

Методические основы анализа и проектирования ПО

Тема 4

Методы структурного анализа и проектирования

Структурный анализ — один из формализованных методов анализа требований и проектирования ПО (автор Том Де Марко).

Основная задача – описание:

- а) функциональной структуры системы;
- б) последовательности выполняемых действий;
- в) передачи информации между функциональными процессами;
- г) отношений между данными.

Модели структурного анализа и проектирования:

- Функциональная модель SADT (Structured Analysis and Design Technique);
- Модель IDEF3;
- DFD (Data Flow Diagrams) – диаграммы потоков данных

-
- **Метод SADT (IDEF0)** – совокупность правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области (производимые действия и связи между ними).
 - **Модель IDEF3** – предназначена для моделирования последовательности выполняемых действий и взаимозависимости между ними, основа модели – сценарий процесса

DFD (Data Flow Diagrams) – иерархия функциональных процессов, связанных потоками данных.

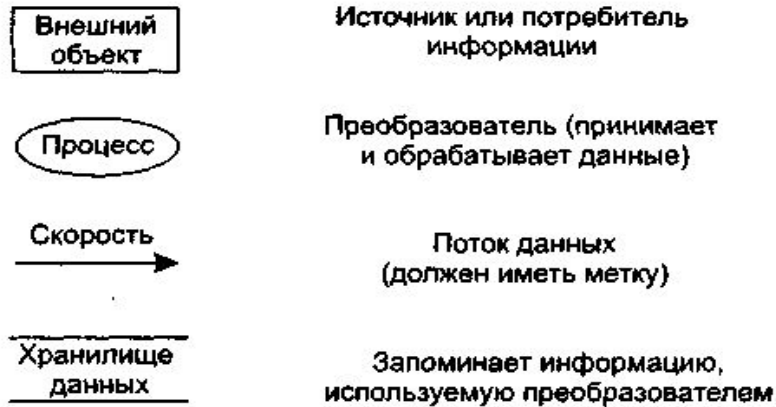
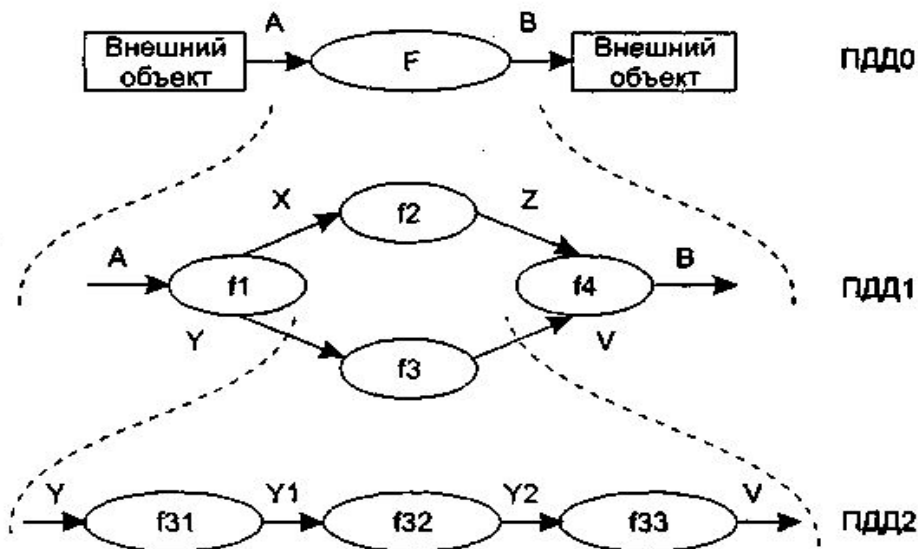


Рис. 3.1. Элементы диаграммы потоков данных



Методы объектно-ориентированного анализа и проектирования

Концептуальная основа ООАП – объектная модель, ее основные принципы (абстрагирование, инкапсуляция, модульность, иерархия) и понятия (объект, класс, атрибут, операция, интерфейс и др.).

UML (Unified Modeling Language) – язык для определения, представления, проектирования и документирования систем различной природы.

Основные задачи UML:

- предоставить пользователям готовый к использованию выразительный язык визуального моделирования, позволяющий им разрабатывать осмысленные модели и обмениваться ими;
- предусмотреть механизмы расширяемости и специализации для расширения базовых концепций;
- обеспечить независимость от конкретных языков программирования и процессов разработки;
- обеспечить формальную основу для понимания этого языка моделирования (язык должен быть одновременно точным и доступным для понимания, без лишнего формализма);
- стимулировать рост рынка объектно-ориентированных инструментальных средств;
- интегрировать лучший практический опыт.

Структурные (structural) модели:

- •диаграммы классов (class diagrams)– для моделирования статической структуры классов системы и связей между ними;
- •диаграммы компонентов (component diagrams) – для моделирования иерархии компонентов (подсистем) системы;
- •диаграммы размещения (deployment diagrams) – для моделирования физической архитектуры системы.

Модели поведения (behavioral):

- •диаграммы вариантов использования (use case diagrams) – для моделирования функциональных требований к системе (в виде сценариев взаимодействия пользователей с системой);
- •диаграммы взаимодействия (interaction diagrams):
- •диаграммы последовательности (sequence diagrams) и кооперативные диаграммы (collaboration diagrams) – для моделирования процесса обмена сообщениями между объектами;
- •диаграммы состояний (statechart diagrams) – для моделирования поведения объектов системы при переходе из одного состояния в другое;
- •диаграммы деятельности (activity diagrams) – для моделирования поведения системы в рамках различных вариантов использования, или потоков управления.

Анализ требований

Что должно делать будущее ПО?

1. Система должна предоставлять пользователю доступ к балансу его банковского счета.
2. Балансы счетов клиентов будут храниться в таблице под названием «балансы» в базе данных Access.

Результат анализа требований – спецификация требований к ПО (SRS – Software Requirements Specification)

Требования заказчика и разработчика

- С-требования – требования заказчика
- D-требования – требования разработчика

Каждое требование должно быть:

четко выражено;

легко доступно;

пронумеровано;

сопровождаться подтверждающими требованиями;

предусматриваться проектом;

учтено кодом;

протестировано отдельно;

протестировано совместно с остальными требованиями;

подтверждено тестированием после сборки приложения.

Типичная схема процесса анализа С-требований

1. Идентифицировать «заказчика»

2. Провести интервью с представителями заказчика

- Определить желания и потребности
- Использовать инструменты поддержки
- Набросать графический интерфейс пользователя
- Определить конфигурацию оборудования

Согласовать с заказчиком

3. Написать С-требования в форме стандартного документа

4. Проверить С-требования

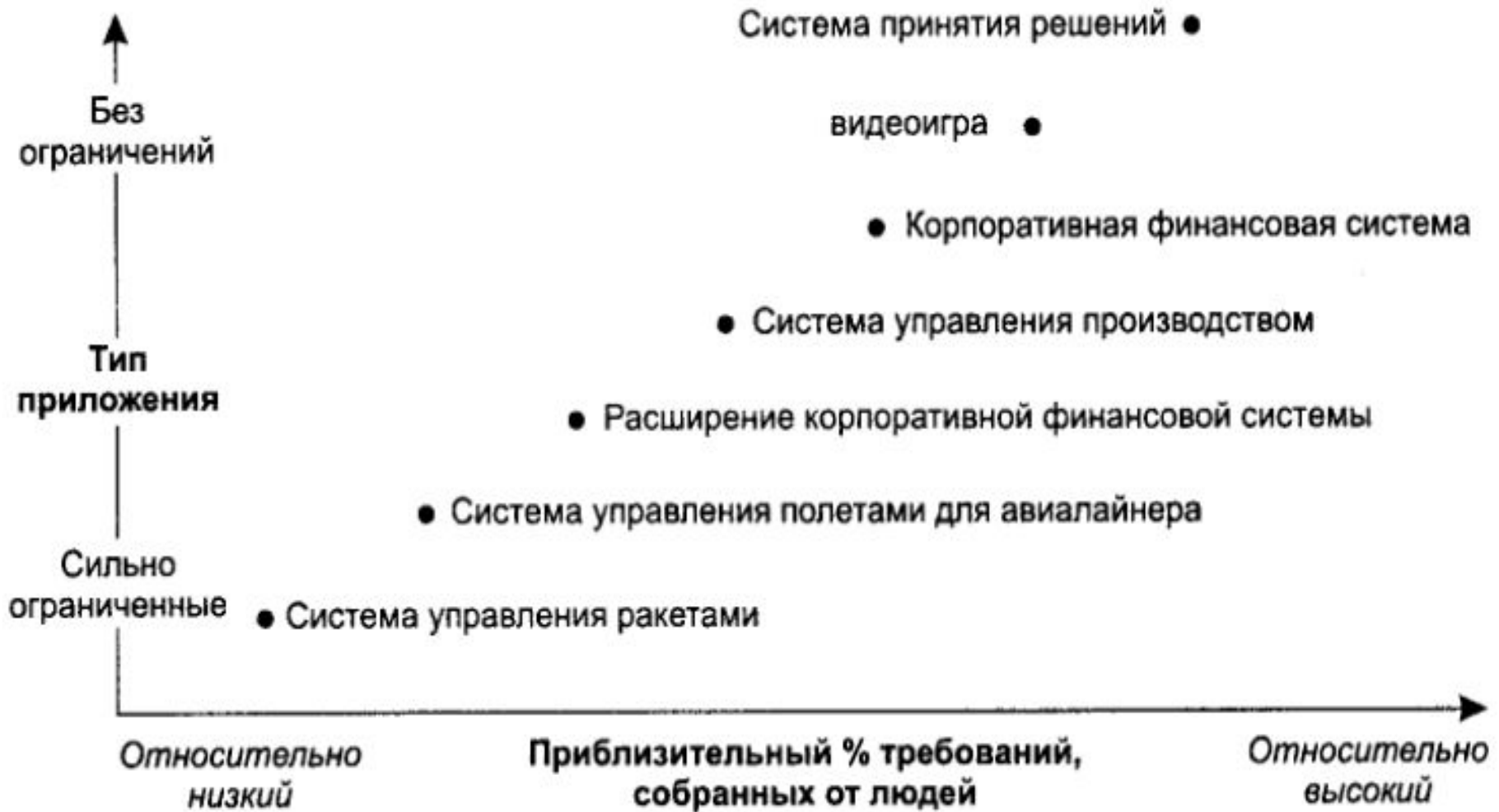
По согласованию с заказчиком...

5. Построить D-требования

Для всех этапов отследить метрики, например:

- Затраченное время
- Полученные величины:
 - Количество страниц С-требований
 - Количество минут общения с заказчиком на страницу
- Самооценка качества (шкала 1–10)
- Оценка дефектов по проверкам

Источники возникновения требований



Заинтересованные лица




Сайт электронной коммерции

Финансово заинтересованные стороны (доля в результирующем продукте):

Описание С-требований

1. Концепция работы.

Приложение «Бюро погоды»

-  Приложение для преобразования необработанных данных метеоцентра в графическое представление
-  Система реального времени для предсказания погоды
-  Приложение, прогнозирующее погодные аномалии

Варианты использования (Якобсон)

Требование выражается через взаимодействие приложения с внешним пользователем.

Например: команда *открыть файл*

1. *Пользователь* открывает меню *Файл*.
2. *Система* показывает команды *Новый* и *Открыть*.
3. *Пользователь* выбирает *Открыть*.
4. *Система* показывает окно *файлов*.
5. *Пользователь* вводит *каталог* и *имя файла*.
6. *Пользователь* щелкает на кнопке *Открыть*.
7. *Система* находит упомянутый *файл* и открывает его в окне *текстового редактора*.

Варианты использования

Действующие лица



Игрок



Разработчик

Инициализировать

Перейти в соседнюю зону

Вступить в контакт с внешним персонажем

Установить правила

1 подробности вариантов использования

Инициализировать

1. Система показывает главный персонаж игрока
2. Система показывает окно для установки характеристик персонажа
3. Игрок назначает характеристики своему главному персонажу
4. Игрок направляет персонаж
5. Система перемещает главный персонаж игрока

2 подробности вариантов использования

Вступить в контакт с внешним персонажем

1. Система показывает внешний персонаж в той же зоне, что и персонаж игрока
2. Система сравнивает значения характеристик двух персонажей
3. Система показывает результаты контакта
4. Система показывает персонаж игрока в случайно выбранной зоне

Диаграммы потоков данных

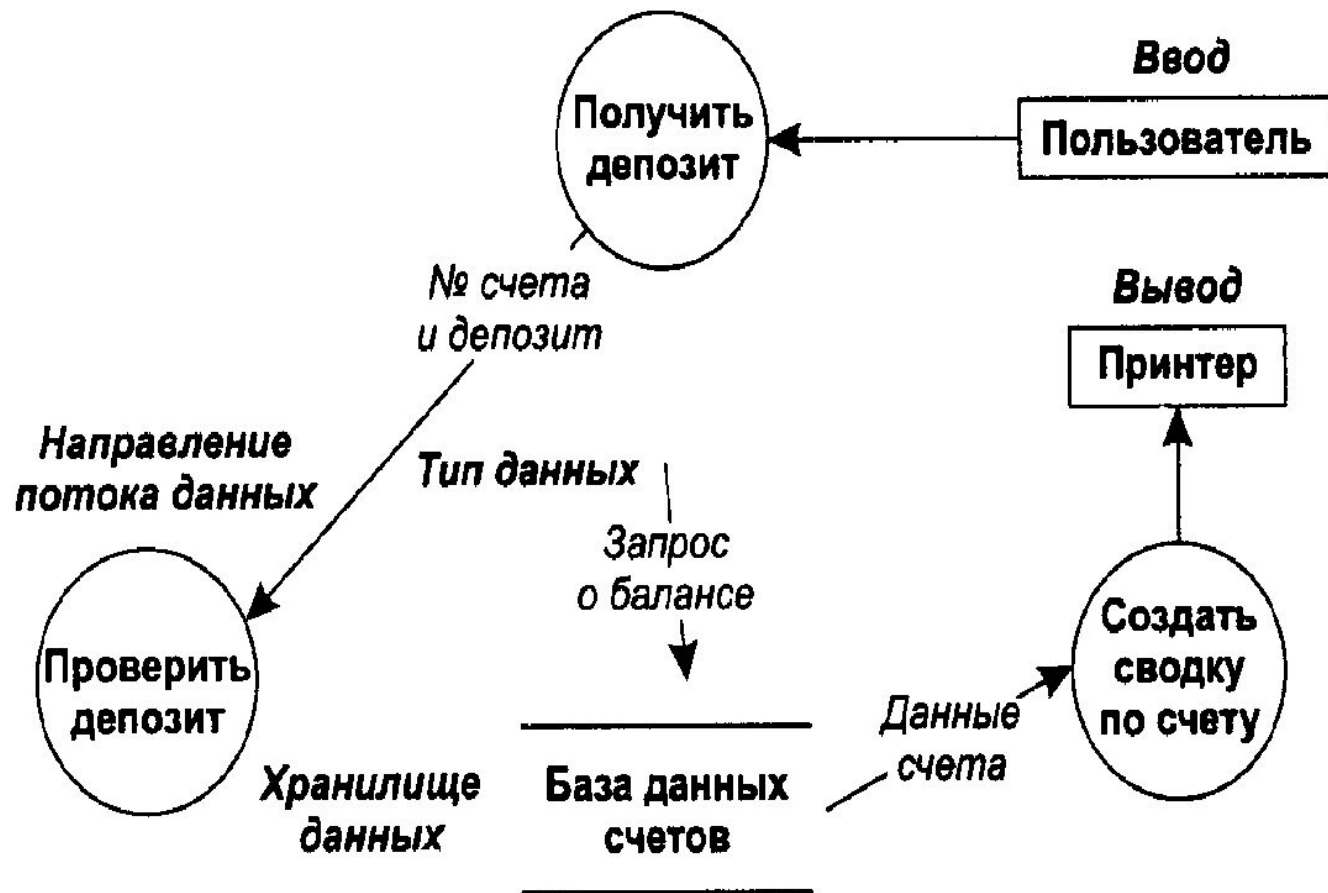
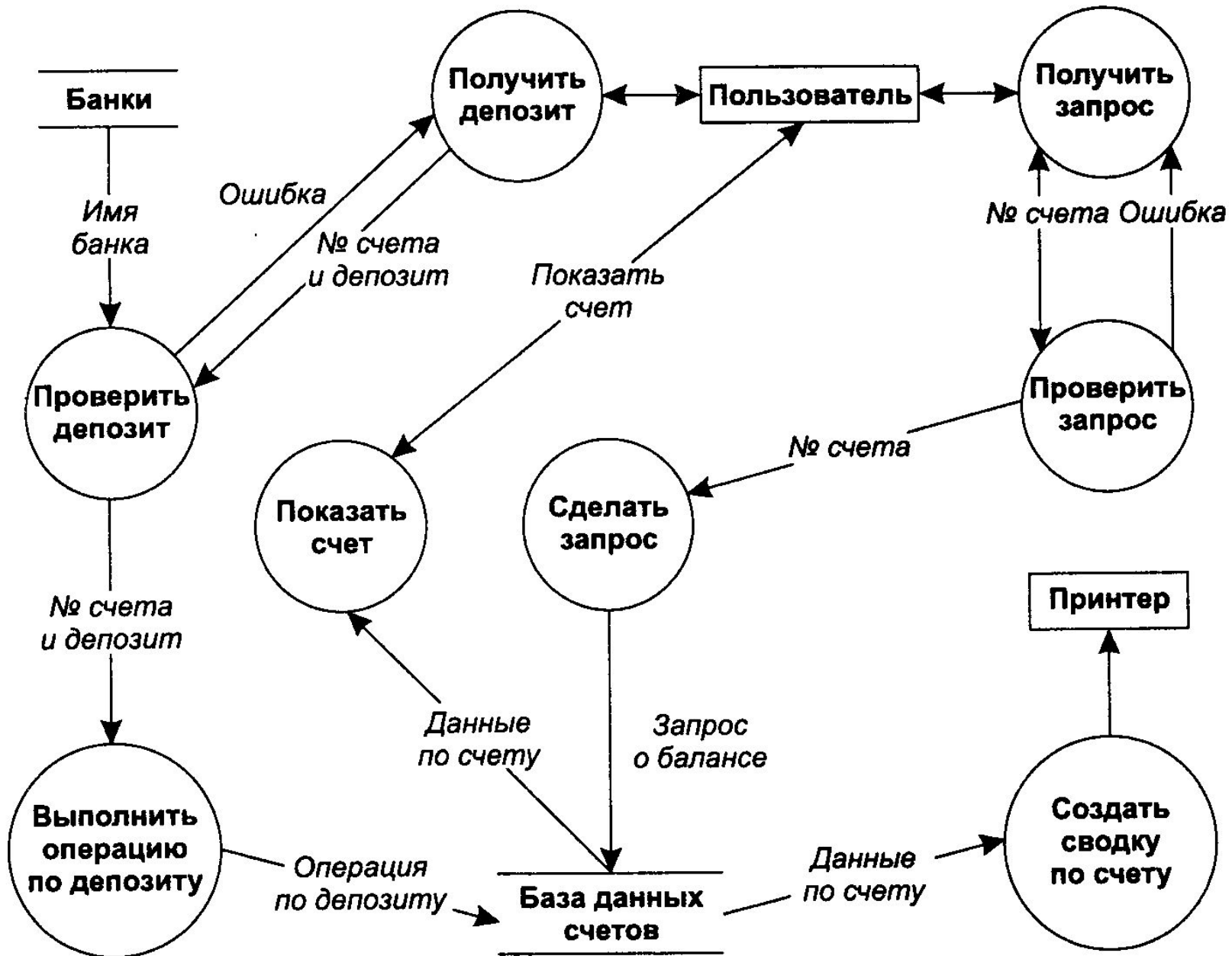
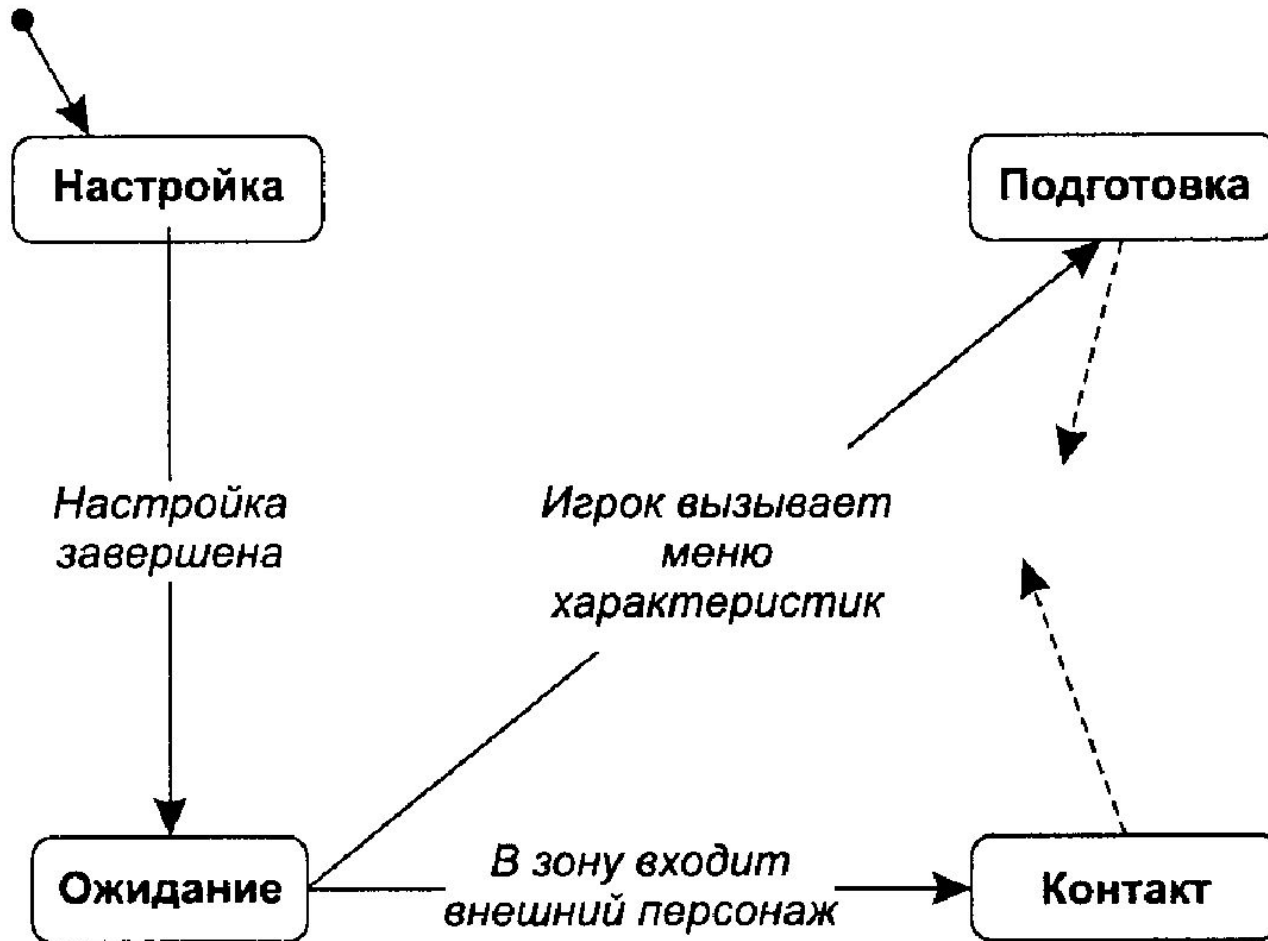


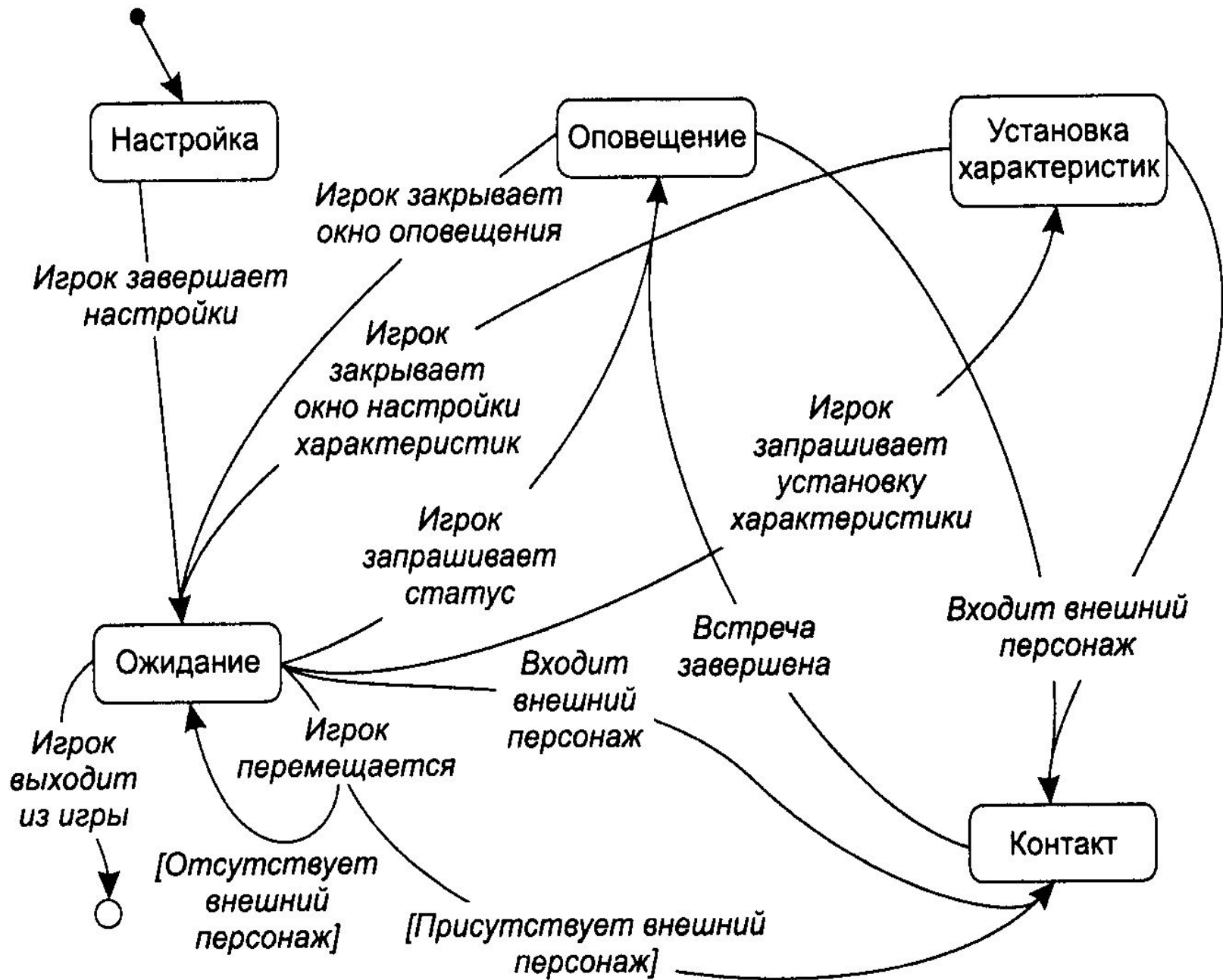
Рис. 3.7. Диаграмма потоков данных: объяснение символов



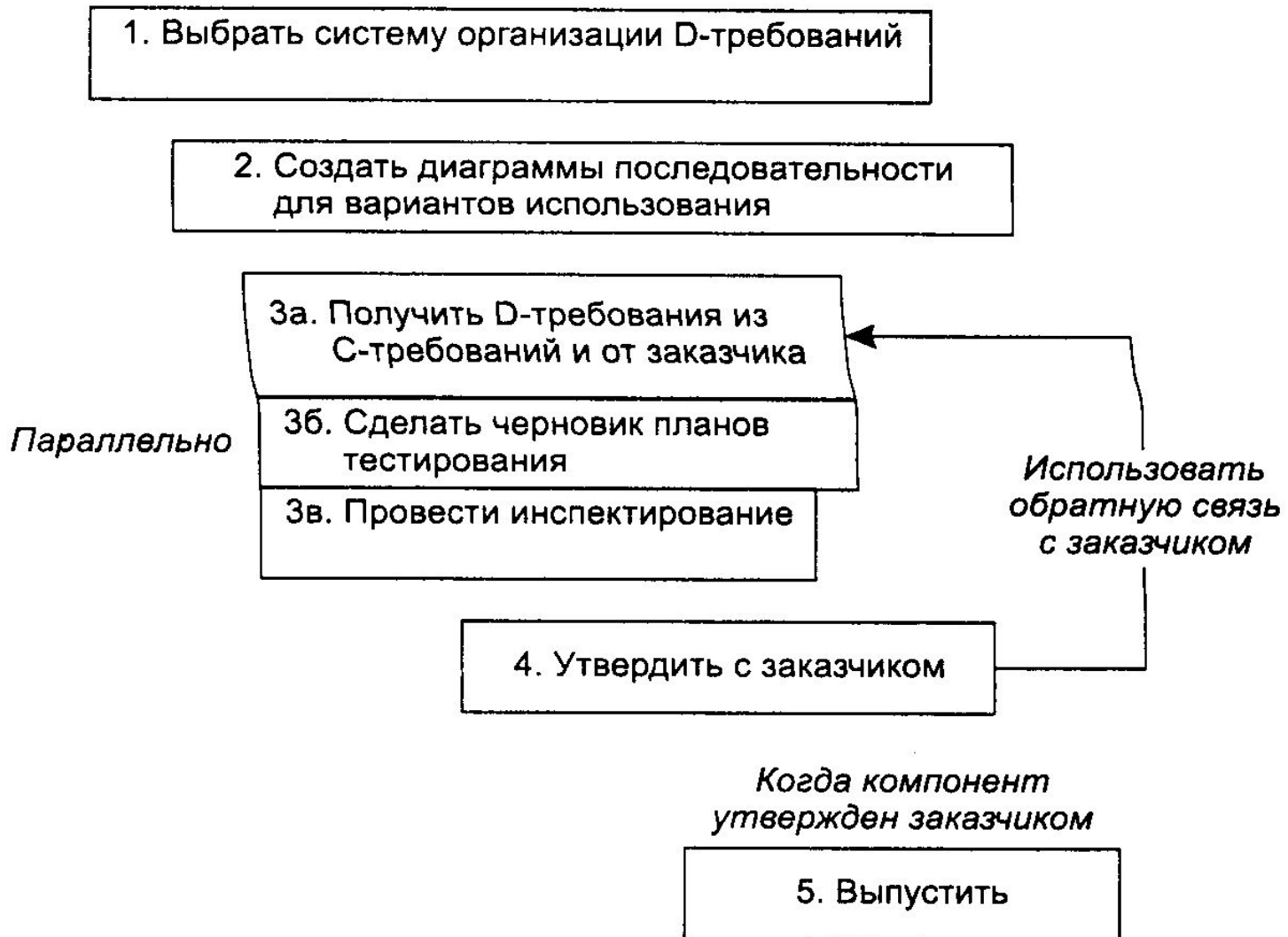
Неполная диаграмма потоков данных для программы управления банкоматом

Диаграмма переходов состояний





D-требования - конкретные требования, функциональные спецификации, требования разработчика



Типы D-требований

1. Функциональные требования:
 - функциональность приложения.
2. Нефункциональные требования:
 - Производительность (скорость, пропускная способность, использование памяти и т.д.);
 - Надежность и доступность;
 - Обработка ошибок;
 - Интерфейсные требования;
 - Ограничения (точность, ограничения на инструменты и язык, ограничения проектирования, использование стандартов, использования платформ и т.д.).
3. Обратные требования.

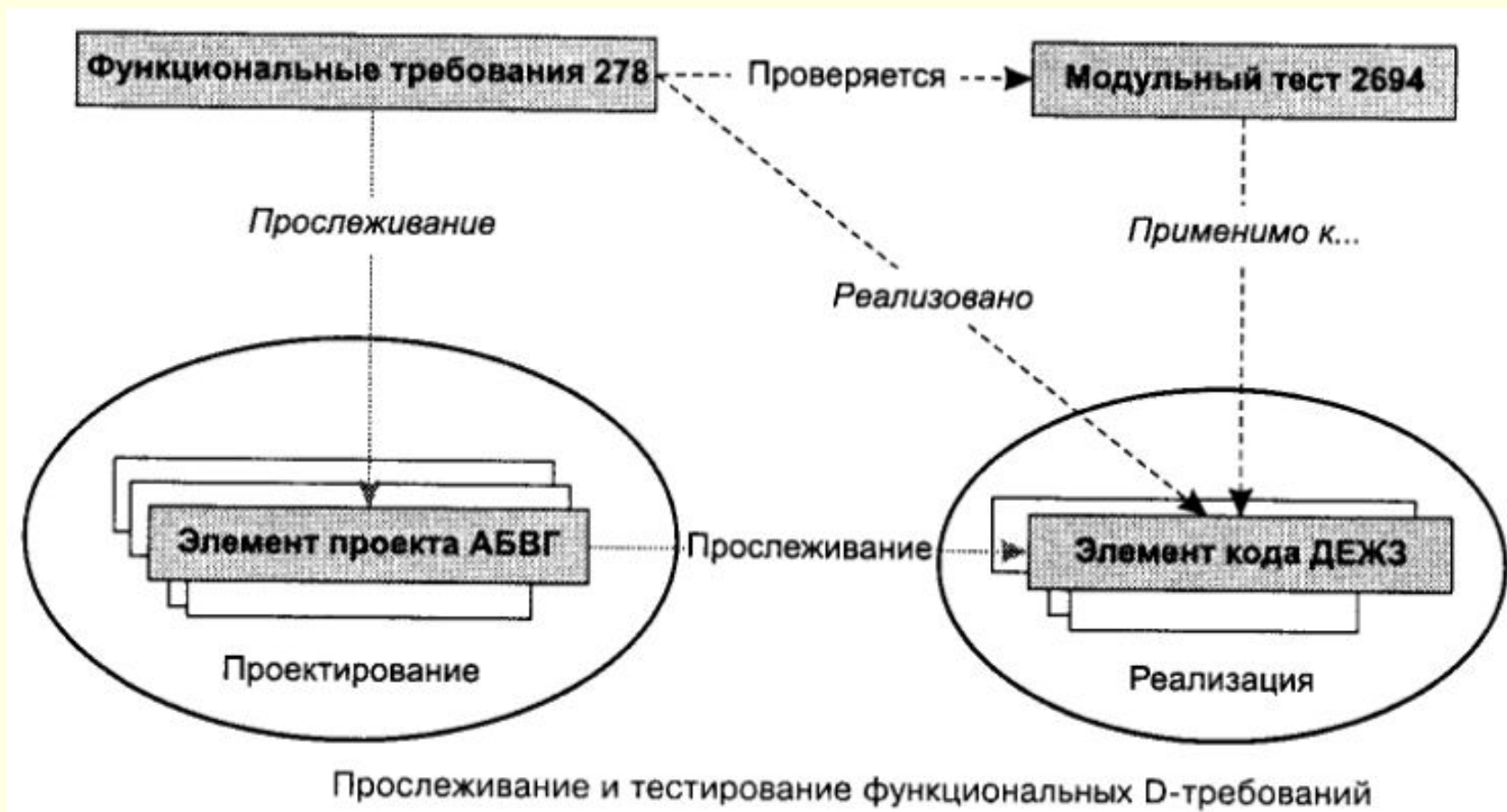
Примеры

- Приложение будет вычислять стоимость портфеля акций пользователя.
- Для любой балки анализатор давления должен создавать отчет типа 5 о давлении менее чем за минуту.
- Приложение, управляющее радаром аэропорта, должно давать не более двух ошибок в месяц.
- Приложение, управляющее радаром аэропорта, должно быть доступно как на основном, так и на запасном компьютере не более 2% времени в любой 30-дневный период.
- Стоимость посылки статьи от адресата получателю должна постоянно показываться в текстовом окне «Цена».
- Формат, используемый для передачи сообщений для взаимодействия с почтовыми компаниями, будет представлять собой строку вида *exp<source>*, где *<source>* - это строка из таблицы стандартов городов.

-
- Вычисления оценки ДТП системой должны быть выполнены с точностью до одного сантиметра.
 - Система АЕФ должна использовать систему UCSF для демонстрации результатов столкновения.
 - Документация системы должна удовлетворять требованиям федерального стандарта 1234.56
 - Система АЕФ не обязательно должна анализировать данные ДТП.

Свойства детальных требований

- Прослеживание – возможность отображать каждое требование на соответствующие части проекта и программы



-
- Пригодность к тестированию и однозначность
Пример: Система должна показывать разницу в зарплате между зарплатой клиента и средней зарплатой в мире для той же специальности.

Система будет показывать разницу в зарплате между зарплатой клиента и предполагаемой средней зарплатой для той же специальности согласно цифрам, опубликованным на сайте ООН на время запроса.

-
- Приоритет: классификация требований – важные, желательные и необязательные
 - Полнота требований
 - Состояние ошибки
 - Согласованность: между требованиями не имеется противоречий

Описание детальных требований

■ Диаграммы последовательностей – графическое представление передачи управления

*1 по роности
вариантов
использования*

Инициализировать

1. Система показывает главный персонаж игрока
2. Система показывает окно для установки характеристик персонажа
3. Игрок назначает характеристики своему главному персонажу
4. Игрок направляет персонаж
5. Система перемещает главный персонаж игрока



Диаграмма последовательности для варианта использования «Инициализировать»

Организация детальных требований

- По основным свойствам – группировка требований по различным свойствам программы;
- По режиму: пример – системы управления радарными могут иметь тренировочный, нормальный и аварийный режимы;
- По вариантам использования: детальное требование – часть варианта использования;
- По классу;

- По иерархии функций – разбиение программы на множество высокоуровневых функций и последующего разбиения их на подфункции.

Пример: приложение «Домашний бюджет» -

1. функции проверки (функции чековой книжки, баланса счета, составления отчетов и т.д.), 2. функции сбережений, 3. функции инвестирования

- По состояниям – указание детальных требований, применимых к каждому состоянию.