Лекция 3

Проектирование

На первичном этапе разработки ПО, необходимо определить спецификацию программ и выполнить проектирование. При выполнении данных действий используются 2 подхода:

1. Структурный. Предполагает выделение основных функций системы, разработку алгоритмов их реализующих, установку взаимосвязей между потоками данных, циркулирующих в системе.
2. ООП. Предполагает выделение предметной области подлежащей автоматизации, набора действующих объектов и объектов принимающих действия, определение их функционала, с последующей разработкой требуемых классов.

Структурный подход

Определение спецификации при структурном подхода, предполагает точное описание функций и ограничений разрабатываемого ПО, при этом выделяют функциональную спецификацию и эксплуатационную.

Точность спецификации определяет посредством формальных моделей. Формальные модели делят на 2 группы:

1. Зависящие от подхода.
2. Не зависящие от подхода.

Каждый тип модели целесообразно использовать для своего специфического класса ПО. Методология структурного проектирования и анализа основана на моделировании потока данных и использует представление о проектировании системы в виде комплекта моделей.

Модели:

* Диаграмма потока данных.
* Диаграмма спецификации процессов.
* Словари терминов
* Диаграмма «сущность-связь»
* Диаграммы состояния
* Функциональные диаграммы

При построении моделей структурного подхода изначально определяется концептуальная диаграмма (контекстная) наиболее общим образом описывающая проектирование процессов. На концептуальной диаграмме определяется набор внешних данных, как входных таки выходных и набор воздействий. Выделяется внешние и внутренние сущности системы с минимальным описание передаваемых функций. Затем выполняется постепенная детализация автоматизируемого процесса. Для каждого процесса составляется спецификация процесса. Если процесс использует хранилище данных, то используется «сущность-связь», если управляющим, то диаграмма переходов состояний, для регламентирования используемых в системных терминов используется словарь терминов. Детализация выполняется до того уровня, пока алгоритм процесса не приобретает вид известного и элементарного.

Для каждого детального процесса и концепции может быть составлена диаграмма потока данных. Она описывает взаимодействие источников и приемников информации через процессы системы. Позволяет специфицировать функции. Существует 2 основных нотации для отображения диаграмм:

* Йордана
* Гейна-сарсона

Основными элементами является:

* Внешняя сущность. Материальный объект выступающий в качестве источника или приемника информации.
* Функция или процесс. Выполняет преобразование входных потоков данных с выходными, в соответствии с алгоритмом. Каждый процесс имеет свое имя и номер. Имя для однозначной идентификации, а номера для иерархии
* Поток данных. Определяет дуги, соединяющие между собой процессы и хранилище. Каждый поток имеет имя, соответствующий передаваемой информации.
* Хранилище данных. Определяется как абстрактное устройство для хранения информации. Тип устройства, способ занесения и извлечения информации не регламентируется. Физически в качестве устройства используется БД, файл, картотека и тд.

Контекстная диаграмма состоит из одного элемента процесса и набора элементов внешних сущностей.

Детализация. Правила выполнения детализации. Все входные и выходные воздействия должны сохранятся. Не допускается появления новых воздействий. Необходимо выполнять нумерацию детализированных процессов. Детализация выполняется до тех пор, пока:

1. Процесс взаимодействует не более чем с двумя входными и выходными потоками данных
2. Описание процесса – последовательный алгоритма
3. Процесс выполняет единственную логическую функцию преобразования входной информации в выходную

­Для не детализированных процессов составляется спецификация, которая содержит описания логики поведения данного процесса. Для определения спецификации используются следующие методы:

* Псевдокод. Представляет собой формализованное текстовое описание алгоритма, приближенное к языку программирования высокого уровня. Согласно алгоритмическим структурам в псевдокоде определяют конструкции:
  + Следование
  + Ветвление
  + Цикл с пред условием
  + Пост условием
  + С заданным количеством повторений
* Flow-формы. Графическая нотация для описания структурных алгоритмов. Каждый символ Flow-формы определяет одну из возможных вершин, символы могут быть вложены друг в друга. Вершин может быть 6 видов.
* Диаграммы Насси-Шнейдермана. Графическое представление с другими условными обозначениями.

Для каждого процесса составляется словарь терминов – краткое описание понятий для составления спецификации. Словарь нужен для того, чтобы заказчик понимал разработчика.

*Термин*

*Категория*

*<Краткая категория>*

Для описания структуры ПО используют структурные карты. Структурные карты используются на этапе проектирования для демонстрации того как программный продукт будет выполнять системные требования.

Типы карт:

* Структурные карты Константийна. Представляет собой отношение между модулями программы. Узлы соответствуют модулям и областям данных, потоки отображают межмодульные связи.
  + Модуль. Реализация\подсистема\библиотека
  + Поток определяющий вызов модуля.
  + Связь между модулями по данным.
  + Связь между модулями по управлению
  + Область данных. Содержит только глобальные переменные
* Схема Джексона. Выявляет соответствие между потока данных и выходных данных.
  + Операция
  + Следование
  + Выбор
  + Итерация
* Диаграмма Варение-Орра. Вывод структуры программы и структуры входных потоков исходя из структур выходных данных. Включает в себя физические хранилища информации. Магнитные носители, печатные копии и потоки ввода-вывода
* Hipo. Иерархическая схема описывающая входные данные, процедуры обработки и набор выходных параметров

Диаграмма «сущность-связь». Используется нотация IDEF1. Согласно ей, вводятся понятия:

* Сущность. Представляет собой класс однотипных объектов информация о которых должна учтена в модели. Сущность имеет наименование. Именуются существительным в ед. числе Им. Падеже. Сущность определяет параметры каждого элемента. Параметры определяются набором свойств (атрибуты сущности). Наименование атрибута – существительное. Каждая сущность должна иметь ключ или первичный индекс. Ключ сущности – не избыточный набор атрибутов, значение которых обеспечивает уникальность каждой сущности. Ключевой атрибут пишется первым и подчеркивается. Сущности связаны между собой. Связь – отношение одной сущности к другой или к самой себе. Виды связи:
  + Один-к-одному
  + Каждый элемент связан с несколькими. Один ко многим
  + Много-ко-многим

Связи могут как быть обязательными, так и не обязательными.

Инфологическая модель данных – средство описания области. Определяет набор хранимых и обрабатываемых объектов. Из предположений инфологической модели определяют сущности на основе которых формируют логическую модель.

Физическая модель определяет правила хранения и правила определения ряда атрибутов. Физическая модель привязывается к будущей и связывается с выбранной СУБД. Необходимо выполнить соответствие имеющих типов. При преобразовании логической модели к физической. Устраняются связи многие ко многим. За счет ведения дополнительных таблиц. В случае использования обязательной связи выполняется транспорт ключевого поля родительской таблицы к дочерней. Такой ключ называют внешним.

Диаграммы состояний определяют набор состояний системы и правило перехода между этими состояниями. Диаграмма состояний реализуется в виде конечного автомата. На этапе анализа требований диграмма состояний демонстрирует поведение системы. Под управляющей информации понимается информация передаваемая системе из вне.