способ представления базовых адресов в гиперссылках. Используется в XLink для поддержки соответствующих типов гиперссылок, фигурирующих в HTML-ресурсах
Обеспечение информационной безопасности	XML - Signature	XML-Signature Syntax and Processing. Спецификация электронной цифровой подписи для XML-документов
	XML Encryption	XML Encryption Syntax and Processing. Спецификация шифрования XML-данных

10.12 Развитие средств моделирования приведено по статье «Применение средств моделирования бизнес-процессов», КомпьютерПресс 1'2008.

BPEL представляет собой язык на основе XML для формального описания бизнес-процессов и протоколов их взаимодействия между собой. Он расширяет модель взаимодействия web-служб и включает в нее поддержку транзакций. Спецификация BPEL была совместно создана компаниями BEA Systems, IBM и Microsoft.

WSDL (Web Services Description Language) представляет собой основанный на XML язык описания интерфейса для доступа к web-сервису. Подобное описание содержит всю информацию, необходимую для доступа к данному сервису. Язык WSDL появился в результате объединения двух технологий: Network Accessible Service Specification Language (NASSL) фирмы IBM и Service Description Language (SDL) фирмы Microsoft.

Спецификация WSDL находится по адресу: <http://www.w3.org/TR/wsdl/>. С точки зрения WSDL-документа web-сервис представляет собой коллекцию портов, которые, в свою очередь, являются коллекцией абстрактных операций и сообщений. Абстракция операций и сообщений позволяет связывать их с различными протоколами и форматами данных типа SOAP, HTTP GET/POST или MIME.

10.13 Описание новой платформы проектирования приводится по статье Н. Елмановой «Eclipse сегодня и завтра», КомпьютерПресс 5'2006.

Eclipse (www.eclipse.org) - один из самых амбициозных и при этом успешных проектов в мире приложений с открытым кодом.

Инициатором проекта Eclipse стала в 1998 году корпорация IBM, поставившая перед собой цель создания интегрированной среды Java-разработки нового поколения. Осознав необходимость объединения с другими поставщиками Java-инструментов, IBM в конце 2001 года предоставила сообществу OpenSource часть исходного кода своего средства разработки Java-приложений WebSphere Studio Workbench и сформировала консорциум Eclipse. Целью данного проекта было создание платформы с открытым кодом, не уступающей по качеству коммерческим продуктам и расширяемой за счет встраиваемых в нее интегрируемых инструментов от различных производителей.

В 2004 году консорциум Eclipse был преобразован в независимую некоммерческую организацию Eclipse Foundation, насчитывающую в настоящий момент 115 членов). Стратегические вопросы работы Eclipse Foundation решаются Советом директоров Eclipse Board, в который входят четыре категории участников: стратегические разработчики (Strategic Developers), стратегические потребители (Strategic Consumers), поставщики расширений (Add-in Providers) и руководители проектов OpenSource. Из стратегических разработчиков и потребителей Eclipse Foundation следует

назвать компании BEA Systems, Borland Software, Computer Associates, Ericsson, HP, IBM, Intel, QNX, SAP, и Sybase.

10.14 Платформа Eclipse предоставляется в виде заранее скомпилированных исполняемых файлов для широкого спектра платформ, включая Windows, Linux, Solaris, HP-UX, AIX, QNX и Mac OS X. Благодаря продуманным прикладным программным интерфейсам и архитектуре, создание и добавление новых встраиваемых модулей в Eclipse является довольно простой задачей.

Основные направления развития Eclipse в настоящее время сосредоточены в шести проектах:

1. Eclipse - проект, в рамках которого создается платформа для разработки интегрированных инструментов и приложений;
2. Eclipse Tools - проект, в рамках которого осуществляется координация разработок в области инструментов (сред разработки для различных языков программирования, инструментов моделирования) для платформы Eclipse;
3. Eclipse Technology - проект по обеспечению исследовательской и образовательной деятельности с привлечением научных организаций и университетов;
4. Eclipse Web Tools Platform - проект по созданию общей инструментальной платформы для разработки специализированных J2EE- и Web-ориентированных решений;
5. Eclipse Test & Performance Tools Platform - проект, в рамках которого осуществляется формирование набора технологий для создания инструментов, позволяющих проводить полный цикл тестирования ПО;
6. Business Intelligence & Reporting Tools - проект, в рамках которого решаются задачи бизнес-аналитики и управления отчетами.

10.15 Причинами нынешнего успеха Eclipse называют удачные принципы управления его проектами, участие в проекте многих производителей ПО, поддержку многих платформ, языков программирования и методологий моделирования, а также возможность создать коллективными усилиями реальную альтернативу платформе Microsoft.NET.

Последняя версия Eclipse имеет кодовое название Callisto.

Особое место в планах консорциума Eclipse Foundation занимает развитие набора «каркасов» готовых приложений Eclipse Client Rich Platform (RCP), которые должны стать основой для приложений, предназначенных для конечных пользователей (первые подобные приложения уже доступны пользователям ряда продуктов SAS и IBM).

Учитывая, что сегодня в развитие Eclipse вкладывают средства такие солидные компании, как IBM, Hewlett-Packard, Sybase, SAP, Nokia и Borland, стоит ожидать дальнейшего роста популярности этой платформы и

массового перехода на нее многих производителей инструментальных средств.

10.16 План лабораторной работы «Подсистема управления и мониторинга».

Проектирование модуля мониторинга и управления.

Разработка собственных компонентов управления.

Анализ устойчивости системы.

Тестирование информационной и физической безопасности.

10.17 Контрольные задания для СРМП:

1. Выбор и защита процессов мониторинга, реализуемых приложением;
2. Ответить на контрольные вопросы четвертого модуля;
3. Провести отладку кода спроектированного приложения;
4. Защитить отчет по четвертой лабораторной работе;
5. Защита реализации механизма мониторинга в проекте;
6. Защита работы собственных компонентов управления;
7. Защитить раздел методического обеспечения дисциплины;
8. Тест второго рубежного контроля.

10.18 Контрольные задания для СРМ:

1. Изучить конспект 9,10 лекций;
2. Изучить методические указания к четвертой лабораторной работе;
3. Ответить на примеры тестовых заданий к четвертому модулю;
4. Выбор и настройка компонентов механизма управления;
5. Выбор и настройка компонентов мониторинга;
6. Авторизация пользователей и ограничения доступа;
7. Разработать политику безопасности.

СПИСОК ЛИТЕРАТУРЫ

1 Лодон, Дж. Управление информационными системами : учебник/ Дж. Лодон, К. Лодон; пер. с англ. под ред. Д. Р. Трутнева. - 7-е изд. - М.; СПб.; Нижний Новгород: Питер, 2005. - 910 с.

2 Загайнов И.А. Проектирование информационных систем: Методические указания и задания к курсовому проектированию для студентов специальностей 370440 – «Программное и аппаратное обеспечение вычислительной техники и сетей», 050704 – «Вычислительная техника и программное обеспечение». - Усть-Каменогорск: Издательство ВКГТУ, 2006. – 32с.

3 Загайнов И.А. Проектирование информационных систем Ч1: Методические указания и задания к лабораторным работам для студентов специальности 370440 «Программное и аппаратное обеспечение вычислительной техники и сетей». – Усть-Каменогорск: Издательство ВКГТУ, 2006. – 90с.

4 Калянов Г.Н. Моделирование, анализ, реорганизация и автоматизация бизнес-процессов: учеб. пособие для вузов / Г. Н. Калянов. - М.: Финансы и статистика, 2006. - 238 с.

5 Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии: учеб. пособие/ М.: МГТУ им. Баумана, 2005. - 302 с.

6 Загайнов И.А. Проектирование приложений информационных систем: Методические указания, задания к лабораторным работам, СРС, СРСП для магистрантов специальности 6N0703 – «Информационные системы» / ВКГТУ.- Усть-Каменогорск, 2008. – 32с.

7 Ильин В.В. Моделирование бизнес-процессов. Практический опыт разработчика : произв.-практ. изд. / В. В. Ильин. - М. : ООО "И.Д. Вильямс", 2006. - 166 с.

8 Андрейчиков А.В. Интеллектуальные информационные системы : учебник / А. В. Андрейчиков, О. Н. Андрейчикова. - М. : Финансы и статистика, 2006. - 423 с.

9 Фаронов В.В. Программирование баз данных в Delphi 7: учебный курс /В.В.Фаронов. - М.; СПб.; Нижний Новгород; Воронеж; Ростов н/Д ; Екатеринбург; Самара ; Киев ; Харьков ; Минск : Питер, 2005. - 458 с.

10 Старыгин А.А. XML: разработка Web-приложений: к изучению дисциплины /А.А.Старыгин. - СПб. : БХВ-Петербург, 2003. - 585 с.

11 Базы данных: Учебник / А.Д.Хомоненко, В.М.Цыганков, М.Г.Мальцев; Под ред. А.Д.Хомоненко. - 4-е изд., доп. и перераб. - СПб.: КОРОНАпринт, 2004. - 736 с.

12 www.compress.ru. - КомпьютерПресс 8'2007 Наталия Елманова, Краткое введение в моделирование бизнес-процессов Часть 1. Моделирование процессов и CASE-средства.