

# Классические методы анализа

**Структурный анализ** — один из формализованных методов анализа требований к ПО.

Автор этого метода — **Том Де Марко (1979)**.

В этом методе программное изделие рассматривается как преобразователь информационного потока данных. Основной элемент структурного анализа — **диаграмма потоков данных**.

# Диаграммы потоков данных

**Диаграмма потоков данных** — графическое средство для изображения информационного потока и преобразований, которым подвергаются данные при движении от входа к выходу системы.

# Элементы диаграммы потоков данных

Внешний  
объект

Источник или потребитель  
информации

Процесс

Преобразователь (принимает  
и обрабатывает данные)

Скорость



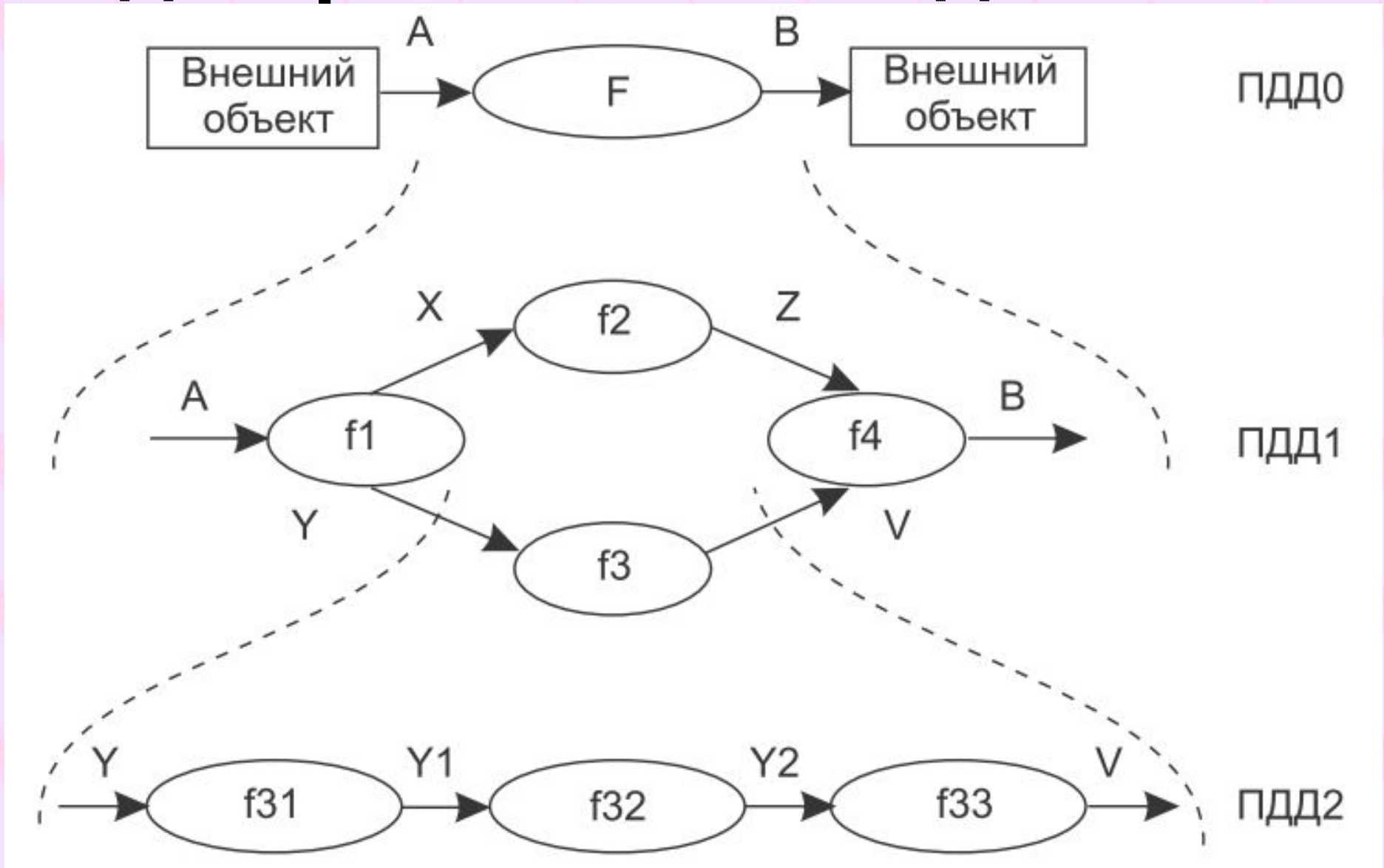
Поток данных  
(должен иметь метку)

Хранилище  
данных

Запоминает информацию,  
используемую преобразователем



# Система взаимосвязанных диаграмм потоков данных



# Словарь требований

**Словарь требований (данных)** содержит описания потоков данных и хранилищ данных.

1. Имя (основное имя элемента данных, хранилища или внешнего объекта).
2. Прозвище (Alias) — другие имена того же объекта.
3. Где и как используется объект — список процессов, которые используют данный элемент, с указанием способа использования (ввод в процесс, вывод из процесса, как внешний объект или как память).

# Словарь требований

4. Описание содержания — запись для представления содержания.
5. Дополнительная информация — дополнительные сведения о типах данных, допустимых значениях, ограничениях и т. д.



# **Спецификация процесса**

**Спецификация процесса** — это описание преобразователя.

Спецификация поясняет: ввод данных в преобразователь, алгоритм обработки, характеристики производительности преобразователя, формируемые результаты.

# Расширения для систем реального времени

П. Вард и С. Меллор

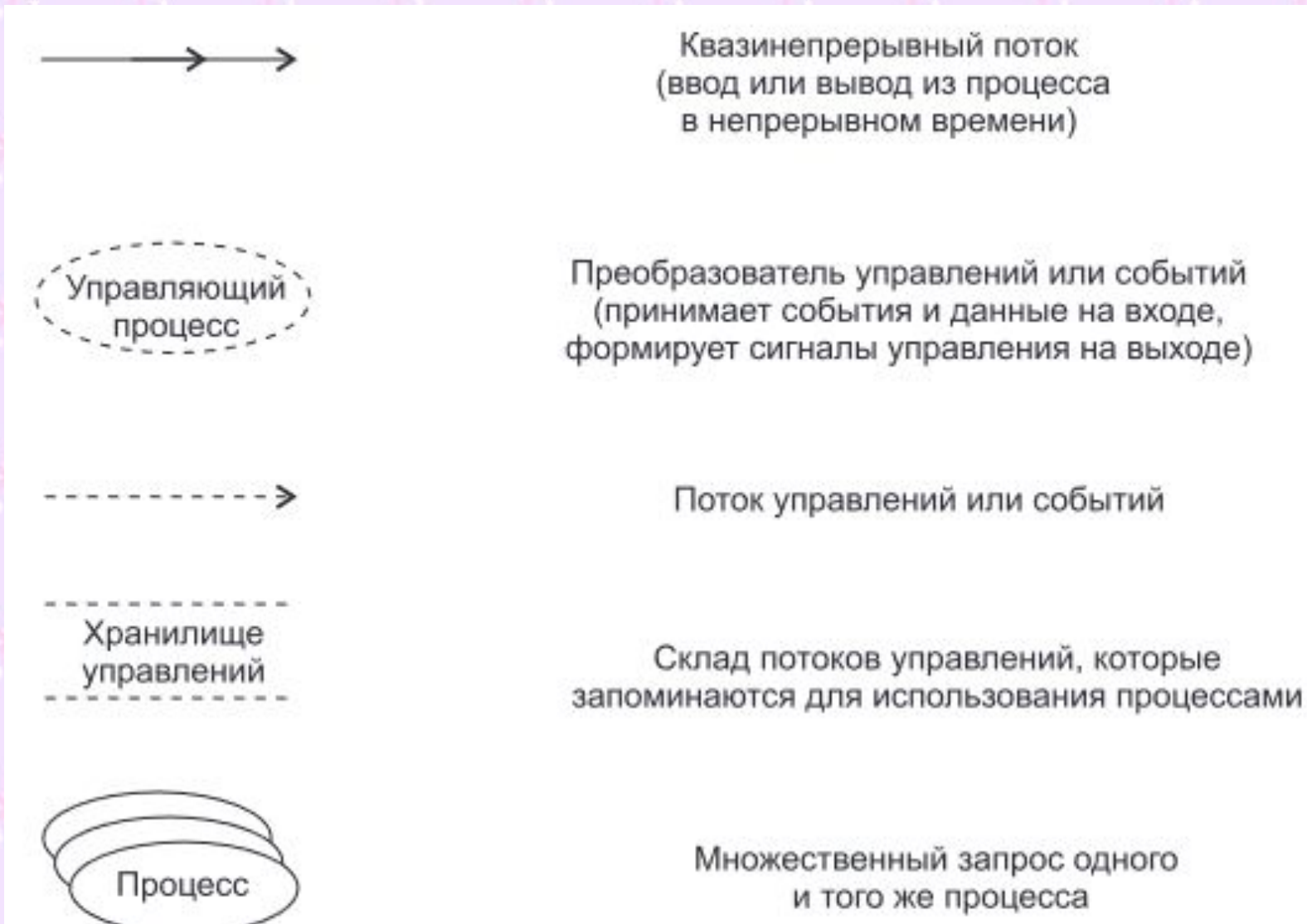




# Требованиям систем реального времени

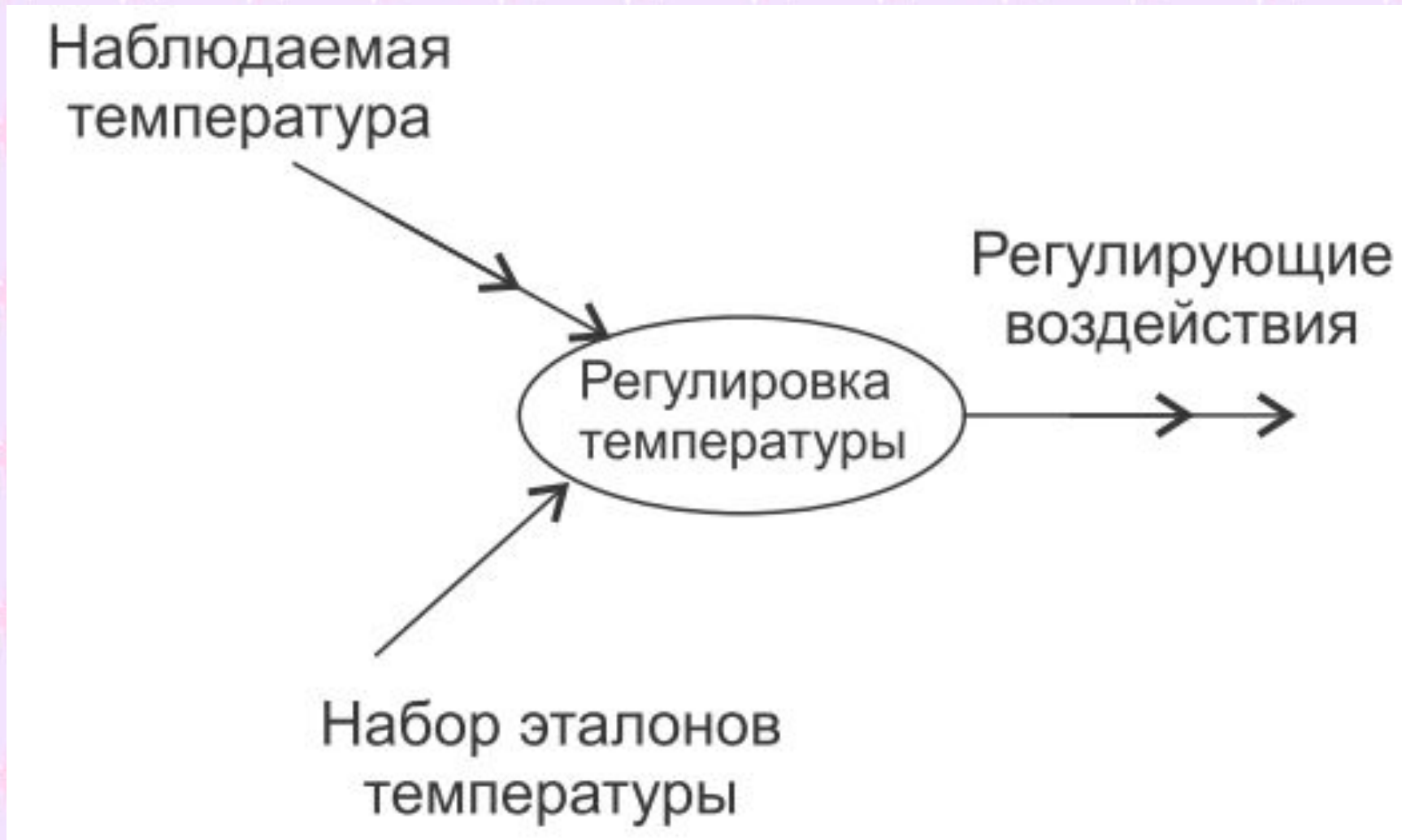
- 1) информационный поток накапливается или формируется в непрерывном времени;
- 2) фиксируется управляющая информация. Считается, что она проходит через систему и связывается с управляющей обработкой;
- 3) допускается множественный запрос на одну и ту же обработку (из внешней среды).

# Расширения диаграмм для систем реального времени



# Пример

Модель ПО для системы слежения за газовой турбиной





# Пример

## Модель ПО для управления роботом



# Диаграммы управляющих потоков

Д. Хетли и И. Пирбхаи

**Диаграмма управляющих потоков  
содержит:**

- обычные преобразователи (управляющие преобразователи исключены вообще);
- потоки управления и потоки событий (без потоков данных).



# Композиция модели обработки и управления





# Композиция модели обработки и управления

**Спецификация процесса ПСПЕЦ может  
включать:**

- 1) поясняющий текст (обязательно);
- 2) описание алгоритма обработки;
- 3) математические уравнения;
- 4) таблицы;
- 5) диаграммы.

# Модель системы регулирования давления космического корабля

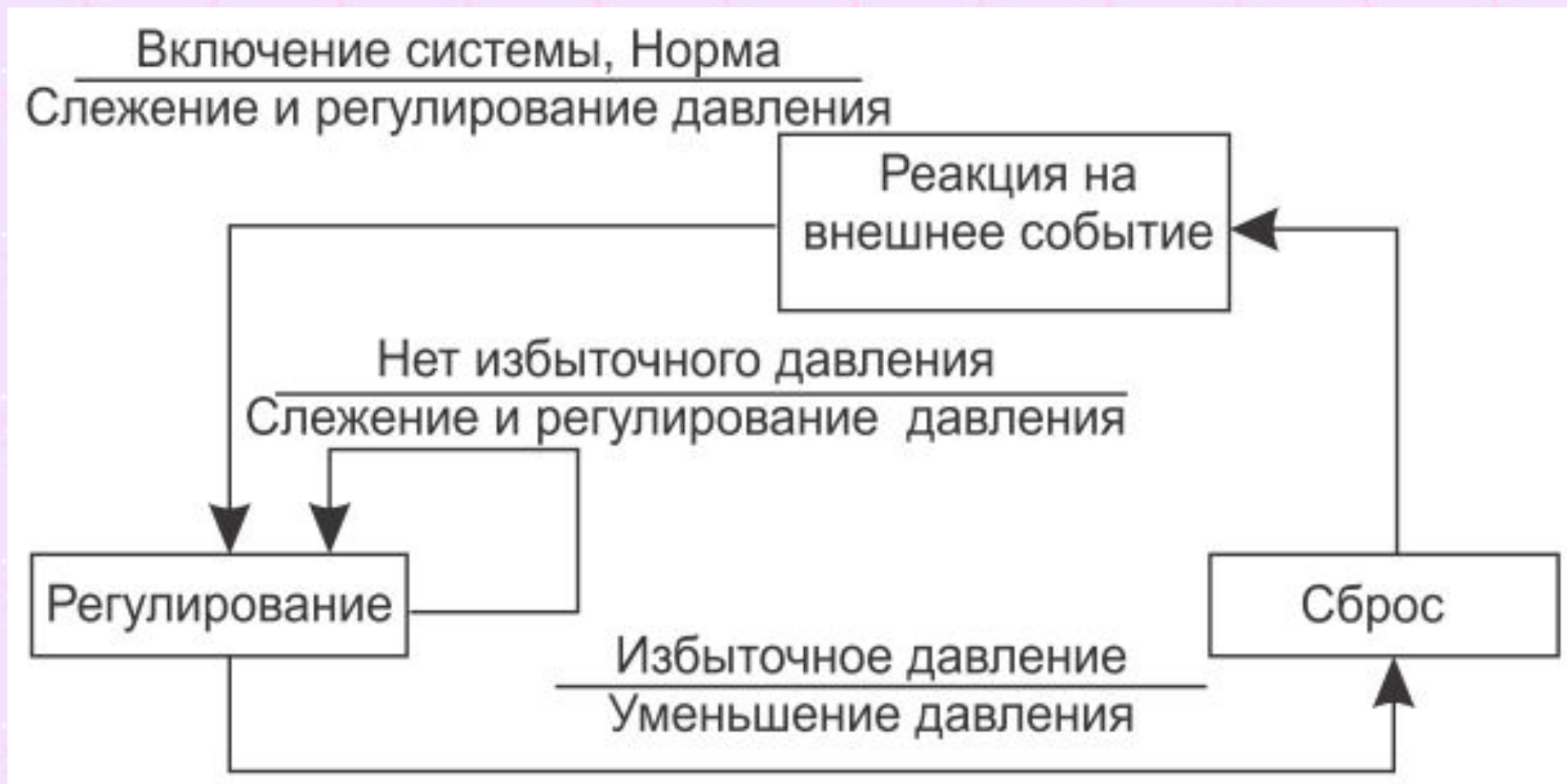


# Таблица активации процессов

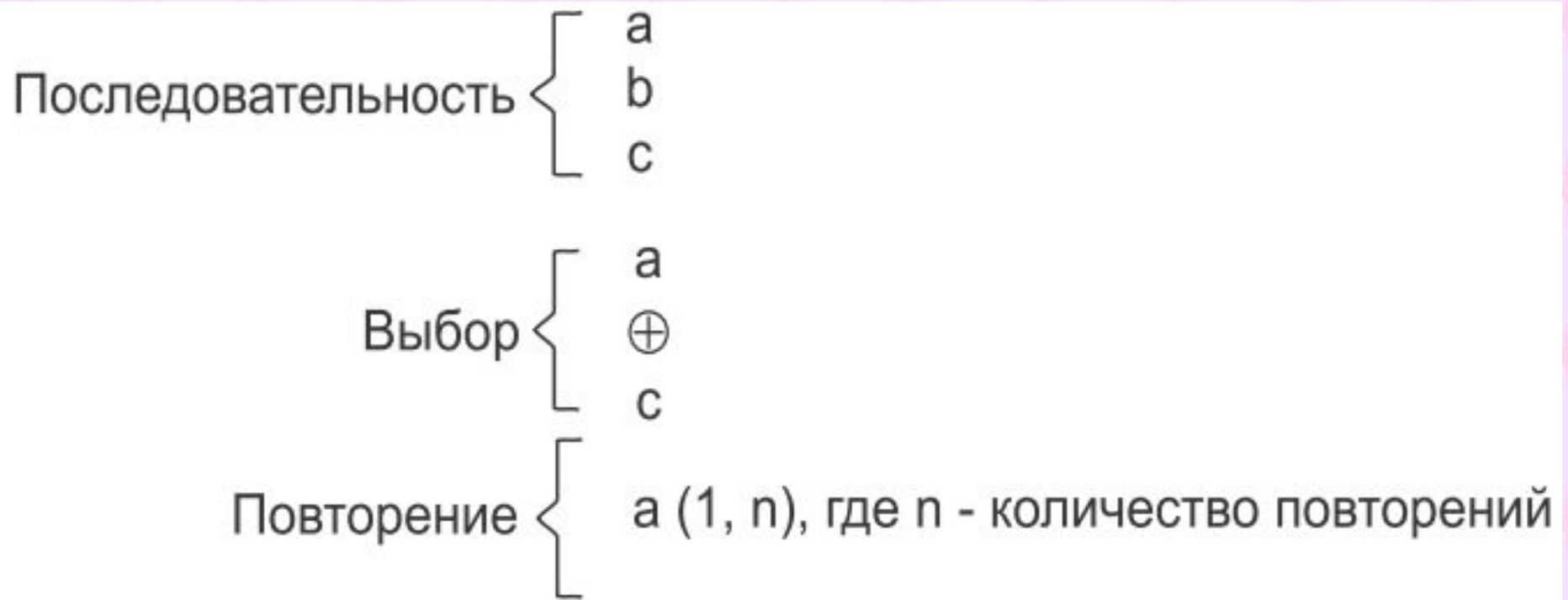
<b>Входные события:</b>			
Включение системы	1	0	0
Избыточное давление	0	1	0
Норма	0	0	1
<b>Выходные события:</b>			
Тревога	0	1	0
Работа	1	0	1
<b>Активация процессов:</b>			
Слежение и регулирование давления	1	0	1
Уменьшение давления	0	1	0



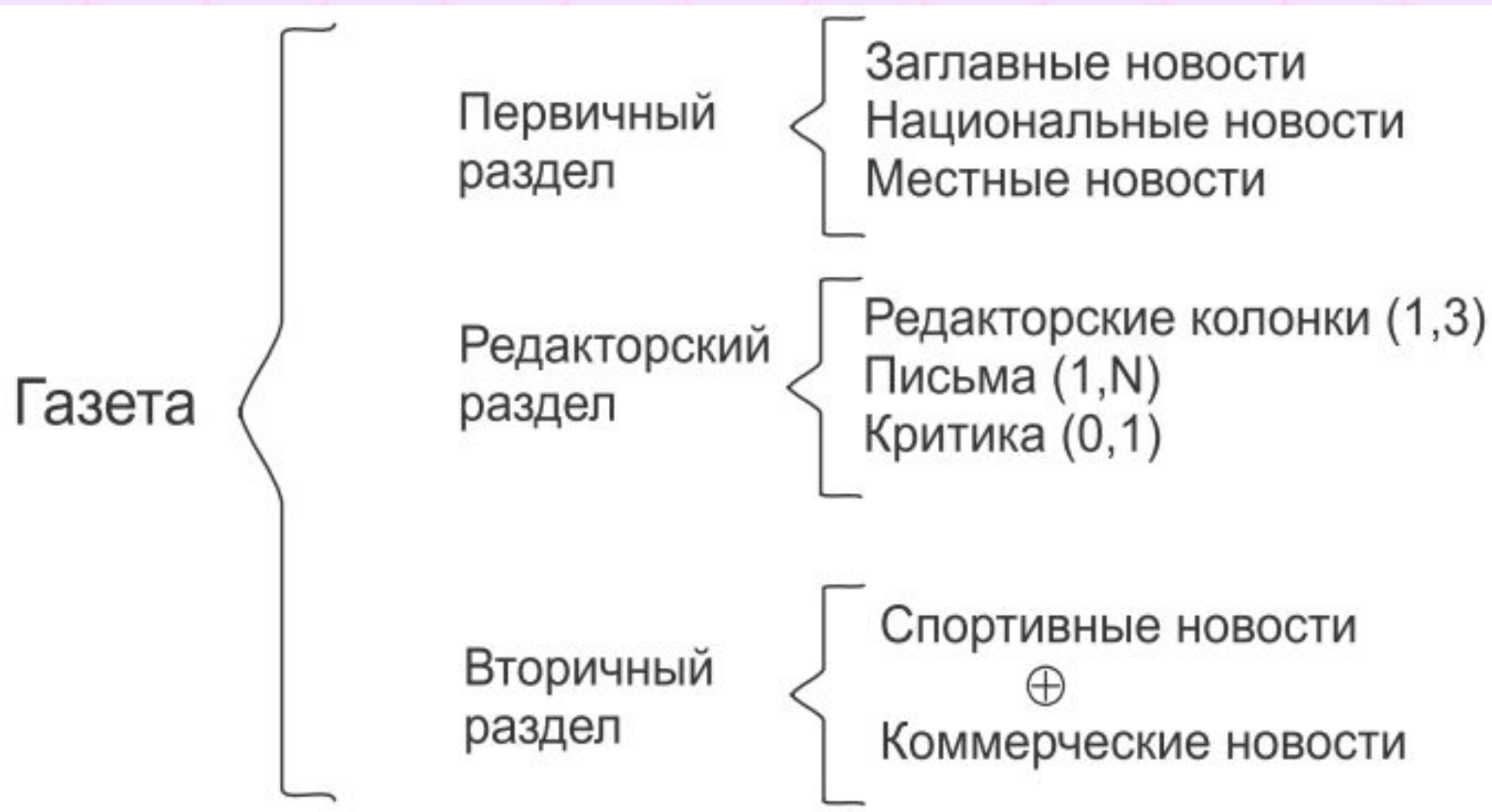
# Диаграмма переходов-состояний



# Базовые элементы в диаграммах Варнье



# Структура газеты в виде диаграммы Варнье





# Методика Джексона

1. Объект-действие. Определяются объекты — источники или приемники информации и действия — события реального мира, воздействующие на объекты.
2. Объект-структура. Действия над объектами представляются диаграммами Джексона.
3. Начальное моделирование. Объекты и действия представляются как обрабатывающая модель. Определяются связи между моделью и реальным миром.

# Методика Джексона

4. Доопределение функций. Выделяются и описываются сервисные функции.

5. Учет системного времени. Определяются и оцениваются характеристики планирования будущих процессов.

6. Реализация. Согласование с системной средой, разработка аппаратной платформы.

# Шаг объект-действие

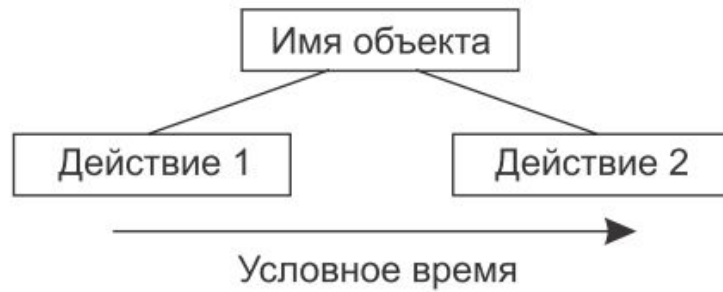
При нажатии кнопки:

- если транспорт на остановке, то студенты заходят в него и перемещаются на другую остановку;
- если транспорт в пути, то студенты ждут прибытия на другую остановку, приема студентов и возврата на текущую остановку;
- если транспорт на другой остановке, то он ее покидает, прибывает на текущую остановку и принимает студентов, нажавших кнопку.



# Шаг объект-структура

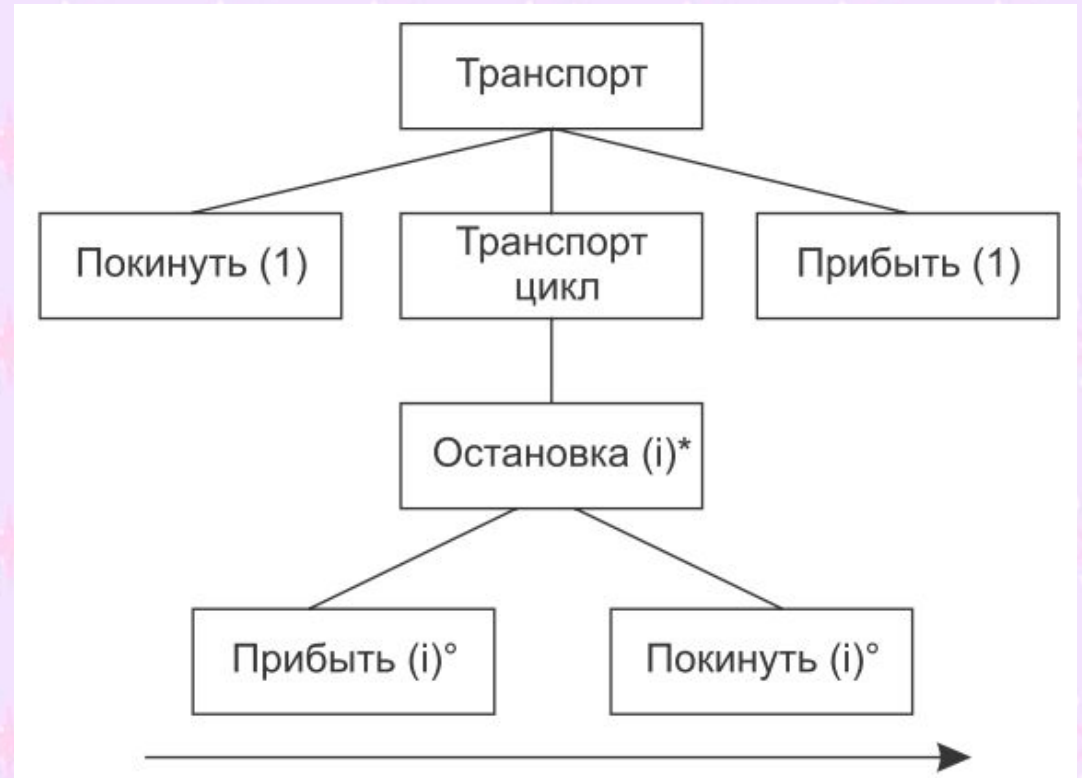
Действие-последовательность



Действие-выбор



Действие-итерация



# Шаг начального моделирования

1. Соединение потоком данных



2. Соединение по вектору состояний



# Шаг начального моделирования

## КНОПКА-1

```
читать BD;  
НАЖАТЬ цикл ПОКА BD  
    нажать;  
    читать BD;  
конец НАЖАТЬ;  
конец КНОПКА-1;
```

## ТРАНСПОРТ-1

```
опрос TSV;  
ЖДАТЬ цикл ПОКА ПРИБЫЛ(1)  
    опрос TSV;  
конец ЖДАТЬ;  
покинуть(1);  
ТРАНЗИТ цикл ПОКА УБЫЛ(1)  
    опрос TSV;  
конец ТРАНЗИТ;  
ТРАНСПОРТ цикл  
    ОСТАНОВКА  
        прибыть(i);  
        ЖДАТЬ цикл ПОКА ПРИБЫЛ(i)  
            опрос TSV;  
        конец ЖДАТЬ;  
        покинуть(i);  
        ТРАНЗИТ цикл ПОКА УБЫЛ(i)  
            опрос TSV;  
        конец ТРАНЗИТ;  
    конец ОСТАНОВКА;  
конец ТРАНСПОРТ;  
прибыть(1);  
конец ТРАНСПОРТ-1;
```



# Структурная диаграмма модели транспорта

