

Учебная дисциплина
Управление жизненным циклом ИС

Лекция 2

**АВТОМАТИЗИРОВАННОЕ
ПРОЕКТИРОВАНИЕ
ИНФОРМАЦИОННЫХ СИСТЕМ
С ИСПОЛЬЗОВАНИЕМ
CASE-ТЕХНОЛОГИЙ**

Лектор:

Шлаев Дмитрий Валерьевич

кандидат технических наук,

доцент

Учебные вопросы:

- 1. Основные понятия и классификация CASE-технологий**
- 2. Функционально-ориентированное проектирование ИС**

Литература:

1. Вендров А.М.

Проектирование программного обеспечения экономических информационных систем: Учебник. – М. Финансы и статистика, 2006. – 352 с.

1. Основные понятия и классификация CASE-технологий

Появлению CASE-технологии способствовало:

- подготовка аналитиков и программистов, восприимчивых к концепциям модульного и структурного программирования;
- широкое внедрение и постоянный рост производительности персональных ЭВМ, позволяющих 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.

Преимущества CASE-технологии:

- улучшение качества разрабатываемого программного приложения за счет средств автоматического контроля и генерации;
- возможность повторного использования компонентов разработки;
- поддержание адаптивности и сопровождения ЭИС;
- снижение времени создания системы, что позволяет на ранних стадиях проектирования получить прототип будущей системы и оценить его;
- освобождение разработчиков от рутинной работы по документированию проекта, так как при этом используется встроенный документатор;
- возможность коллективной разработки ЭИС в режиме реального времени.

CASE-технология в рамках методологии включает в себя методы, с помощью которых на основе графической нотации строятся диаграммы, поддерживаемые инструментальной средой.

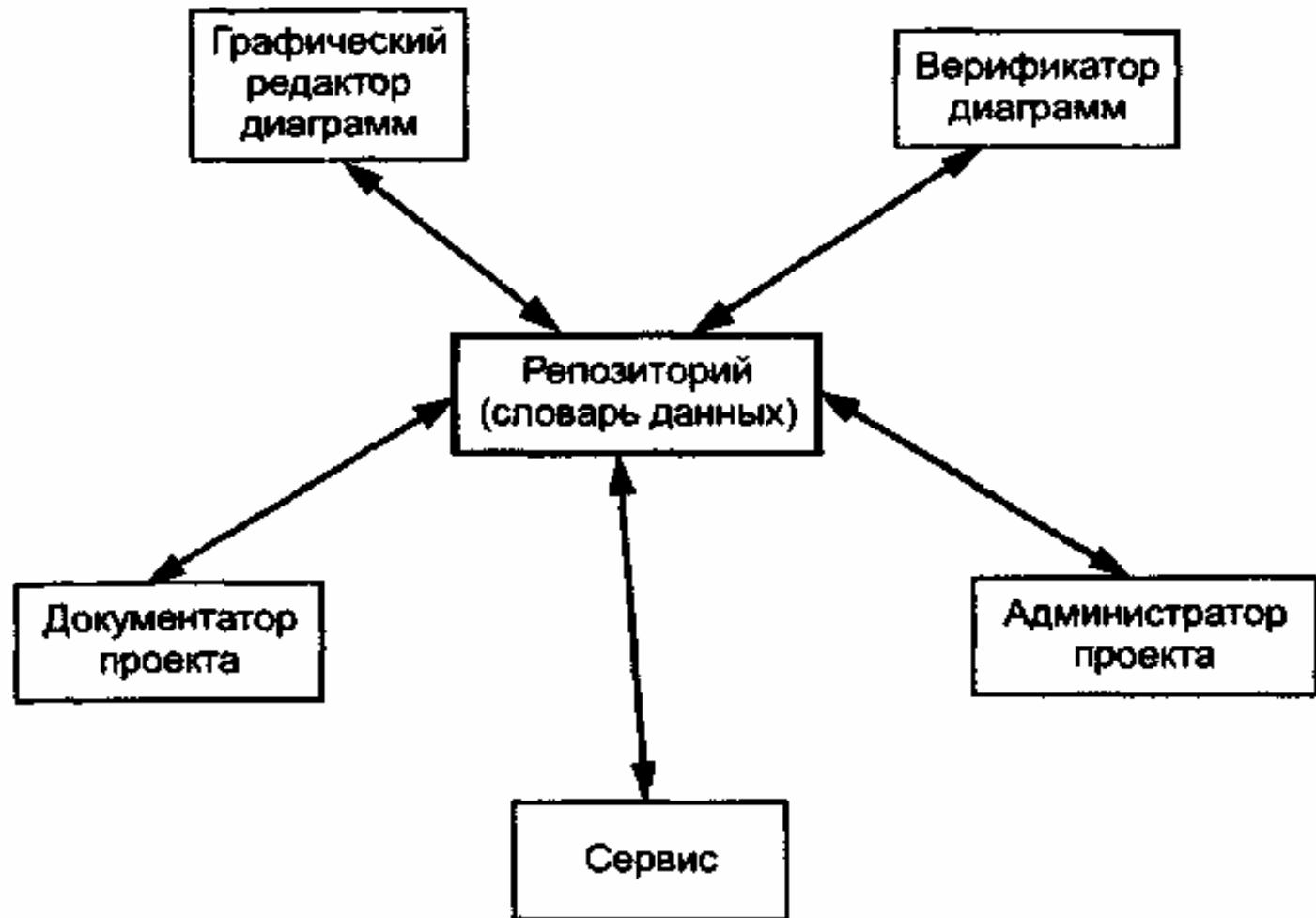
Методология определяет шаги и этапность реализации проекта, а также правила использования методов, с помощью которых разрабатывается проект.

Метод - это процедура или техника генерации описаний компонентов ЭИС (например, проектирование потоков и структур данных).

Нотация - отображение структуры системы, элементов данных, этапов обработки с помощью специальных графических символов диаграмм, а также описание проекта системы на формальных и естественных языках.

Инструментальные средства CASE - специальные программы, которые поддерживают одну или несколько методологий анализа и проектирования ИС.

Архитектура CASE-средства



Репозиторий (словарь данных) представляет собой специализированную базу данных, предназначенную для отображения состояния проектируемой ЭИС в каждый момент времени.

В репозитории хранятся описания следующих объектов:

- проектировщиков и их прав доступа к различным компонентам системы;
- организационных структур;
- диаграмм;
- компонентов диаграмм;
- связей между диаграммами;
- структур данных;
- программных модулей;
- процедур;
- библиотеки модулей и т.д.

Графический редактор диаграмм предназначен для отображения в графическом виде в заданной нотации проектируемой ЭИС.

Он позволяет выполнять следующие операции:

- создавать элементы диаграмм и взаимосвязи между ними;
- задавать описания элементов диаграмм;
- задавать описания связей между элементами диаграмм;
- редактировать элементы диаграмм, их взаимосвязи и описания.

Верификатор диаграмм служит для контроля правильности построения диаграмм в заданной методологии проектирования ЭИС.

Он выполняет следующие функции:

- мониторинг правильности построения диаграмм;
- диагностику и выдачу сообщений об ошибках;
- выделение на диаграмме ошибочных элементов.

Документатор проекта позволяет получать информацию о состоянии проекта в виде различных отчетов. Отчеты могут строиться по нескольким признакам, например по времени, автору, элементам диаграмм, диаграмме или проекту в целом.

Администратор проекта представляет собой инструменты, необходимые для выполнения следующих административных функций:

- инициализации проекта;
- задания начальных параметров проекта;
- назначения и изменения прав доступа к элементам проекта;
- мониторинга выполнения проекта.

Сервис представляет собой набор системных утилит по обслуживанию репозитория. Данные утилиты выполняют функции архивации данных, восстановления данных и создания нового репозитория.

Современные CASE-системы классифицируются по следующим признакам:

по поддерживаемым методологиям проектирования:

функционально (структурно) - ориентированные,
объектно-ориентированные и комплексно-
ориентированные (набор методологий проектирования);

по поддерживаемым графическим нотациям

построения диаграмм: с фиксированной нотацией, с
отдельными нотациями и наиболее распространенными
нотациями;

по степени интегрированности: tools (отдельные локальные средства), toolkit (набор неинтегрированных средств, охватывающих большинство этапов разработки ЭИС) и workbench (полностью интегрированные средства, связанные общей базой проектных данных - репозиторием);

по типу и архитектуре вычислительной техники: ориентированные на ПЭВМ, ориентированные на локальную вычислительную сеть (ЛВС), ориентированные на глобальную вычислительную сеть (ГВС) и смешанного типа;

по режиму коллективной разработки проекта: не поддерживающие коллективную разработку, ориентированные на режим реального времени разработки проекта, ориентированные на режим объединения подпроектов;

по типу операционной системы (ОС): работающие под управлением WINDOWS 3.11 и выше; работающие под управлением UNIX и работающие под управлением различных ОС (WINDOWS, UNIX, OS/2 и др.)

2. Функционально-ориентированное проектирование ЭИС

Основными идеями функционально-ориентированной CASE-технологии являются идеи структурного анализа и проектирования информационных систем.

Они заключаются в следующем:

- декомпозиция всей системы на некоторое множество иерархически подчиненных функций;
- представление всей информации в виде графической нотации. Систему всегда легче понять, если она изображена графически.

В качестве инструментальных средств структурного анализа и проектирования выступают следующие диаграммы:

- **BFD (Business Function Diagram)** - диаграмма бизнес-функций (функциональные спецификации);
- **DFD (Data Flow Diagram)** - диаграмма потоков данных;
- **STD (State Transition Diagram)** - диаграмма переходов состояний (матрицы перекрестных ссылок);
- **ERD (Entity Relationship Diagram)** - ER-модель данных предметной области (информационно-логические модели «сущность-связь»);
- **SSD (System Structure Diagram)** - диаграмма структуры программного приложения.

Диаграммы функциональных спецификаций

позволяют представить общую структуру ИС, отражающую взаимосвязь различных задач (процедур) в процессе получения требуемых результатов.

Диаграммы потоков данных (ДПД), как правило, жестко ориентированы на какую-либо технологию обработки данных и отражают передачу информации от одной функции к другой в рамках заданной технологии обработки.

Диаграммы переходов состояний (ДПС)

моделируют поведение системы во времени в зависимости от происшедших событий (нажатая клавиша, дата отчетного периода и т.д.).

Диаграммы инфологических моделей «сущность-связь» (ER-диаграммы) ориентированы на разработку базы данных, структура которой не зависит от конкретных информационных потребностей и позволяет выполнять любые запросы пользователей.

Диаграмма структуры программного приложения (SSD) задает взаимосвязь функций и программных модулей, которые их реализуют (меню, формы, отчеты и т.д.).