

**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования «Амурский государственный университет»**

Кафедра Информационных и управляющих систем
(наименование кафедры)

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

Применение методов моделирования в исследованиях и проектировании
(наименование дисциплины)

Основной образовательной программы по направлению подготовки:

230100.68 «Информатика и вычислительная техника»
по магистерской программе «Компьютерное моделирование»
(код и наименование направления)

Благовещенск 2012

УМКД разработан _____
(степень, звание, фамилия, имя, отчество разработчиков)

Рассмотрен и рекомендован на заседании кафедры

Протокол заседания кафедры от « ____ » _____ 201__ г. № ____

Зав. кафедрой _____ / _____ /
(подпись) (И.О. Фамилия)

УТВЕРЖДЕН

Протокол заседания УМСС _____
(указывается название специальности (направления подготовки))

от « ____ » _____ 201__ г. № ____

Председатель УМСС _____ / _____ /
(подпись) (И.О.Фамилия)

1. Рабочая программа учебной дисциплины

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Амурский государственный университет»

УТВЕРЖДАЮ
Проректор по УР
_____ В.В.Проказин
« ____ » _____ 2012г.

РАБОЧАЯ ПРОГРАММА

ПРИМЕНЕНИЕ МЕТОДОВ МОДЕЛИРОВАНИЯ В ИССЛЕДОВАНИЯХ И ПРОЕКТИРОВАНИИ

Направление подготовки 230100.68 «Информатика и вычислительная техника»
по магистерской программе «Компьютерное моделирование»
Квалификация (степень) выпускника – магистр
Специальное звание – магистр-инженер

Курс – 2

Лекции – 18 (час.)

Практические (семинарские) занятия – нет

Лабораторные занятия – 18 (час.)

Самостоятельная работа – 72 (час.)

Общая трудоемкость дисциплины – 108 (час.), 3 (з.е.)

Курсовая работа (проект) – нет

Составитель – А.В. Бушманов, доцент, канд. техн. наук

Факультет математики и информатики

Кафедра информационных и управляющих систем

Семестр – 3

Экзамен – нет

Зачет – 3

2012 г.

Рабочая программа составлена на основании Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 230100 «Информатика и вычислительная техника» (квалификация (степень) «магистр»)

Рабочая программа обсуждена на заседании кафедры информационных и управляющих систем

«__» _____ 20__ г., протокол № _____

Заведующий кафедрой _____ А.В. Бушманов

Рабочая программа одобрена на заседании учебно-методического совета направления подготовки 230100.68 «Информатика и вычислительная техника»

«__» _____ 20__ г., протокол № _____

Председатель _____ В.В. Еремина

Рабочая программа переутверждена на заседании кафедры информационных и управляющих систем

«__» _____ 20__ г., протокол № _____

Заведующий кафедрой _____ А.В. Бушманов

СОГЛАСОВАНО

Учебно-методическое
управление

«__» _____ 20__ г.

СОГЛАСОВАНО

Председатель учебно-методического
совета факультета

_____ С.Г. Самохвалова
«__» _____ 20__ г.

СОГЛАСОВАНО

Заведующий выпускающей кафедрой
_____ А.В. Бушманов

«__» _____ 20__ г.

СОГЛАСОВАНО

Директор научной библиотеки

_____ Л.А. Проказина
«__» _____ 20__ г.

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ, ЕЕ МЕСТО В УЧЕБНОМ ПРОЦЕССЕ

Целью данной дисциплины является показать, как используются известные методы и модели в научных исследованиях и проектировании информационных систем.

После изучения данной дисциплины магистрант должен знать принципы построения методик моделирования, уметь ими пользоваться, а также широко использовать информационные и вычислительные ресурсы. Изучить особенности методов моделирования. Иметь представление, о возможностях решения научных задач применяя определенную методику. Уметь применять для решения своих задач методы математического и имитационного моделирования. Кроме того, должен освоить основные понятия и положения по системам искусственного интеллекта.

2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО

Дисциплина относится к дисциплинам профессионального цикла вариативной части (М2.В.ОД.1) Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 230100 «Информатика и вычислительная техника» (квалификация (степень) «магистр»).

Для успешного освоения данной дисциплины необходимы знания, умения и навыки, приобретенные в результате освоения дисциплин базовой части общенаучного цикла: Современные проблемы информатики и вычислительной техники (М2.Б.1), Имитационное моделирование (М1.В.ОД.3), Автоматизированные комплексы информационных систем (М2.В.ОД.4) Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки 230100 «Информатика и вычислительная техника» (квалификация (степень) «бакалавр»).

Знания, умения и навыки, приобретенные в результате освоения данной дисциплины необходимы для использования типовых программных продуктов, ориентированных на решение научных, проектных и технологических задач, а также для выполнения научно-исследовательской работы написании магистерской диссертации.

3 КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

- 1) Знать: современные программно-технические средства автоматизированных систем для научных исследований.
- 2) Уметь: применять методы моделирования в исследованиях и проектировании.
- 3) Владеть: методами научного мышления.

4 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 3.0 зачетных единиц, 108 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость в часах				Формы текущего контроля успеваемости Форма промежуточной аттестации
				Лек	Пр	Лаб	Сам	
1	Структура, содержание и задачи курса, его связь с другими дисциплинами.	3	1-2	2	0	2	6	Защита лаб. работы
2	Этапы проектирования и методологические основы мо-	3	3-4	2	0	2	6	Защита лаб. работы

	делирования.		5-6	2	0	2	6	Защита лаб. работы
3	Технические средства моделирования.	3	7-8	2	0	2	6	Защита лаб. работы
			9-10	2	0	2	6	Защита лаб. работы
4	Базы знаний в проектируемых системах.	3	11-12	2	0	2	6	Защита лаб. работы
			13-14	2	0	2	14	Защита лаб. работы
5	Разработка приложений.	3	15-16	2	0	2	12	Защита лаб. работы
			17-18	2	0	2	10	Защита лаб. работы
6	Всего по разделам	3	1-18	18	0	18	72	Зачет

5 СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ

5.1 Лекции

5.1.1 Раздел 1. Структура, содержание и задачи курса, его связь с другими дисциплинами.

Современные программные средства для научных исследований; технические средства автоматизированных систем; перспективные методы исследования и решения профессиональных задач на основе знания мировых тенденций развития вычислительной техники и информационных технологий.

5.1.2 Раздел 2. Этапы проектирования и методологические основы моделирования.

Проблемы моделирования. Мировые концепции управления ИС. Этапы моделирования в ИС. Системы и подсистемы. Методологические основы моделирования и проектирования информационных систем.

5.1.3 Раздел 3. Технические средства моделирования.

Требования к техническим средствам моделирования. Определения общих ограничений целостности, триггеров и хранимых процедур.

5.1.4 Раздел 4. Базы знаний в проектируемых системах.

Базы данных в проектировании. Встраивание операторов языка SQL в программу.

5.1.5 Раздел 5. Разработка приложений.

Моделирование и архитектура информационных приложений. Интегрированные распределенные приложения.

5.2 Лабораторные занятия.

5.2.1 Лабораторная работа 1. Формирование контекстных диаграмм IDEFO и Workflow (IDEF3).

5.2.2 Лабораторная работа 2. Дополнение моделей процессов диаграммами DFD и Workflow.

5.2.3 Лабораторная работа 3. Разработка модели с использованием нотации IDEFX1.

5.2.4 Лабораторная работа 4. Моделирование с использованием CASE средств, оценка и выбор.

5.2.5 Лабораторная работа 5. Модель информационного пространства с помощью CASE-средств верхнего уровня ERWIN.

5.2.6 Лабораторная работа 6. ООП. Декомпозиция. Отношение между классами. Иерархия классов. Объекты и классы. Абстрагирование и обобщение.

5.2.7 Лабораторная работа 7. ООП. Ограниченность доступа. Модульность.

5.2.8 Лабораторная работа 8. Пакеты и классы. Уточнение методов и свойств классов.

5.2.9 Лабораторная работа 9. Создание логической модели.

6 САМОСТОЯТЕЛЬНАЯ РАБОТА

№ п/п	Раздел дисциплины	Форма (вид) самостоятельной работы	Трудоемкость в часах
1	Структура, содержание и задачи курса, его связь с другими дисциплинами.	Выполнение лабораторной работы, оформление отчета.	6
2	Этапы проектирования и методологические основы моделирования.	Выполнение двух лабораторных работ, оформление отчетов.	12
3	Технические средства моделирования.	Выполнение двух лабораторных работ, оформление отчетов.	12
4	Базы знаний в проектируемых системах.	Выполнение двух лабораторных работ, оформление отчетов.	20
5	Разработка приложений.	Выполнение двух лабораторных работ, оформление отчетов, подготовка к сдаче зачета	24

7 МАТРИЦА КОМПЕТЕНЦИЙ

№ п/п	Раздел дисциплины	Компетенции		Общее количество компетенций
		ОК-4	ПК-1	
1	Структура, содержание и задачи курса, его связь с другими дисциплинами.	+		2
2	Этапы проектирования и методологические основы моделирования.		+	2
3	Технические средства моделирования.		+	2
4	Базы знаний в проектируемых системах.		+	2
5	Разработка приложений.		+	2

8 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Образовательный процесс по дисциплине строится на основе комбинации следующих образовательных технологий.

Интегральную модель образовательного процесса по дисциплине формируют технологии методологического уровня: модульно-рейтинговое обучение, технология поэтапного формирования умственных действий, технология развивающего обучения, элементы технологии развития критического мышления.

Реализация данной модели предполагает использование следующих технологий стратегического уровня (задающих организационные формы взаимодействия субъектов образовательного процесса), осуществляемых с использованием определенных тактических процедур:

- лекционные (вводная лекция, информационная лекция, обзорная лекция, лекция-консультация, проблемная лекция);
- лабораторные (углубление знаний, полученных на теоретических занятиях, решение задач);

- тренинговые (формирование определенных умений и навыков, формирование алгоритмического мышления);
- активизации познавательной деятельности (приемы технологии развития критического мышления через чтение и письмо, работа с литературой, подготовка презентаций по темам домашних работ);
- самоуправления (самостоятельная работа студентов, самостоятельное изучение материала).

Рекомендуется использование информационных технологий при организации коммуникации со студентами для представления информации, выдачи рекомендаций и консультирования по оперативным вопросам (электронная почта), использование мультимедиа-средств при проведении лекционных и практических занятий.

9 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

9.1 Оценочные средства для текущего контроля успеваемости

9.1.1 Контрольные вопросы допуска к выполнению лабораторных работ

9.1.2 Отчеты о выполнении индивидуальных вариантов заданий лабораторных работ

9.2 Оценочные средства для промежуточной аттестации

Тест

по дисциплине «Применение методов моделирования в исследованиях и проектировании»

1. Современные системы управления базами данных поддерживают:
 - А) и модель данных «сущность-связь», и реляционную модель данных;
 - Б) модель данных «сущность-связь»;
 - В) реляционную модель данных.
2. Системы реляционного исчисления являются теоретической основой для разработки:
 - А) схемы базы данных;
 - Б) языков запросов к базам данных;
 - В) диаграммы «сущность-связь».
3. Язык запросов SQL близок:
 - А) к исчислению с переменными кортежами;
 - Б) к исчислению с переменными на доменах;
 - В) к языку на основе операторов реляционной алгебры.
4. Укажите правильные операторы SELECT для запроса: выдать имена поставщиков, поставляющих детали с названием «болт» для изделия с названием «рама 02-01»:
 - А) SELECT имя FROM S,P,J,SPJ WHERE S.номер поставщика = SPJ.номер_поставщика AND J.название= «рама02-01» AND P.название= «болт»;
 - Б) SELECT имя FROM S,P,J,SPJ WHERE S.номер поставщика = SPJ.номер_поставщика AND P. номер_детали=SPJ.номер_детали AND J. название= «рама02-01» AND P.название= «болт»;
 - В) SELECT имя FROM S,P,J,SPJ WHERE S.номер поставщика = SPJ.номер_поставщика AND J.номер изделия=SPJ.номер_изделия AND.номер_детали=SPJ.номер_детали AND J .название= «рама 02-01»AND P.название= «болт».
5. Оптимизация запросов к базе данных выполняется:
 - А) автоматически специальным оптимизатором СУБД;
 - Б) вручную специалистом, программирующим запросы;
 - В) конечным пользователем, запускающим запросы на выполнение.
6. Оптимизация запросов к базе данных выполняется с целью:
 - А) уменьшить время выполнения запроса;

- Б) обеспечить правильность выполнения запроса;
 В) уменьшить число записей в результирующей таблице.
7. Пусть заданы отношения $R1=(AB)$, $R2=(CD)$. Оптимизировать следующую формулу реляционной алгебры: $\pi_{A,C}(R1 \times R2)$:
- А) $\sigma_A(R1) \times \sigma_C(R2)$;
 Б) $\pi_A(R1) \times \pi_C(R2)$;
 В) $\pi_C(R2) \times \pi_A(R1)$.
8. Пусть $R1=(AB)$, $R2=(CD)$ и задана формула реляционной алгебры $\pi_A(\sigma_{B>5}(R1)) \times \pi_D(\sigma_{D<3}(R2))$. Написать операторы SELECT, соответствующие подзапросам:
- А) Select D From R1 Where D<3;
 Select A From R2 Where B>5;
 Б) Select A From R1 Where D<3;
 Select D From R2 Where B>5;
 В) Select A From R1 Where B>5;
 Select D From R2 Where D<3.
9. Какие действия выполняются оптимизатором при построении физического плана:
- А) определяются метод выбора записей из исходных таблиц и порядок соединения таблиц подзапросов;
 Б) выполняются преобразования запроса в формулу реляционной алгебры и ее оптимизация;
 В) определяется метод соединения таблиц.
10. В системах на основе модели сервера базы данных по шине сети передаются:
- А) промежуточные данные (индексы, сортируемые данные);
 Б) SQL-операторы, результаты их выполнения и промежуточные данные (индексы, сортируемые данные);
 В) SQL-операторы и результаты их выполнения.
11. При проектировании объектных баз данных используются:
- А) нотация языка UML;
 Б) нотация Чена;
 В) нотация IDEF1x.
12. Модель необходима для того, чтобы:
- А) понять как устроен конкретный объект;
 Б) управлять объектом или процессом;
 В) прогнозировать последствия воздействия на объект.
13. Что является проектной процедурой?
- А) построение сетки;
 Б) выбор или расчет внешних воздействий;
 В) представление результатов моделирования;
 Г) подготовка детализированных чертежей;
 Д) моделирование полей напряжений и деформаций.
14. Основные функции САЕ – систем?
- А) разработка технологических процессов;
 Б) моделирование процессов обработки;
 В) связаны с проектными процедурами анализа, моделирования, оптимизации проектных решений;
 Г) 2-х мерное проектирование;
 Д) 3-х мерное проектирование.

Выполнение заданий в неустановленный срок приводит к снижению рейтинга.

9.3 Учебно-методическое обеспечение самостоятельной работы

9.3.1 Карточки с заданиями и методическими указаниями по выполнению лабораторных работ

9.3.2 СТО СМК 4.2.3.05-2011. Стандарт ФГБОУВПО «АмГУ». Оформление выпускных квалификационных и курсовых работ (проектов), 2011. – 95 с.

10 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература:

10.1 Максимов Н.В. Современные информационные технологии: учеб.:рек. Мин.обр. РФ/ Н.В.Максимов, Т.Л.Партырка, И.И.Попов. –М.:ФОРУМ, 2008.

10.2 Григорьев Ю.А.,Плутенко А.Д. Теория и практика проектирования систем на основе баз данных. Благовещенск: Амурский гос.ун-т, 2008. -395 с.

10.3 Mathematica для студента: [учеб. пособие]/А.М.Половко. –СПб.: БХВ-Петербург, 2007.

б) дополнительная литература:

10.4 Мельников В.П. Информационные технологии: учеб. пособие: рек. УМО/ В.П.Мельников. –М. Академия, 2008. -426 с.

10.5 Пшихопов В.Х. Оценивание и управление в сложных динамических системах: моног./ В.Х.Пшихопов, М.Ю.Медведев. –М.: Физматлит, 2009. -297 с.

10.6 Управляющие системы и автоматика: производственно-практическое издание/ Д.Шмид и др.: пер. с нем. Л.Н.Казанцева. –М.: Техносфера, 2007. -584 с.

10.7 Гагарина Л.Г. Разработка и эксплуатация автоматизированных информационных систем : учеб. пособие: рек. Мин. обр. РФ/ Л. Г. Гагарина, Д. В. Киселев, Е. Л.. Федотова; под ред. Л. Г. Гагариной. -М.: ФОРУМ: ИНФРА-М, 2007.-384 с.

10.8 Соловьев И.В. Проектирование информационных систем. Фундаментальный курс: учеб. пособие: рек. УМО/ И. В. Соловьев, А. А. Майоров. -М.: Академический Проект, 2009.-399 с.

10.9 Блюмин А.М. Проектирование систем информационного, консультационного и инновационного обслуживания: учеб. пособие/ А. М. Блюмин, Л. Т. Печеная, Н. А. Феоктистов. -М.: Дашков и К, 2009.-350 с.

10.10 И.Г.Безуглов. Основы научного исследования: учеб.пособие для аспирантов и студентов-дипломников/Ию.Г.Безуглов, В.В.Лебединский, А.И.Безуглов. –М.: Академический Проект, 2008. -195 с.

в) периодические издания:

10.11 Мехатроника, автоматизация, управление

10.12 Российский журнал биомеханики

10.13 Информатика и системы управления

10.14 Informations Processing & Menagement

г) программное обеспечение и Интернет-ресурсы:

Свободно распространяемое программное обеспечение

№ п/п	Наименование ресурса	Характеристика
1	http://www.iqlib.ru	Интернет библиотека образовательных изданий, в которой собраны электронные учебники, справочные и учебные пособия. Удобный поиск по ключевым словам, отдельным темам и отраслям знаний.
2	http://www.intuit.ru	Интернет-университет информационных технологий, в котором собраны электронные и видео-курсы по отраслям знаний
3	http://amursu.ru	Сайт АмГУ, Библиотека – электронная библиотека АмГУ
4	http://www.biblioclub.ru	Электронная библиотечная система «Уни-

	верситетская библиотека – online»: специализируется на учебных материалах для ВУ-Зов по научно-гуманитарной тематике, а так же содержит материалы по точным и естественным наукам
--	---

11 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

11.1 Лекционная аудитория, оборудованная мультимедийными средствами

11.2 Лаборатории, оборудованные рабочими местами пользователей ЭВМ

12 РЕЙТИНГОВАЯ ОЦЕНКА ЗНАНИЙ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ

Семестровый модуль дисциплины						
№ п/п	Раздел дисциплины	Виды контроля	Сроки выполнения (недели)	Максимальное кол-во баллов	Посещение, активность на занятиях	Максимальное кол-во баллов за модуль
1	Структура, содержание и задачи курса, его связь с другими дисциплинами.	ЛР № 1	1-2	11	1	12
2	Этапы проектирования и методологические основы моделирования.	ЛР № 2	3-4	5	1	12
		ЛР № 3	5-6	5	1	
3	Технические средства моделирования.	ЛР № 4	7-8	5	1	12
		ЛР № 5	9-10	5	1	
4	Базы знаний в проектируемых системах.	ЛР № 6	11-12	5	1	12
		ЛР № 7	13-14	5	1	
5	Разработка приложений.	ЛР № 8	15-16	5	1	12
		ЛР № 9	17-18	5	1	
6	Промежуточная аттестация	зачет	18	6	0	40
Итого						100

2 Краткое изложение программного материала

Раздел 1. Структура, содержание и задачи курса, его связь с другими дисциплинами.

Моделирование (в широком смысле) – это основной метод исследования во всех областях знаний. Методы моделирования используются для оценок характеристик сложных систем и принятия научно обоснованных решений в разных сферах человеческой деятельности. Существующую или проектируемую систему можно эффективно исследовать с помощью математических моделей (аналитических и имитационных) с целью оптимизации процесса функционирования системы. Модель системы реализуется на современных компьютерах, которые в этом случае выступают в качестве инструмента экспериментатора с моделью системы.

В настоящее время моделирование широко используется в сфере управления различными системами, где основными являются процессы принятия решений на основе получаемой информации. Методы моделирования находят применение при исследовании, проектировании, внедрении вычислительных систем (ВС) и автоматизированных систем управления (АСУ).

Выбор метода моделирования и необходимая детализация проекта зависят от этапа разработки системы. На этапах обследования объекта управления (например, промышленно-

го предприятия) и разработки технического задания на проектирование ВС, АСУ модели носят описательный характер и преследуют цель наиболее полно представить в компактной форме информацию об объекте, необходимую разработчику системы.

На этапе разработки технического проекта ВС, АСУ моделирование служит для решения задачи проектирования, т.е. выбора оптимального варианта по определенному критерию при заданных ограничениях из множества допустимых (построение оптимизационных моделей).

На этапе внедрения и эксплуатации ВС, АСУ строятся модели для проигрывания возможных ситуаций для принятия обоснованных и перспективных решений по управлению объектом. Моделирование (имитацию) также широко применяют при обучении и тренировке персонала (деловые игры).

Методы моделирования можно классифицировать на три основные группы: аналитические, численные и имитационные.

1. Аналитические методы моделирования. Аналитические методы позволяют получить характеристики системы как некоторые функции параметров ее функционирования. Таким образом, аналитическая модель представляет собой систему уравнений, при решении которой получают параметры, необходимые для расчета выходных характеристик системы (среднее время обработки задания, пропускную способность и т.д.). Аналитические методы дают точные значения характеристик системы, но применяются для решения только узкого класса задач. Причины этого заключается в следующем. Во-первых, вследствие сложности большинства реальных систем их законченное математическое описание (модель) либо не существует, либо еще не разработаны аналитические методы решения созданной математической модели. Во-вторых, при выводе формул, на которых основываются аналитические методы, принимаются определенные допущения, которые не всегда соответствуют реальной системе. В этом случае от применения аналитических методов приходится отказываться.

2. Численные методы моделирования. Численные методы предполагают преобразование модели к уравнениям, решение которых возможно методами вычислительной математики. Класс задач, решаемых этими методами, значительно шире. В результате применения численных методов получают приближенные значения (оценки) выходных характеристик системы с заданной точностью.

3. Имитационные методы моделирования. С развитием вычислительной техники широкое применение получили имитационные методы моделирования для анализа систем, обладающими в которых являются стохастические воздействия.

Суть имитационного моделирования (ИМ) заключается в имитации процесса функционирования системы во времени, с соблюдением таких же соотношений длительности операций как в системе оригинале. При этом имитируются элементарные явления, составляющие процесс, сохраняется их логическая структура, последовательность протекания во времени. В результате применения ИМ получают оценки выходных характеристик системы, которые необходимы при решении задач анализа, управления и проектирования.

В основу классификации видов моделирования можно положить различные признаки. В зависимости от характера изучаемых процессов в системе моделирование может быть разделено на детерминированное и стохастическое; статическое и динамическое; дискретное и непрерывное.

Детерминированное моделирование применяется для исследования систем, поведение которых можно абсолютно точно предвидеть. Например, путь, пройденный автомобилем, при равноускоренном движении в идеальных условиях; устройство, возводящее в квадрат число и т.п. Соответственно в этих системах протекает детерминированный процесс, который адекватно описывается детерминированной моделью. *Стохастическое (теоретико-вероятностное) моделирование* применяется для исследования системы, состояние которой зависит не только от контролируемых, но и от неконтролируемых воздействий или в ней самой есть источник случайности. К стохастическим системам относятся все системы, которые

включают человека, например, заводы, аэропорты, вычислительные системы и сети, магазины, предприятия бытового обслуживания и т.п.

Статистическое моделирование служит для описания систем в какой-либо момент времени. *Динамическое моделирование* отражает изменение системы во времени (выходные характеристики системы в данный момент времени определяются характером входных воздействий в прошлом и настоящем). Примером динамических систем являются биологические, экономические, социальные системы; такие искусственные системы как завод, предприятие, поточная линия и т.п.

Дискретное моделирование применяют для исследования систем, в которых входные и выходные характеристики измеряется или изменяется во времени дискретно, в противном случае применяют *непрерывное моделирование*. Например, электронные часы, электросчетчик – дискретные системы; солнечные часы, нагревательные приборы – непрерывные системы.

В зависимости от формы представления объекта (системы) можно выделить *мысленное и реальное моделирование*.

При *реальном (натурном) моделировании* исследование характеристик системы проводится на реальном объекте, либо на его части. Реальное моделирование – наиболее адекватно, но его возможности, с учетом особенностей реальных объектов, ограничены. Например, проведение реального моделирования с АСУ предприятия требует, во-первых, создания АСУ; во-вторых, проведения экспериментов с предприятием, что невозможно. К реальному моделированию относят производственный эксперимент и комплексные испытания, которые обладают высокой степенью достоверности. Другой вид реального моделирования – *физическое*. При физическом моделировании исследование проводится на установках, которые сохраняют природу явления и обладают физическим подобием.

Мысленное моделирование применяется для моделирования систем, которые практически не реализуемы на заданном интервале времени. В основе мысленного моделирования лежит создание идеальной модели, основанной на идеальной, мыслительной аналогии. Различают два вида мысленного моделирования: *образное (наглядное) и знаковое*.

При *образном моделировании* на базе представлений человека о реальных объектах создаются различные наглядные модели, отображающие явления и процессы, протекающие в объекте. Например, модели частиц газов в кинетической теории газов в виде упругих шаров, воздействующих друг на друга во время столкновения.

При *знаковом моделировании* описывают моделируемую систему с помощью условных знаков, символов, в частности, в виде математических, физических и химических формул. Наиболее мощный и развитый класс знаковых моделей представляют математические модели.

Математическая модель – это искусственно созданный объект в виде математических, знаковых формул, который отображает и воспроизводит структуру, свойства, взаимосвязи и отношения между элементами исследуемого объекта. Далее рассматриваются только математические модели и соответственно математическое моделирование.

Математическое моделирование – метод исследования, основанный на замене исследуемого объекта-оригинала его математической моделью и на работе с ней (вместо объекта). Математическое моделирование можно разделить на аналитическое (АМ), имитационное (ИМ), комбинированное (КМ).

При АМ создается аналитическая модель объекта в виде алгебраических, дифференциальных, конечно-разностных уравнений. Аналитическая модель исследуется либо аналитическими методами, либо численными методами.

При ИМ создается имитационная модель, используется метод статистического моделирования для реализации имитационной модели на компьютере.

При КМ проводится декомпозиция процесса функционирования системы на подпроцессы. Для тех из них, где это возможно, используют аналитические методы, в противном случае – имитационные.

Ключевые вопросы:

1. Современные программные средства для научных исследований;
2. Технические средства автоматизированных систем;
3. Перспективные методы исследования и решения профессиональных задач на основе знания мировых тенденций развития вычислительной техники и информационных технологий.

Литература:

- 1 Максимов Н.В. Современные информационные технологии: учеб.:рек. Мин.обр. РФ/ Н.В.Максимов, Т.Л.Партырка, И.И.Попов. –М.:ФОРУМ, 2008.
- 2 Григорьев Ю.А., Плутенко А.Д. Теория и практика проектирования систем на основе баз данных. Благовещенск: Амурский гос.ун-т, 2008. -395 с.
- 3 Mathematica для студента: [учеб. пособие]/А.М.Половко. –СПб.: БХВ-Петербург, 2007.

Раздел 2. Этапы проектирования и методологические основы моделирования.

Моделирование является одним из наиболее эффективных методов исследования. Оно заключается в построении и изучении специальных объектов (моделей), свойства которых подобны наиболее важным, с точки зрения исследователя, свойствам исследуемых объектов (оригиналов). В широком смысле моделирование представляет собой научную дисциплину, в которой изучаются методы построения и использования моделей для познания реального мира.

Всякая научная дисциплина, как правило, основывается на исходных понятиях и определениях, позволяющих однозначно понимать язык, применяемый для изложения этой дисциплины. Моделирование, как научная дисциплина, также содержит ряд специальных понятий, которые составляют начало методологических основ этой науки. (Концептуальная часть науки).

Философскую концепцию моделирования составляют теория отражения и теория познания, а формально-методическую основу моделирования составляют теория подобия, теория эксперимента, математическая статистика, математическая логика и научные дисциплины, изучающие те предметные области, которые подлежат исследованию методами моделирования.

В данном разделе рассмотрим основные положения науки о моделировании реально существующих объектов и объектов, подлежащих созданию в будущем. Моделирование этих объектов имеет общую философскую основу, но существенно отличается в части прикладных методов исследования.

Изучение методологических основ моделирования целесообразно начать с рассмотрения философских понятий теории отражения реальной действительности в сознании некоторого субъекта. Это должно обеспечить нам однозначное понимание положений излагаемых далее прикладных теорий. Приведем определения наиболее существенных для моделирования философских понятий. Некоторыми из них мы уже пользовались без определения.

Предметная область - это мысленно ограниченная область реальной действительности или область идеальных представлений, подлежащая описанию (моделированию) и исследованию. Предметная область состоит из объектов, различаемых по каким-либо признакам (свойствам) и находящихся в определенных отношениях между собой, или взаимодействующих каким-либо образом.

В нашем представлении *объект* - это все что мы различаем как нечто целое, реально существующее, или возникающее в нашем сознании и обладающее свойствами, значения которых позволяют нам однозначно распознавать это нечто. Объект, на котором сосредоточивается внимание субъекта с целью исследования, называется *объектом исследования*.

Объекты воспринимаются и различаются субъектами лишь постольку, поскольку они обладают характерными свойствами или способностями. "Свойство" и "способность" также являются весьма важными понятиями в рассуждениях человека.

Свойством называется характерная особенность объекта, которая может быть замечена и оценена субъектом, например, вес, цвет, длина, плотность и тому подобное. Для оценки исследуемого свойства объекта субъект устанавливает определенную меру называемую *показателем свойства*. Для каждого показателя определяется множество значений (уровней, или градаций меры свойства), которые присваиваются ему в результате оценивания свойства. Следовательно, свойство объекта является реальностью, а показатель - субъективной мерой этой реальности, если, конечно, речь идет о реальных объектах.

Показатели всеобщих свойств материальных объектов, таких как пространство и время называются *основными показателями*. Подавляющее большинство показателей других свойств выражаются через показатели этих основных свойств. Поэтому единицы измерения основных показателей служат основой для построения стандартной системы единиц измерения физических величин и называются *основными единицами измерения*.

Выражение показателя некоторого свойства через основные единицы измерения, принятые в определенной стандартной системе единиц (мер), называется *размерностью* данного показателя.

С точки зрения субъекта свойства делятся на внутренние (собственные) свойства объектов, показатели этих свойств называются *параметрами*, и внешние, представляющие собой свойства среды, связанные некоторыми отношениями с параметрами данного объекта. Показатели свойств внешней среды, влияющих на параметры исследуемого объекта, называются *факторами*.

Свойства объектов выявляются только при их взаимодействии, или при сопоставлении объектов друг с другом. Сопоставление (комбинация) значений показателей, наблюдаемых свойств определенных объектов называется *отношением*. Говорят, что отношение *истинно*, если оно подтверждается практическим экспериментом, или логическим выводом. Отношение считается *ложным*, если оно опровергается практической проверкой, или логическим выводом. Иначе отношение считается *неопределенным*. Понятия "истинно", "ложно", "неопределенно" являются логическими значениями любого отношения, результатами субъективной его оценки.

Отношение называется *функциональным* (функцией F), если оно представляет собой однозначное отображение множества X значений показателя некоторого свойства в множество Y значений показателя того же, или иного свойства. Формально это записывают как $F := X \rightarrow Y$, или как $F(X) = Y$, или $F \in X \rightarrow Y$, где " \rightarrow " декартово произведение множеств.

Взаимодействие объектов определяется по результатам изменения значений показателей наблюдаемых свойств этих объектов. Поэтому каждому действию, или взаимодействию, мы присваиваем определенный результат. Это может быть значение, или определенная комбинация значений, показателей свойств взаимодействующих объектов. Действия над значениями показателей свойств объектов, выполняемые по определенным правилам и приводящие к предполагаемому результату, называются *операцией* или *процедурой*.

Значения показателей свойств объектов обозначаются символами из некоторого заранее определенного множества A , называемого *алфавитом*.

Множество объектов, взаимосвязанных между собой определенными отношениями, и выполняющих определенную общую для них целевую функцию или имеющих общее предназначение, называется *системой*.

Система, состоящая из алфавита A , строго определенных множеств отношений (G), операций (Q) и предназначенная для символического описания объектов и систем определенного класса, называется *формальной системой*. Такие системы используются в качестве языков математического моделирования.

Способность, по нашему мнению, есть готовность объекта проявлять определенные свойства в определенных условиях. И наоборот, способность объекта вести себя определенным образом квалифицируется как его свойство. Например, забегая вперед, отметим, что одним из свойств сознания человека является способность применять ранее накопленные зна-

ния для решения возникающих логических проблем. Эта способность называется *интеллектом*.

Энергия является одним из свойств материи, в силу которого все материальные объекты совершают движение в пространстве и времени, находясь в энергетическом взаимодействии и пространственно-временном отношении.

Пространство и *время* также являются всеобщими свойствами материи. Многочисленные эксперименты подтверждают, что все материальные объекты существуют не иначе как в пространстве и во времени. Как известно, значения показателей пространства и времени входят в состав основных единиц измерения всех физических свойств объектов.

Так как все свойства объектов изменяются во времени, то любой набор значений показателей этих свойств относится к определенному значению показателя времени (к моменту времени). Это отношение называется *состоянием* объекта.

Значения показателей свойств меняются с течением времени. В результате этого происходит смена состояний объектов. Акт смены состояний объекта, отнесенный к определенному промежутку времени, называется *событием*, а последовательность взаимосвязанных событий, происходящих на некотором интервале времени, называется *процессом*.

Важным всеобщим свойством материи является способность материальных объектов сохранять вещественные и энергетические результаты (следы) взаимодействия материальных объектов. В философии эта способность называется *отражением*. Высшая форма отражения проявляется в биологических системах, как способность чувственного восприятия окружающей среды, сохранения результатов восприятия и использования их для управления своим поведением.

Часть материально-энергетической системы, предназначенная для восприятия и хранения результатов отражения, с целью воспроизведения и использования их в интересах системы в целом, называется *памятью*. Результаты отражения объектов внешнего мира и внутренних ощущений в памяти человека называются *образами*.

Как правило, чувственные органы человека воспринимают не полный образ наблюдаемого объекта, а только те его свойства, которые данный человек считает наиболее существенными по каким-то причинам. Человек способен присваивать образам символические *имена* из некоторого языка и связывать эти имена определенными логическими (мысленными) отношениями. Сформированная в памяти человека логическая система имен (идентификаторов образов) называется *понятием*.

С другой стороны, понятие можно определить и как некоторую языковую конструкцию, имеющую определенный смысл, т.е. образное содержание.

Система понятий и логических отношений между ними, отражающая какую-нибудь сторону реальной действительности, называется *знаниями*. Каждый субъект обладает памятью и механизмом целенаправленной манипуляции понятиями и знаниями. В целом эта система называется *сознанием*.

Процесс целенаправленной манипуляции знаниями в сознании субъекта называется *мышлением*.

Ключевые вопросы:

1. Проблемы моделирования;
2. Мировые концепции управления ИС;
3. Этапы моделирования в ИС;
4. Системы и подсистемы;
5. Методологические основы моделирования и проектирования информационных систем.

Литература:

- 1 Максимов Н.В. Современные информационные технологии: учеб.:рек. Мин.обр. РФ/ Н.В.Максимов, Т.Л.Партырка, И.И.Попов. –М.:ФОРУМ, 2008.
- 2 Григорьев Ю.А., Плутенко А.Д. Теория и практика проектирования систем на основе баз данных. Благовещенск: Амурский гос.ун-т, 2008. -395 с.
- 3 Mathematica для студента: [учеб. пособие]/А.М.Половко. –СПб.: БХВ-Петербург, 2007.

Раздел 3. Программно-технические средства моделирования.

Большое значение при реализации модели на ЭВМ имеет вопрос правильного выбора языка программирования.

Язык программирования должен отражать внутреннюю структуру понятий при описании широкого круга понятий. Высокий уровень языка моделирования значительно упрощает программирование моделей. Основными моментами при выборе ЯМ является:

- проблемная ориентация;
- возможности сбора, обработки, вывода результатов;
- быстрое действие;
- простота отладки;
- доступность восприятия.

Этими свойствами обладают процедурные языки высокого уровня. Для моделирования могут быть использованы языки Имитационного моделирования (ЯИМ) и общего назначения (ЯОМ).

Более удобными являются ЯИМ. Они обеспечивают:

- удобство программирования модели системы;
- проблемная ориентация.

Недостатки ЯИМ:

- неэффективность рабочих программ;
- сложность отладки;
- недостаток документации.

Основные функции языка программирования:

- управление процессами (согласование системного и машинного времени);
- управление ресурсами (выбор и распределение ограниченных средств описываемой системы).

Как специализированные языки, ЯИМ обладают некоторыми программными свойствами и понятиями, которые не встречаются в ЯОМ. К ним относятся:

Совмещение. Параллельно протекающие в реальных системах S процессы представляются с помощью последовательно работающей ЭВМ. ЯИМ позволяют обойти эту трудность путём введения понятий системного времени.

Размер. ЯИМ используют динамическое распределение памяти (компоненты модели системы M появляются в ОЗУ и исчезают в зависимости от текущего состояния). Эффективность моделирования достигается так же использованием блочных конструкций: блоков, подблоков и т.д.

Изменения. ЯИМ предусматривают обработку списков, отражающих изменения состояний процесса функционирования моделируемой системы на системном уровне. Взаимосвязь. Для отражения большого количества между компонентами модели в статике и динамике ЯИМ включаем системно организованные логические возможности и реализации теории множеств.

Стохастичность. ЯИМ используют специальные программные генерации последовательностей случайных чисел, программы преобразования в соответствующие законы распределения.

Анализ. ЯИМ предусматривают системные способы статистической обработки и анализа результатов моделирования.

Наиболее известными языками моделирования являются SIMULA, SIMSCRIPT, GPSS, SOL, CSL.

Для языков, используемых в задачах моделирования, можно составить классификацию следующего вида.

Классификация языков моделирования.

Язык ДИНАМО используется для решения разностных уравнений. Представление системы S в виде типовой схемы, в которой участвуют как дискретные, так и непрерывные величины, называются комбинированными. Предполагается, что в системе мо-

гут наступать события двух видов: 1) события, от состояния Z_i ; 2) события, зависящие от времени t . При использовании языка GAPS на пользователя возлагается работа по составлению на языке FORTRAN подпрограмм, в которых описываются условия наступления событий, законы изменения непрерывной величины, правил перехода из одного состояния в другое. SIMSCRIPT - язык событий, созданный на базе языка FORTRAN. Каждая модель M_j состоит из элементов, с которыми происходят события, представляющие собой последовательность формул, изменяющих состояние моделируемой системы с течением времени. Работа со списками, определяемые пользователем, последовательность событий в системном времени, работа с множествами. FORSIT - пакет III на языке FORTRAN позволяет оперировать только фиксированными массивами данных, описывающих объекты моделируемой системы. Удобен для описания систем с большим числом разнообразных ресурсов. Полное описание динамики модели можно получить с помощью ПП. SIMULA - расширение языка ALGOL. Блочное представление моделируемой системы. Функционирование процесса разбивается на этапы, происходящие в системном времени. Главная роль в языке SIMULA отводится понятию параллельного оперирования с процессами в системном времени, универсальной обработки списков с процессами в роли компонент. GPSS- интегрирующая языковая система, применяющаяся для описания пространственного движения объектов. Такие динамические объекты в языке GPSS называются транзактами и представляют собой элементы потока. Транзакты "создаются" и "уничтожаются". Функцию каждого из них можно представить как движение через модель M с поочередным воздействием на её блоки. Функциональный аппарат языка образуют блоки, описывающие логику модели, сообщая транзактам, куда двигаться и что делать дальше. Данные для ЭВМ подготавливаются в виде пакета управляющих и определяющих карт, которым составляется по схеме модели, набранной из стандартных символов. Созданная программа GPSS, работая в режиме интерпретации, генерирует и передаёт транзакты из блока в блок. Каждый переход транзакта приписывается к определенному моменту системного времени.

При моделировании предпочтение отдают языку, который более знаком, универсален. Вместе с увеличением числа команд возрастают трудности использования ЯИМ. Получены экспертные оценки ЯИМ по степени их эффективности.

Ключевые вопросы:

1. Требования к техническим средствам моделирования;
2. Определения общих ограничений целостности, триггеров и хранимых процедур;
3. Этапы моделирования в ИС;
4. Системы и подсистемы;
5. Методологические основы моделирования и проектирования информационных систем.

Литература:

- 1 Максимов Н.В. Современные информационные технологии: учеб.:рек. Мин.обр. РФ/ Н.В.Максимов, Т.Л.Партырка, И.И.Попов. –М.:ФОРУМ, 2008.
- 2 Григорьев Ю.А., Плутенко А.Д. Теория и практика проектирования систем на основе баз данных. Благовещенск: Амурский гос.ун-т, 2008. -395 с.
- 3 Mathematica для студента: [учеб. пособие]/А.М.Половко. –СПб.: БХВ-Петербург, 2007.

Раздел 4. Базы знаний в проектируемых системах.

Процесс получения знаний от эксперта (или из каких-либо других источников знаний) и передачи их экспертной системе называют приобретением знаний. Обычно источником знания является эксперт-человек, но могут быть и эмпирические данные и тексты, в которых содержатся сведения об области экспертизы. Процесс приобретения знаний важен и сложен. Важность этого процесса обусловлена тем, что качество и эффективность решения задач экспертной системой определяются качеством и количеством используемых ею знаний. Сложность данного процесса обусловлена как тем, что объем знаний, используемых экспертом, велик, так и тем, что знания эти не полностью осознаются экспертом.

Целесообразно осуществить разбиение процесса приобретения знаний на фазы, отражающие изменение функций участников проектирования (эксперта и инженера по знаниям) и/или экспертной системы: 1) предварительная фаза; 2) начальная фаза; 3) фаза накопления.



Предварительная фаза приобретения знаний характеризуется тем, что экспертной системы еще не существует (отсюда и название фазы). Знания приобретаются инженером по знаниям от эксперта. На этой фазе задача инженера по знаниям состоит в том, чтобы получить от эксперта основные сведения об области экспертизы (основные понятия, отношения, подзадачи и т.п.) и сформировать на их основе общее представление о структуре данных и принципах построения экспертной системы. Эта фаза приобретения знаний выполняется на этапах идентификации, концептуализации и формализации.

На начальной фазе осуществляется наполнение системы знаниями о представлении, т.е. значениями, определяющими организацию, структуру и способ представления базы знаний. В связи с тем, что для определения указанных знаний необходимо владеть основами программирования и детально понимать функционирование проектируемой экспертной системы, введение знаний на начальной фазе может осуществлять только инженер по знаниям, а не эксперт. Начальная фаза осуществляется в ходе первой стадии этапа выполнения.

В ходе фазы накопления осуществляется приобретение основных знаний об области экспертизы. На современном уровне развития приобретение знаний на этой фазе осуществляется экспертом совместно с инженером по знаниям. На фазе накопления решаются следующие задачи: 1) обнаружение неправильности, неполноты или противоречивости знаний, используемых экспертной системой; 2) извлечение новых знаний, устраняющих обнаруженную неправильность, неполноту или противоречивость; 3) преобразование новых знаний в вид, понятный экспертной системе; 4) объединение "новых" знаний со "старыми". Следует отметить, что на данной фазе приобретаются все виды знаний, необходимые для эффективного и качественного функционирования ЭС.

Процесс приобретения знаний является наиболее сложным этапом разработки экспертной системы. Это объясняется тем, что обычно инженер по знаниям плохо разбирается в предметной области, а эксперт не знает программирования. В связи с этим лексика, используемая экспертом, непонятна инженеру по знаниям. Чтобы уточнить и расширить лексику, требуется совместная работа эксперта и инженера по знаниям. Одна из наиболее сложных задач, стоящих перед инженером по знаниям, заключается в том, чтобы помочь эксперту структурировать знания о проблемной области.

Процесс приобретения знаний можно свести к последовательности выполнения следующих задач:

- 1) определяется необходимость модификации (расширения) знаний;
- 2) при необходимости модификации осуществляется извлечение новых знаний, в противном случае процесс приобретения знаний заканчивается;

- 3) новые знания преобразуются в форму, "понятную" экспертной системе;
- 4) знания системы модифицируются, и осуществляется переход к первой задаче.

В выполнении перечисленных задач могут принимать участие эксперт, инженер по знаниям (программист) и экспертная система. В зависимости от того, кто выполняет задачу, можно выделить различные модели приобретения знаний.



В ранних работах по искусственному интеллекту взаимодействие с разрабатываемой системой осуществлял только программист. При разработке системы программисты не отделяли знания (данные) от механизма вывода. В задачу программиста входило освоить с помощью эксперта предметную область и затем при разработке системы выступать в роли и эксперта, и программиста. Недостаточное знание области экспертизы не позволяло программисту гарантировать полноту и непротиворечивость приобретенных знаний. Кроме того, неизбежные модификации системы приводили (при отсутствии разделения системы на базу знаний и механизм вывода) к невозможности сохранить однажды достигнутую непротиворечивость знаний. В этой модели все перечисленные выше задачи по приобретению знаний выполнял программист.

Последующие разработки систем искусственного интеллекта основывались на отделении знаний от программ и оформлении знаний в виде простых информационных структур, называемых базами знаний.

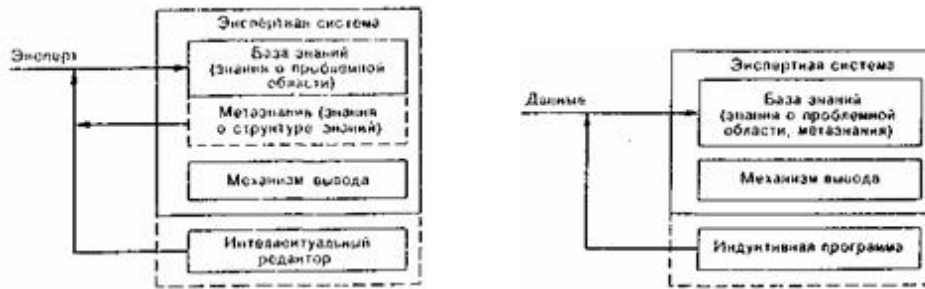
В этом случае эксперт взаимодействует с системой через инженера по знаниям.



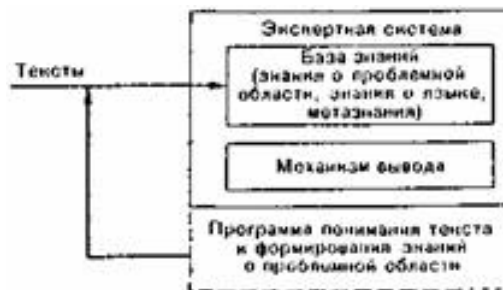
Преимущество данного подхода состоит в том, что база знаний упрощает модификацию знаний. В данной модели первую и вторую задачи приобретения знаний выполняет эксперт с помощью инженера по знаниям, третью задачу выполняет инженер по знаниям, а четвертую - экспертная система. Важным недостатком этого подхода является его большая трудоемкость. Действительно, из четырех задач по приобретению знаний автоматизирована только одна.

Эксперт, минимально сведующий в вопросах программирования, может взаимодействовать с экспертной системой через интеллектуального редактора без посредничества инженера по знаниям. В этой модели интеллектуальный редактор должен обладать развитыми диалоговыми способностями и значительными знаниями о структуре базы знаний (т.е. метазнаниями). Заметим, что интеллектуальный редактор может быть включен в состав экспертной системы. При использовании интеллектуального редактора эксперт (с минимальной помощью инженера по знаниям и экспертной системы) решает первую и вторую задачи приобретения знаний, третья и четвертая задачи выполняются экспертной системой.

ЭС могут приобретать знания аналогично тому, как это делает эксперт-человек. В этом случае индуктивная программа будет анализировать данные, содержащие сведения о некоторой области экспертизы, автоматически формируя значимые отношения и правила, описывающие предметную область. Предполагается, что в базе знаний в явном виде хранятся конкретные факты о проблемной области, а задача индуктивной программы — сделать значимые обобщения. Основным достоинством этого подхода является автоматизация всех перечисленных выше четырех задач по приобретению знаний.



Дальнейшие перспективы развития экспертных систем связываются с приобретением знаний непосредственно из текстов на естественном языке. В данном случае требуется читать обычные печатные тексты (книги, статьи и т.п.) и извлекать из них знания, т.е. понимать текст, схемы, графики и т.п.



Сложность задачи понимания состоит не только в обработке естественного языка, но и в необходимости воссоздать по тексту модель некоторой проблемной области. Эти требования пока превосходят возможности существующих программ понимания, несмотря на то, что в данном случае речь идет об анализе текстов, ограниченных достаточно узкой областью экспертизы. Следует отметить, что приведенные выше методы (модели) приобретения знаний различаются с точки зрения их независимости от эксперта. Методы приведены в порядке возрастания этой независимости, т.е. в порядке увеличивающейся степени автоматизации процесса приобретения знаний.

Статический SQL встраивается в виде препроцессорной обработки в традиционные языки программирования.

Операторы SQL обрабатываются прекомпилятором.

В SQL-операторах могут использоваться переменные из прикладной программы.

Операторы:

DECLARE CURSOR - определяет запрос;

OPEN и CLOSE - начинает и завершает процесс обработки.

Приведем пример приложения, использующего статический SQL:

```
main()
// Включение структуры для обработки ошибок
{ EXEC SQL INCLUDE SQLCA;
// Объявление хост-переменных
// (связи и INTO-переменные)
EXEC SQL BEGIN DECLARE SECTION;
int NumIndID;
int NumID; // Эти переменные
// указываются после символа :
char Sal[10];
EXEC SQL END DECLARE SECTION;
// Обработка ошибок
EXEC SQL WHENEVER SQLERROR GOTO err_1;
```

```

EXEC SQL WHENEVER NOT FOUND GOTO err_2;
// Запрос параметров
printf ("Type individual number: ");
scanf ("%d",&NumIndID);
// Выполнение SQL запроса
EXEC SQL SELECT NumID, Sal FROM tbl1
WHERE NumIndID =: NumIndID
INTO :NumID, :Sal;
// Отображение результата
std::cout<< "Number: "<<NumID;
exit(0);
err_1:
std::cout<<" SQLERROR";
exit(1);
err_2:
std::cout<<" NOT FOUND";
exit(1); }

```

Типы данных преобразуются автоматически для каждой СУБД.

СУБД возвращает информацию об ошибках через специальные переменные: структуру SQL Communication AREA (SQLCA), переменную SQLSTATE или SQLCODE.

Теперь рассмотрим более подробно синтаксис языка встроенного SQL.

Фактически, чтобы приложение могло обращаться к базе данных, существуют четыре основных варианта:

- встраивание в код некоторого языка программирования SQL-операторов (статический SQL);
- формирование в процессе выполнения программы на некотором языке программирования кода SQL-операторов и дальнейшего их выполнения (динамический SQL);
- вызов из программ, написанных на других языках программирования, SQL-модулей, которые представляют собой код на языке SQL;
- использование API (Application Programming Interface), позволяющего реализовывать работу с базой данных через предоставляемый набор функций. API может быть целевым, предоставленным производителем коммерческой СУБД для работы именно с этой базой данных, или межплатформенным, реализующим унифицированные средства доступа к СУБД различных производителей. К такому API относятся ODBC (Open DataBase Connectivity) и SQL/CLI (SQL Call Level Interface).

Основная программа

Основной программой, или HOST-программой, называется программа, в которую встраиваются SQL-операторы.

Встраиваемый SQL-оператор указывается после фразы EXEC SQL.

Стандартом SQL-92 предусмотрена возможность встраивания SQL-операторов в следующие языки программирования: C, Pascal, Java (SQLJ), Ada, Cobol, Fortran, PL/1, M.

Ключевые вопросы:

1. Базы данных в проектировании;
2. Встраивание операторов языка SQL в программу.

Литература:

- 1 Максимов Н.В. Современные информационные технологии: учеб.:рек. Мин.обр. РФ/ Н.В.Максимов, Т.Л.Партырка, И.И.Попов. –М.:ФОРУМ, 2008.
- 2 Григорьев Ю.А., Плутенко А.Д. Теория и практика проектирования систем на основе баз данных. Благовещенск: Амурский гос.ун-т, 2008. -395 с.
- 3 Mathematica для студента: [учеб. пособие]/А.М.Половко. –СПб.: БХВ-Петербург, 2007.

Раздел 5. Разработка приложений.

Сегодня трудно найти компанию, производящую компьютерную технику или программное обеспечение, которая не поставляла бы средства разработки информационных приложений, компоненты информационных систем или законченные готовые к использованию решения. Одним из немногих исключений из этого правила является компания InformixSoftware, которая (по крайней мере, пока) производит только базовые программные системы и средства разработки, но не предлагает собственных готовых приложений, оставляя их создание своим партнерам.

Вообще, в компьютерном мире понятие стороннего поставщика (third-partycompany) имеет очень большое значение, поскольку значительная часть прикладного программного обеспечения производится именно небольшими независимыми софтверными компаниями.

Oracle предлагает приложения (естественно, основанные на использовании серверов баз данных Oracle) для использования в следующих предметных областях:

- управление финансами (распределенные мультивалютные бухгалтерские системы, системы финансового планирования, системы финансового анализа и т.д.);
- управление человеческими ресурсами (автоматизация процедуры найма на работу, планирование обучения персонала, удержания его в компании, продвижения по службе и т.д.);
- автоматизация производства (автоматизированные гибридные производственные системы, системы поддержки новых методов инженерии, системы планирования и моделирования и т.д.);
- системы поддержки ведения проектов (организация и мониторинг процесса проектирования, отслеживание проектных расходов и т.д.);
- автоматизация деятельности по продажам (анализ рынка, управление рыночной деятельностью, управление продажами);
- системы сетевого планирования.

Другая картина представлена компанией HewlettPackard. Как один из ведущих производителей вычислительной техники, компания в основном зарабатывает деньги за счет продажи компьютеров. Но с другой стороны, для увеличения спроса желательно предлагать заказчикам готовые решения. Это и делает компания. В кооперации с компаниями-разработчиками программного обеспечения (в частности, с Oracle) разрабатываются законченные аппаратно-программные конфигурации, пригодные для использования в различных предметных областях.

В частности, имеются решения HP для применения в следующих сферах:

- проверка качества окружающей среды;
- управление финансами;
- автоматизация государственной деятельности (федерального уровня, уровня штата, уровня города и т.д.);
- автоматизация деятельности в области здравоохранения;
- системы фармацевтического анализа;
- автоматизация производства;
- управление розничной торговлей и т.д.

Компания IBM, являясь крупнейшей (и одной из старейших) в мире производителей компьютеров и программного обеспечения, предлагает компоненты информационных приложений и готовые решения для использования во всех сферах человеческой деятельности. Естественно, как и в случае HewlettPackard, эти решения опираются на аппаратные средства IBM (и базовое программное обеспечение этой же компании).

Совершенно необъятное число информационных приложений предлагает компания Microsoft. Следуя своей технологии компонентной организации программных продуктов, компания Microsoft предлагает компоненты для горизонтального рынка - набор полуфабрикатов, пригодных для построения бухгалтерских систем, систем документооборота, систем планирования ресурсов и т.д., а также готовые продукты для вертикального рынка - финан-

совые системы, здравоохранение, автоматизация работы гостиниц, издательские системы и т.д.

Мы уже отмечали, что большинство прикладных программных продуктов производится небольшими самостоятельными soft-верными компаниями. Такие компании в силу ограниченности своих ресурсов часто не могут обеспечить разработку и сопровождение продуктов на разных платформах и сосредотачиваются на поддержке платформ одного поставщика. Компании этого рода недаром называют "сторонними поставщиками", поскольку они ведут свою собственную политику, сами рекламируют и продают свою продукцию и не связаны со своими "старшими братьями" партнерскими отношениями. Тем не менее, основные поставщики также заинтересованы в том, чтобы их потенциальные заказчики знали о существовании продуктов третьих компаний, поскольку это является дополнительным доводом при принятии решения о покупке. Большинство крупных компаний собирает и распространяет информацию о доступности продуктов сторонних поставщиков.

Например, компания Sun Microsystems каждый год издает специальный каталог под названием "Catalyst", содержащий исключительно информацию о продуктах независимых фирм, работающих на платформах Sun. Catalyst обычно имеет объем более 1000 страниц. Прикладные продукты разбиты на предметно-ориентированные категории. По поводу каждого продукта приводятся его краткая характеристика и адрес и другая контактная информация производителя.

Ключевые вопросы:

1. Моделирование и архитектура информационных приложений;
2. Интегрированные распределенные приложения.

Литература:

- 1 Максимов Н.В. Современные информационные технологии: учеб.:рек. Мин.обр. РФ/ Н.В.Максимов, Т.Л.Партырка, И.И.Попов. –М.:ФОРУМ, 2008.
- 2 Григорьев Ю.А., Плутенко А.Д. Теория и практика проектирования систем на основе баз данных. Благовещенск: Амурский гос.ун-т, 2008. -395 с.
- 3 Mathematica для студента: [учеб. пособие]/А.М.Половко. –СПб.: БХВ-Петербург, 2007.

2. Методические материалы к выполнению лабораторных занятий

Понятие CASE является аббревиатурой и расшифровывается следующим образом: Computer Aided Software Engineering, что в переводе с английского на русский переводится примерно как «Разработка программного обеспечения с помощью компьютера».

В соответствии с ГОСТ 19781-90 «Программное обеспечение – совокупность программ системы обработки информации и программных документов, необходимых для их эксплуатации». То есть, программное обеспечение – это программы, используемые в компьютере вместе с их описанием, или разработка программ, используемых в компьютере с помощью компьютера.

Термин CASE (Computer Aided Software Engineering) используется в настоящее время в весьма широком смысле. Первоначальное значение термина CASE, ограниченное вопросами автоматизации разработки только лишь программного обеспечения (ПО), сейчас приобрело новый смысл, охватывающий процесс разработки сложных информационных систем в целом.

Теперь под термином CASE-средства понимаются программные средства, поддерживающие процессы создания и сопровождения информационных систем, включая анализ и формулировку требований, проектирование прикладного программного обеспечения (приложений) и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы.

CASE-средства вместе с системным программным обеспечением и техническими средствами образуют полную среду разработки информационных систем.

Появлению CASE-технологии и CASE-средств предшествовали исследования в области методологии программирования. Программирование обрело черты системного подхода с

разработкой и внедрением языков высокого уровня, методов структурного и модульного программирования, языков проектирования и средств их поддержки, формальных и неформальных языков описаний системных требований и спецификаций и т.д. Кроме того, появлению CASE-технологии способствовали и такие факторы, как:

- подготовка аналитиков и программистов, восприимчивых к концепциям модульного и структурного программирования;

- широкое внедрение и постоянный рост производительности компьютеров, позволившие использовать эффективные графические средства и автоматизировать большинство этапов проектирования;

- внедрение сетевой технологии, предоставившей возможность объединения усилий отдельных исполнителей в единый процесс проектирования путем использования разделяемой базы данных, содержащей необходимую информацию о проекте.

CASE-технология представляет собой методологию проектирования информационных систем, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения информационных систем и разрабатывать приложения в соответствии с информационными потребностями пользователей.

Большинство существующих CASE-средств основано на методологиях структурного (в основном) 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.

Согласно обзору передовых технологий (Survey of Advanced Technology), составленному фирмой Systems Development Inc. в 1996 г. по результатам анкетирования более 1000 американских фирм, CASE-технология в настоящее время попала в разряд наиболее стабильных информационных технологий (ее использовала половина всех опрошенных пользователей более чем в трети своих проектов, из них 85% завершились успешно). Однако, несмотря на все потенциальные возможности CASE-средств, существует множество примеров их неудачного внедрения, в результате которых CASE-средства становятся «полочными» программными обеспечениями. В связи с этим необходимо отметить следующее:

- CASE-средства не обязательно дают немедленный эффект; он может быть получен только спустя какое-то время;

- реальные затраты на внедрение CASE-средств обычно намного превышают затраты на их приобретение;

- CASE-средства обеспечивают возможности для получения существенной выгоды только после успешного завершения процесса их внедрения.

Ввиду разнообразной природы CASE-средств было бы ошибочно делать какие-либо безоговорочные утверждения относительно реального удовлетворения тех или иных ожиданий от их внедрения. Можно перечислить следующие факторы, усложняющие определение возможного эффекта от использования CASE-средств:

- широкое разнообразие качества и возможностей CASE-средств;

- относительно небольшое время использования CASE-средств в различных организациях и недостаток опыта их применения;

- широкое разнообразие в практике внедрения различных организаций;

- отсутствие детальных метрик и данных для уже выполненных и текущих проектов;

- широкий диапазон предметных областей проектов;

- различная степень интеграции CASE-средств в различных проектах.

Вследствие этих сложностей доступная информация о реальных внедрениях крайне ограничена и противоречива. Она зависит от типа средств, характеристик проектов, уровня сопровождения и опыта пользователей. Некоторые аналитики полагают, что реальная выгода от использования некоторых типов CASE-средств может быть получена только после одного или двухлетнего опыта. Другие полагают, что воздействие может реально проявиться в фазе

эксплуатации жизненного цикла информационной системы, когда технологические улучшения могут привести к снижению эксплуатационных затрат.

Для успешного внедрения CASE-средств организация должна обладать следующими качествами:

- Технология. Понимание ограниченности существующих возможностей и способность принять новую технологию;
- Культура. Готовность к внедрению новых процессов и взаимоотношений между разработчиками и пользователями;
- Управление. Четкое руководство и организованность по отношению к наиболее важным этапам и процессам внедрения.

Если организация не обладает хотя бы одним из перечисленных качеств, то внедрение CASE-средств может закончиться неудачей независимо от степени тщательности следования различным рекомендациям по внедрению.

Для того, чтобы принять взвешенное решение относительно инвестиций в CASE-технологию, пользователи вынуждены производить оценку отдельных CASE-средств, опираясь на неполные и противоречивые данные. Эта проблема зачастую усугубляется недостаточным знанием всех возможных «подводных камней» использования CASE-средств. Среди наиболее важных проблем выделяются следующие:

- достоверная оценка отдачи от инвестиций в CASE-средства затруднительна ввиду отсутствия приемлемых метрик и данных по проектам и процессам разработки программного обеспечения;
- внедрение CASE-средств может представлять собой достаточно длительный процесс и может не принести немедленной отдачи. Возможно даже краткосрочное снижение продуктивности в результате усилий, затрачиваемых на внедрение. Вследствие этого руководство организации-пользователя может утратить интерес к CASE-средствам и прекратить поддержку их внедрения;
- отсутствие полного соответствия между теми процессами и методами, которые поддерживаются CASE-средствами, и теми, которые используются в данной организации, может привести к дополнительным трудностям;
- CASE-средства зачастую трудно использовать в комплексе с другими подобными средствами. Это объясняется как различными парадигмами, поддерживаемыми различными средствами, так и проблемами передачи данных и управления от одного средства к другому;
- некоторые CASE-средства требуют слишком много усилий для того, чтобы оправдать их использование в небольшом проекте, при этом, тем не менее, можно извлечь выгоду из той дисциплины, к которой обязывает их применение;
- негативное отношение персонала к внедрению новой CASE-технологии может быть главной причиной провала проекта.

Пользователи CASE-средств должны быть готовы к необходимости долгосрочных затрат на эксплуатацию, частому появлению новых версий и возможному быстрому моральному старению средств, а также постоянным затратам на обучение и повышение квалификации персонала.

Несмотря на все высказанные предостережения и некоторый пессимизм, грамотный и разумный подход к использованию CASE-средств может преодолеть все перечисленные трудности. Успешное внедрение CASE-средств должно обеспечить такие выгоды как:

- высокий уровень технологической поддержки процессов разработки и сопровождения программного обеспечения;
- положительное воздействие на некоторые или все из перечисленных факторов: производительность, качество продукции, соблюдение стандартов, документирование;
- приемлемый уровень отдачи от инвестиций в CASE-средства.

Сегодня рынок предоставляет широкий спектр CASE-средств, выбор которых индивидуален и зависит от конкретной работы с различными видами программного обеспечения.

По мнению Вендерова А., рынок программного обеспечения располагает следующими наиболее развитыми CASE-средствами:

- Vantage Team Builder (Westmount I-CASE);
- Designer/2000;
- Silverrun;
- ERwin+BPwin;
- S-Designor;
- CASE.Аналитик;
- Rational Rose.

Разумеется, этот список не является полным. CASE-средств намного больше и, что не удивительно, в различных публикациях и критических обзорах каждый автор называет самым лучшим и универсальным именно то средство разработки программного обеспечения, которым пользуется сам. Причины этого понятны, поскольку в некоторых случаях – это реклама, а в других – авторы, используя конкретное CASE-средство и хорошо его зная, естественно, определяют его как «самое-самое».

Рассмотрим некоторые из CASE-средств - SELECT (Select Software Tools), System Architect (Popkin Software & Systems), CASE/4/0 (microTOOL GmbH), Westmount I-CASE Yourdon (Westmount Technology B.V. & CADRE Technologies), CA ERwin 4.0 (Computer Associates).

В настоящее время фирмой Aonix представлена целая серия продуктов SELECT под маркой Select Component Factory, которая разрабатывается подразделением Select Business Solutions. Эта серия состоит из следующих продуктов:

- Select Component Manager (для работы с компонентами);
- Select JSync и Select VBSync (для работы с кодом);
- Reviewer for Select Enterprise (для контроля завершенности, точности, стандартов и т.п.);
- Select Enterprise (для определения бизнес-логики).

Select Component Factory - мощный инструмент для компонентной разработки.

Select Enterprise

Select Enterprise - основное средство моделирования. Характерные особенности этого средства приведены ниже:

- единое местонахождение всех модулей системы (это позволяет легко находить эти модули и раздавать права на них при командной разработке);
- оперативный контроль изменений (при изменении одного модуля изменяются все модули, где он используется);
- поддержка работы в многопользовательском режиме;
- гибкая настройка прав доступа к системе и её компонентам;
- высокая производительность и масштабируемость;
- гибкость настроек под конкретного пользователя;
- поддержка OLE Automation;
- поддержка резервного копирования, восстановления при сбоях;
- интеграция в Windows;
- встроенный контроль версий.

Select Component Manager

Select Component Manager - средство для работы с компонентами. В его функции входит публикация компонентов, управление ими и их использованием.

Select JSync

Select JSync - это синхронизатор кода, который имеет возможность настройки двухсторонней интеграции Select Enterprise и среды Java.

Select VBSync

Select VBSync - это синхронизатор кода, который имеет возможность настройки двусторонней интеграции Select Enterprise и среды Visual Basic.

Reviewer for Select Enterprise

Reviewer for Select Enterprise - средство для проверки построенной модели на соответствие базовым стандартам. Средство базируется на UML. Оно позволяет провести оценку точности, завершенности и некоторых важных параметров модели.

Общая характеристика

Помимо Select Component Factory фирма Aonix разработала еще ряд компонент, позволяющих расширить возможности базового продукта. В настоящее время из всех рассматриваемых продуктов этот наиболее активно продвигается на рынке. К достоинствам также можно отнести гибкость решения (в том числе и с экономической точки зрения) за счет компонентной структуры.

Средство System Architect разработано фирмой Popkin Software&Systems. Основными задачами продукта являются:

- моделирование бизнес-процессов;
- объектно-ориентированное моделирование;
- компонентная разработка;
- моделирование реляционных БД;
- структурный анализ и дизайн.

System Architect представляет собой универсальное CASE-средство, позволяющее осуществить не только проектирование данных, но и структурное моделирование. Средство проектирования данных и создания ER-диаграмм является одной из составных частей этого продукта.

Возможности продукта

Этот продукт поддерживает СУБД практически всех ведущих производителей, включая Oracle (Oracle 8), Sybase, DB2, SQL Server, IBM (AS400, DB2), Informix, Sybase, Access, dBASE, Paradox и др.

В процессе логического моделирования можно проверить модель на соответствие правилам проектирования данных (например, на соответствие ее первой, второй или третьей нормальным формам). При генерации DDL-скрипта можно сгенерировать триггеры (в том числе и нестандартные).

Все компоненты System Architect позволяют документировать процесс работы над проектом, включая техническое задание, план тестирования и др. Модели System Architect, как и в случае других CASE-средств, можно сохранять в репозитории. Однако в отличие от традиционных репозитариев, обладающих более или менее стандартной структурой хранимых данных, репозитарий System Architect является настраиваемым – к сохраняемым объектам можно добавлять дополнительные свойства, определенные пользователем.

System Architect обладает встроенным Visual Basic for Application, что позволяет создавать разнообразные решения на базе этого продукта, включая автоматическую генерацию моделей и проектной документации.

System Architect позволяет генерировать код клиентских приложений для Visual Basic, Delphi и PowerBuilder (на сегодняшний день это практически единственное CASE-средство, поддерживающее генерацию кода Delphi), классы C++, а также код и текстовые экранные формы COBOL.

Общая характеристика

В целом продукт производит хорошее впечатление. Несмотря на некоторые откровенные глупости в интерфейсе, сама идея реализации интерфейса неплохая. Уже после первой загрузки становится интуитивно понятно куда надо нажимать и что редактировать сейчас. Положительный момент - четкое визуальное деление, на каком этапе и кем используется каждая диаграмма или модель. Отражена также взаимосвязь модели.

К недостаткам можно отнести чрезмерное количество параметров у моделей. Может, это и обеспечивает дополнительную гибкость, но уж очень сильно приближает к языкам программирования.

Средство CASE/4/0 разработано фирмой microTOOL. Его назначение позиционируется создателями как разработка и модернизация структурированных программных систем. Продукт основан на репозитории с реляционной структурой.

Все операции в продукте разделены на относящиеся к системному анализу (построение функциональной структуры, информационных потоков, структуры данных, ER-модели) и к системному проектированию (создание структуры модулей, структуры типов, построение реляционной модели).

Возможности продукта

Пакет CASE/4/0 включает в себя структурные средства системного анализа, проектирования и программирования и обеспечивает поддержку всего жизненного цикла разработки вплоть до сопровождения, основанную на сетевом репозитории, контролирующем целостность проекта и поддерживающую согласованную работу всех участников проекта (системных аналитиков, проектировщиков, программистов). Анализ базируется на классической структурной методологии Уорда-Меллора, являющейся расширением подхода Йодана/Де Марко с целью его ориентации на разработку систем реального времени, проектирование основано на подходе Джексона.

Для целей анализа и проектирования используются следующие типы диаграмм:

- древовидные диаграммы функциональной декомпозиции;
- диаграммы потоков данных;
- диаграммы переходов состояний;
- диаграммы «сущность-связь»;
- структурные карты Джексона.

Помимо графических редакторов перечисленных диаграмм и репозитория, основными компонентами пакета являются:

- дизайнер диалогов для моделирования интерфейса пользователя;
- средства разработки на Cobol, C/C++, Visual Basic;
- синтаксически-ориентированные редакторы кодов;
- средства генерации документов.

Пакет состоит из двух компонентов: клиентской части, устанавливаемой на рабочих местах разработчиков (MS Windows), и интегрированного сетевого репозитория, устанавливаемого на сервере (Novell, MS Windows, HP Unix, Sinix, IBM OS/2, IBM AIX).

Общая характеристика

В отличие от других продуктов, CASE/4/0, не перегружен графикой и обладает минимально-необходимым количеством картинок и цветов. За счет этого он сильно обогнал конкурентов в производительности. По части функциональности его рекомендуется использовать начинающим, либо тем, кому не требуется излишняя подробность и загроможденность модели.

Итак, положительные моменты: быстрота работы с продуктом, высокая производительность, простой к пониманию интерфейс, необходимый в общем и среднем случае набор функций. Существенных недостатков сразу не видно.

VantageTeam Builder фирмы CADRE (известный ранее как Westmount I-CASE Yourdon) - одно из наиболее мощных на рынке средств разработки информационных систем. Основанный на структурном подходе, он позволяет вести разработку параллельно по трем направлениям - построение модели данных, разработка модели поведения системы (функциональной модели) и проектирование интерфейса системы. VantageTeam Builder позволяет вести разработку приложений для различных СУБД, в том числе для Sybase, Oracle и Ingres.

Общая характеристика CADRE VantageTeam Builder

VantageTeam Builder представляет собой интегрированный программный продукт, обеспечивающий выполнение следующих функций:

- графическое проектирование архитектуры системы (проектирование состава и связи вычислительных средств;
- распределения задач системы между вычислительными средствами;
- моделирование отношений типа «клиент-сервер»;
- анализ использования мониторов транзакций и особенностей функционирования систем в реальном времени);
- проектирование диаграмм потоков данных, «сущность-связь», структур данных, структурных схем программ и последовательностей экранных форм;
- генерация кода программ на 4GL целевой СУБД с полным обеспечением программной среды и генерация SQL-кода для создания таблиц БД, индексов, ограничений целостности и хранимых процедур;
- программирование на языке C со встроенным SQL;
- управление версиями и конфигурацией проекта;
- генерация проектной документации по стандартным и индивидуальным шаблонам
- экспорт и импорт данных проекта в формате CDIF.

VantageTeam Builder можно использовать в конфигурации «клиент-сервер», при этом база проектных данных может располагаться на сервере, а рабочие места разработчиков могут быть клиентами.

VantageTeam Builder функционирует на всех основных UNIX-платформах и VMS. В качестве целевой СУБД могут использоваться ORACLE, Informix, Sybase и Ingres.

В качестве отдельного продукта поставляется интерфейс Westmount-Uniface Bridge, обеспечивающий совместное использование двух систем в рамках единой технологической среды проектирования (при этом схемы БД, структурные схемы программ и последовательности экранных форм непосредственно в режиме on-line, без создания каких-либо файлов экспорта- импорта, переносятся в репозиторий Uniface, и, наоборот, прикладные модели, сформированные средствами Uniface, могут быть перенесены в репозиторий VantageTeam Builder. Возможные рассогласования между репозиториями двух систем устраняются с помощью специальной утилиты).

VantageTeam Builder как инструмент аналитика и дизайнера

В соответствии с реализованным в VantageTeam Builder методом, работа над проектом разбивается на четыре фазы: Анализ, Архитектура (в соответствии с отечественными традициями Глобальное проектирование), Дизайн (соответственно, Детальное проектирование) и Программирование.

программных переменных. В фазе Архитектуры начинается определение принципов построения интерфейса системы с использованием диаграмм Последовательности экранных форм. Они позволяют указывать как условия переходов между экранными формами, так и выполняемые при этом действия.

Достоинства и недостатки

Скорость разработки простых проектов при использовании VantageTeam Builder оказывается вполне сопоставимой с такими популярными средствами разработки приложений, как PowerBuilder и Delphi.

Применение стандартных библиотек позволяет использовать VantageTeam Builder как средство ускоренной разработки приложений с однотипным интерфейсом и функциональными возможностями для различных таблиц. Однако, есть ряд недостатков, существенно снижающих достоинства применения библиотек. Одним из них является выбранный для разработки шаблонов эталон интерфейса и программного кода. Он не поддерживает альтернативных ключей, что весьма затрудняет использование сериальных полей; не обеспечивает возможности проведения групповых операций с записями; в части интерфейса довольно старомоден и не нравится пользователям, уже привыкшим к графике.

Другим серьезным недостатком являются слишком большие трудности, возникающие при необходимости доработки автоматически генерируемого текста.

С учетом соотношения цен вряд ли целесообразно использовать VantageTeam Builder при разработке относительно простых приложений. В более сложных же проектах неизбежно возникает проблема доработки автоматически сгенеренных кодов программ с целью выполнения специфических требований заказчика касательно работы с конкретными таблицами. Причем чем крупнее проект, тем больше таких специфических таблиц и тем лучше эти изменения должны быть задокументированы.

ERwin является инструментом, который дает возможность логично продолжить разработку информационной системы в рамках единой технологической цепочки, создавая тесно интегрированную с функциональной моделью модель данных, генерируя соответствующую структуру на любом из поддерживаемых им серверов баз данных (а таковых более двадцати) и автоматически генерируя код клиентского приложения на PowerBuilder или Visual Basic.

ERwin и BPwin были созданы компанией Logic Works в начале 90-х годов. Как известно, в конце 90-х Logic Works слилась с компанией PLATINUM Technology, которая, в свою очередь, была поглощена компанией Computer Associates. Возможно, вследствие приоритетности проблем, связанных с реорганизацией компаний, новые версии ERwin и BPwin достаточно долго не выпускались, хотя в 1999-2000 годах фирма PLATINUM Technology – Computer Associates выпустила три дополнения (Service Pack) к BPwin, ERwin и ModelMart, которые, вопреки сложившейся практике, помимо исправления некоторых ошибок и недочетов включали существенные расширения функциональности.

И, наконец, в начале 2001 года фирма Computer Associates выпустила новые версии CASE-средства разработки информационных систем - Paradigm Plus 4.0, BPwin 4.0 и ERwin 4.0. Ниже рассматриваются особенности и новые функциональные возможности CASE-средства проектирования баз данных ERwin 4.0.

Наиболее заметным изменением является поддержка новой технологии разработки приложений. В прежних версиях ERwin модель неизменно включала два уровня представления данных – логический и физический. Логический уровень не зависел от СУБД и представлял собой абстрактный взгляд на данные. Физический уровень, по существу, являлся отображением системного каталога конкретной СУБД (или DDL-скрипта). Синхронизация уровней производилась автоматически при переходе от логического к физическому уровню. Для создания физического уровня модели данных для другой СУБД требовалось переключение на другой сервер, что приводило к потере физической модели для предыдущей СУБД. Разумеется, можно было сохранить физический уровень представления в отдельной модели, но в этом случае становилась невозможной синхронизация между логическим и физическим уровнями разных моделей.

ERwin 4.0 поддерживает новую идеологию разработки схемы базы данных. Он позволяет создать как чисто логическую, так и чисто физическую модель, а также синхронизировать модели между собой. Это дает возможность разрабатывать крупные гетерогенные информационные системы, создавая независимо одну логическую и несколько физических моделей (для различных СУБД). Версия ERwin 4.0 сохранила доступность и прежнюю технологию, при которой логический и физический уровни содержатся в одной модели.

Существенно изменен интерфейс ERwin: панели инструментов стали перемещаемыми, изменены структура меню и внешний вид редакторов свойств объектов, появился мощный навигатор моделей – Model Explorer.

Model Explorer имеет три закладки: в первой показываются все объекты модели, во второй – предметные области, в третьей – домены. Щелчок мышью по соответствующему объекту обеспечивает быстрый переход к нему. Кроме того, Model Explorer позволяет создавать, редактировать, копировать и перемещать объекты в модели. Model Explorer является контекстным инструментом, его содержимое зависит от уровня модели (логического или физического), нотации (IDEF1X, IE или Dimensional) и выбранной СУБД.

Новая палитра рисования позволяет размещать на диаграмме графические объекты, не предусмотренные синтаксисом IDEF1X или IE, и облегчает читаемость модели.

ERwin 4.0 содержит палитру трансформации объектов, которая дает возможность преобразовывать одни объекты на логическом уровне в другие на физическом. Каждая кнопка палитры трансформации вызывает гид, с помощью которого можно автоматически преобразовывать объекты модели.

В ERwin 4.0 предусмотрена возможность установления связей с миграцией альтернативных ключей. В результате установления такой связи в дочернюю сущность мигрируют не первичные, а альтернативные ключи.

Изменен интерфейс редакторов прямого и обратного проектирования. Список СУБД для прямого и обратного проектирования существенно расширен, в частности новая версия ERwin 4.0 поддерживает DB2 UDB 6.1, DB2 OS390 6, Access 2000, Informix 9.2x и др.

ERwin 4.0 поддерживает стандарты именования объектов модели. Утилита ERwin Naming Standards Editor позволяет описать стандарты моделирования.

Согласно стандарту имя объекта модели (сущности, таблицы, атрибута, колонки или домена) может состоять из четырех частей: корня, описателя класса и двух модификаторов. Для каждой части имени можно создавать (или импортировать) словарь. Допускается одновременная поддержка полного наименования части имени и аббревиатуры. Созданный в ERwin Naming Standards Editor файл стандартов может быть подключен к модели с возможностью дальнейшей проверки имен объектов модели на соответствие заданному стандарту.

Входящая в состав Erwin 4.0 утилита Datatype Standards Editor позволяет решить две задачи: во-первых, связать тип данных конкретной СУБД со встроенным логическим доменом ERwin и, во-вторых, переопределить таблицу конвертации типов данных при преобразовании физической модели с одной СУБД на другую.

В ERwin 4.0 существенно расширен язык макросов. В язык включены 8 новых макросов для работы с сущностями, 17 – для работы с атрибутами, 3 – для работы со связями и 6 – для работы с ограничениями.

Новые функциональные возможности ERwin 4.0 делают его незаменимым инструментом, особенно при создании крупных информационных систем.

3 Методические материалы к выполнению самостоятельной работы

Создать приложение для вывода данных из таблицы Заказов (Orders) с помощью статического запроса.

Последовательность действий

Поместить на форму компоненты Query1, DataSource1 и DBGrid1.

Задать значения свойств компонента Query1:

– в свойстве DatabaseName указать псевдоним (alias) базы данных (DBDemos). Вместо псевдонима можно задать путь к папке, в которой находятся таблицы БД;

– в свойстве SQL ввести запрос. Для этого в Инспекторе объектов для свойства SQL открыть StringListEditor и записать текст запроса, например: `Select *from orders`

– свойство Active установить в true.

Для компонента DataSource1 в свойстве DataSet указать Query1.

Для компонента DBGrid1 в свойстве DataSource задать DataSource1.

В результате все поля таблицы orders будут выведены в компонент DBGrid1. Если требуется вывести только часть полей, то надо изменить текст запроса:

`Select OrderNo, CastNo, SaleDate, ShipDate from Orders`

Так как любое изменение свойства SQL закрывает набор данных, то надо в свойстве Active задать true.

Программное использование Query

Доступ к свойству SQL возможен не только через Инспектор объектов во время создания проекта, но и программно, во время выполнения приложения

(run time). При программном использовании Query, рекомендуется приведённая ниже последовательность действий.

Закрывать текущий запрос.

Очистить список строк в свойстве SQL.

Добавить новые строки в запрос.

Инициировать выполнение запроса.

Так как свойство SQL имеет тип TStrings, то его можно формировать добавлением строк методом Add либо использованием метода LoadFromFile для загрузки текста из файла.

Сформировать программно запрос для вывода из таблицы Country всех сведений об Аргентине.

```
Query1.Close;
```

```
Query1.SQL.Clear;
```

```
Query1.SQL.Add('SELECT * FROM Country');
```

```
Query1.SQL.Add('WHERE Name=''Argentina''');
```

```
Query1.Open;
```

Если версия языка SQL позволяет использовать шаблоны поиска без учёта регистра (case insensitive), то, немного изменив запрос

```
Query1.SQL.Add('WHERE Name LIKE ''C%'');
```

удастся получить набор данных, содержащий все записи, в которых поле Name начинается с буквы 'C'.

Загрузить из файла запрос для вывода из таблицы biolife

сведений (Category,Length_in) о рыбах категорий, названия которых начинаются на S, упорядоченных по длине.

Для использования запроса, сохранённого в текстовом файле с расширением sql или txt, надо текст запроса занести в свойство SQL. Например, можно добавить в проект компоненты Button1, OpenFileDialog и записать обработчик щелчка по кнопке Button1 (события OnClick):

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
  if OpenFileDialog1.Execute then
```

```
    with Query1 do
```

```
      begin
```

```
        Close;
```

```
        SQL.LoadFromFile(OpenFileDialog1.FileName);
```

```
        Open;
```

```
      end;
```

```
end;
```

Предварительно в файле должен быть записан запрос:

```
SELECT category,Length_in FROM biolife
```

```
WHERE category LIKE 'S%' ORDER BY Length_in
```

Дополнительная литература:

1 Мельников В.П. Информационные технологии: учеб. пособие: рек. УМО/ В.П.Мельников. –М. Академия, 2008. -426 с.

2 Пшихопов В.Х. Оценивание и управление в сложных динамических системах: моно-го./ В.Х.Пшихопов, М.Ю.Медведев. –М.: Физматлит, 2009. -297 с.

3 Управляющие системы и автоматика: производственно-практическое издание/ Д.Шмид и др.: пер. с нем. Л.Н.Казанцева. –М.: Техносфера, 2007. -584 с.

4 Гагарина Л.Г. Разработка и эксплуатация автоматизированных информационных систем : учеб. пособие: рек. Мин. обр. РФ/ Л. Г. Гагарина, Д. В. Киселев, Е. Л. Федотова; под ред. Л. Г. Гагариной. -М.: ФОРУМ: ИНФРА-М, 2007.-384 с.

5 Соловьев И.В. Проектирование информационных систем. Фундаментальный курс: учеб. пособие: рек. УМО/ И. В. Соловьев, А. А. Майоров. -М.: Академический Проект, 2009. - 399 с.

6 Блюмин А.М. Проектирование систем информационного, консультационного и инновационного обслуживания: учеб. пособие/ А. М. Блюмин, Л. Т. Печеная, Н. А. Феоктистов. - М.: Дашков и К, 2009.-350 с.

7 И.Г.Безуглов. Основы научного исследования: учеб.пособие для аспирантов и студентов-дипломников/Ию.Г.Безуглов, В.В.Лебединский, А.И.Безуглов. –М.: Академический Проект, 2008. -195 с.

в) периодические издания:

1 Информатика и системы управления

2 Informations Processing & Menagement

г) программное обеспечение и Интернет-ресурсы:

Свободно распространяемое программное обеспечение

№ п/п	Наименование ресурса	Характеристика
1	see http://www.iqlib.ru	Интернет библиотека образовательных изданий, в которой собраны электронные учебники, справочные и учебные пособия. Удобный поиск по ключевым словам, отдельным темам и отраслям знаний.
2	http://www.intuit.ru	Интернет-университет информационных технологий, в котором вобраны электронные и видео-курсы по отраслям знаний
3	http://amursu.ru	Сайт АмГУ, Библиотека – электронная библиотека АмГУ
4	http://www.biblioclub.ru	Электронная библиотечная система «Университетская библиотека – online»: специализируется на учебных материалах для ВУЗов по научно-гуманитарной тематике, а так же содержит материалы по точным и естественным наукам

3. Перечень используемых программных продуктов

Windows XP; MS Office XP; BPWin; ERWin; Delphi.

4.Фонд тестовых и контрольных заданий

Оценочные средства для текущего контроля успеваемости

Вопросы к зачету:

В каких базах данных используется язык SQL?

Назовите типы команд языка SQL.

Для чего предназначен язык определения данных ЯОД?

Для чего предназначен язык манипулирования данными ЯМД?

Какая из имеющихся в Delphi утилит позволяет выполнять команды SQL в интерактивном режиме?

Как называется операция, позволяющая получить подмножество столбцов таблицы?

В чём суть операции выборки?

Каков будет результат применения декартова произведения к двум отношениям?

Чем представление принципиально отличается от таблицы?

Перечислите служебные слова, используемые в команде SELECT.

Где указывается перечень выводимых полей?

Что записывается после слова FROM в команде SELECT?

В каких случаях при задании поля приходится использовать составные имена?

Какие операции можно использовать при формировании условия в предложении WHERE?

Поясните использование операций сравнения, BETWEEN, IN, LIKE.

Как изменить порядок выводимых строк в результирующем наборе?

Чем определяется порядок действий при вычислении значения условия?

Можно ли упорядочить выводимые записи по нескольким полям?

Какая логическая операция – or или and имеет более высокий приоритет?

Почему при работе с полями даты могут возникнуть проблемы?

Как устранить дублирование в результирующем наборе?

Какие операции можно использовать при выполнении вычислений в запросах?

Что в запросах вычисляют функции AVG и COUNT?

Для чего используется группировка в запросах?

В каких случаях для задания условия отбора записей необходимо использовать предложение HAVING?

Сформировать запросы к таблице employee из DBDemos:

- подсчитать количество сотрудников, у которых заработная плата больше 35 000;
- найти сотрудников, фамилия которых начинается на букву 'B';
- найти сотрудников, поступивших на работу в 1993 г.;
- вывести сведения о сотрудниках, у которых телефон начинается цифрами 22;
- определить среднюю зарплату сотрудников, работающих с 1989 г.

Сформировать запросы к таблице custody из DBDemos:

- определить, кто не имеет e-mail;
- найти людей, фамилия которых начинается на 'Sm';
- найти людей, фамилия которых начинается на 'S', а телефон – на 8;
- подсчитать количество покупателей из каждой страны;
- отсортировать записи по фамилиям.

Какая операция называется соединением?

Приведите правила соединения таблиц.

В чём суть внутреннего (левого, правого, полного) соединения?

Для чего в запросах используются псевдонимы?

В каких случаях используется операция объединения?

Для чего используются подзапросы?

Приведите правила формирования подзапросов.

Сформировать запрос к таблице order из DBDemos, позволяющий получить сведения о заказе, у которого самая большая разница между суммой заказа и оплаченной суммой.

Сформировать запрос для получения сведений о покупателе, сделавшем наибольшее количество заказов (таблицы orders и customer).

Как расшифровывается аббревиатура API?

Чем по сути является API?

Назовите недостатки универсальных механизмов доступа к данным.

При использовании ODBC для каждой СУБД нужен собственный ODBC-драйвер или есть универсальный?

Как организовать доступ к данным по технологии OLE DB, если для нужной СУБД нет OLE DB-драйвера, но есть ODBC-драйвер?

Приведите схему доступа к данным с применением ADO.

Почему активно используется доступ к данным с использованием ADO?

Какие объекты используются в технологии ADO?

Какие компоненты Delphi используются для организации доступа к данным по технологии ADO?

Как задаются параметры соединения при разработке в Delphi приложения, использующего технологию ADO?