

Нотация ЕРС была разработана в 90х годах XX века
(немецкий профессор Вильгельм-Август Шеер)

Нотация ЕРС



Подзаголово
к

Введение

В настоящее время существует множество различных принципов графического представления бизнес-процессов, именуемых нотациями.

- **Разные задачи.** Не все нотации одинаково удобны для решения различных задач. Например, нотация может быть удобна для бизнес-процесса верхнего уровня и совсем не удобной для описания рабочего процесса.
- **Разные разработчики** таких нотаций. В разное время разные разработчики пытались придумать новые принципы описания схем. Иногда в процессе эволюции такие нотации стали как бы параллельными, т.е. выглядят по разному, а задачи решают одинаковые.
- **Стремление выделиться.** Это когда по непонятным причинам вдруг появляется новая нотация, не имеющая в себе ничего выдающегося, но, почему-то продвигающаяся ее создателем как совершеннейшее ноу-хау. Такое происходит до сих пор.

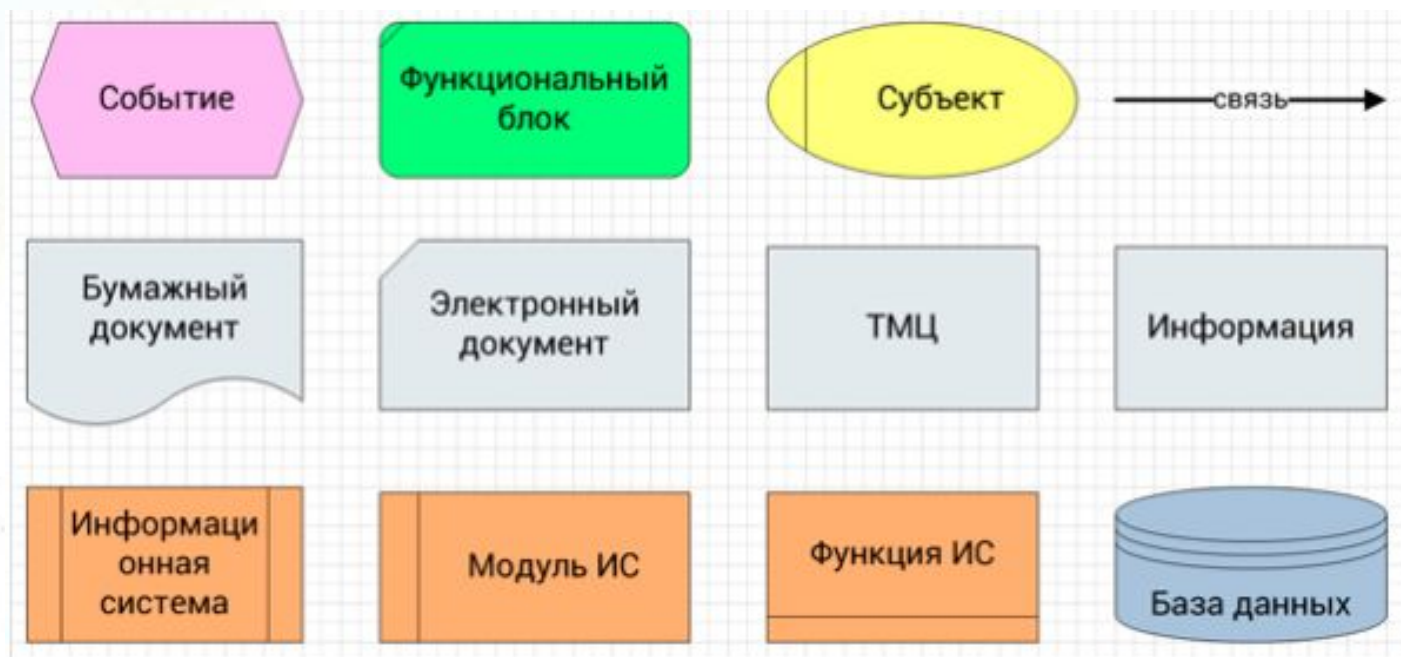
Введение

Нотация ЕРС (Event-Driven Process Chain – событийная цепочка процессов) используется для **описания процессов** нижнего уровня.

Диаграмма процесса в нотации ЕРС, представляет собой упорядоченную комбинацию **событий и функций**.

Для каждой функции могут быть определены начальные и конечные события, участники, исполнители, материальные и документальные потоки, сопровождающие её, а также проведена декомпозиция на более низкие уровни.

Основные элементы



Основные элементы

Функция

Блок представляет собой функцию – действие или набор действий, выполняемых над исходным объектом с целью получения заданного результата. Внутри блока помещается наименование функции.

Событие

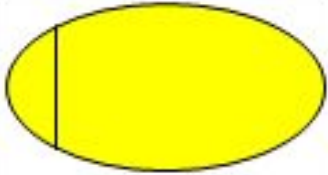


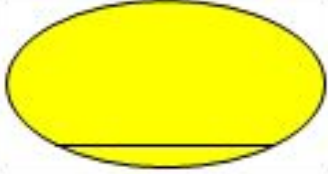
Событие – состояние, которое является существенным для целей управления бизнесом и оказывает влияние или контролирует дальнейшее развитие одного или более бизнес-процессов. Элемент отображает события, активизирующие функции или порождаемые функциями.

Связку событие – функция – событие принято отображать **сверху вниз** в одну линию либо **слева направо**.

Направление цепочки указывается связывающими линиями со стрелками.



Символы исполнителей

3.1		Организационная единица (Organizational unit)
3.2		Должность, тип исполнителя (Position, Role, Person type)
3.3		Исполнитель (Person)
3.4		Местоположение (Location)



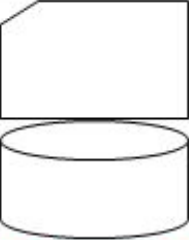

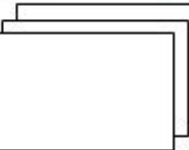


Структурное подразделение, которому поручено выполнение действия (фирма, организация, отдел, служба).

Должность исполнителя или роль субъекта, которому поручено выполнение действия. Составная часть организационной единицы.

Конкретный исполнитель, которому поручено выполнение действия (имя исполнителя). Экземпляр должности.

Местоположение объекта, выполнения действия или возникновения события (фирма, организация, отдел, служба, завод, здание, комната, адрес).

Символы объектов

	ТМЦ, информация (Information, Entity)
	Документ (Document)
	Файл, база данных (File, Database)
	Контекстные данные, кластер (Cluster)
	Набор объектов, картотека (CardFile)
	Сообщение (Message)
	Вход / Выход, продукт (Input / Output, Product)

Товарно-материальные ценности (ТМЦ) или информация, используемые или получаемые в результате действий. Может использоваться вместо элемента «Вход / Выход».

Информация, представляемая не в компьютерном виде (на бумаге, пленках, слайдах).

Информация, представляемая в компьютерном виде (файл, таблица, БД, электронный документ).

Набор данных, необходимых для выполнения функции (модель, диаграмма, заказ).

Набор ТМЦ или документов.

Требование отправителя к получателю на создание ТМЦ, предоставление информации или оказание услуги.

Объект, необходимый для выполнения процесса (план работ, заказ, материалы) или являющийся результатом процесса (документация, изделие, выполненная услуга).

Логические символы




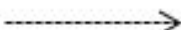


	Логическое «И» (AND)
	Логическое «ИЛИ» (OR)
	Исключающее «ИЛИ» (XOR)

При слиянии выполнение (наступление) следующего действия (события) возможно **только при завершении выполнения (наступления) всех предыдущих действий (событий)**. При ветвлении должны выполняться (наступать) **все последующие действия (события)**.

При слиянии выполнение (наступление) следующего действия (события) возможно при завершении выполнения (наступления) **хотя бы одного из предыдущих действий (событий)**. При ветвлении должно выполняться (наступить) **хотя бы одно из последующих действий (событий)**.

При слиянии выполнение (наступление) следующего действия (события) возможно при завершении выполнения (наступления) **только одного из предыдущих действий (событий)**. При ветвлении должно быть выполнено (наступить) **только одно из последующих действий (событий)**.

Линии и стрелки

	Поток управления (Control Flow Arrow)
	Организационный поток (Organizational Flow Arrow)
	Поток ресурсов (Resources Flow Arrow)
	Информационный поток (Information Flow Arrow)
	Поток информационных услуг (Information output Flow Arrow)
	Поток ТМЦ (Material output Flow Arrow)

Задаёт последовательность (до-после) возникновения событий и выполнения действий.

Иерархическая связь между однотипными элементами (организационная единица – должность – персона).

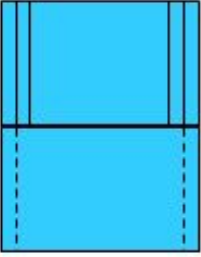

Связь между действием и ресурсами, необходимыми для его выполнения (организационными единицами, персонами, приложениями, модулями и т.п.).

Связывает действие и элемент, являющийся источником и/или приемником информации (приложение, кластер).

Связь между действием и информационным входом/выходом.

Связь между действием и материальным входом/выходом.

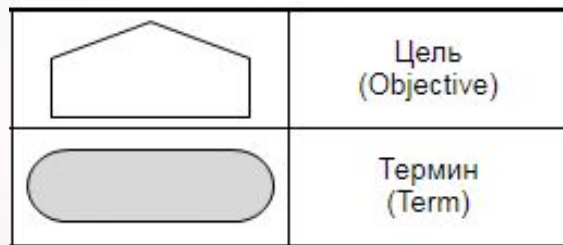
Символы ПО

	Приложение, прикладная система (Application)
	Модуль (Module)

Информационная система (программный продукт), с помощью которой выполняется функция.

Составная часть информационной системы.

Дополнительные символы



Внешний (по отношению к текущей диаграмме) процесс или функция. Используется для указания взаимосвязи процессов:

- обозначает предыдущий или следующий процесс по отношению к текущему процессу (диаграмме);
 - обозначает процесс, откуда поступил или куда передается объект.
- Цель выполнения процесса (высокое качество, низкая себестоимость, сокращение издержек, увеличение прибыли).

Используется:

- для обозначения данных, передаваемых между процессами или обрабатываемых при выполнении процессов (Техническое задание, Форма № 1, Ведомость ЦДЛ №3, пин-код);
- для обозначения статусов бумажных или электронных документов (подписанный, утвержденный).

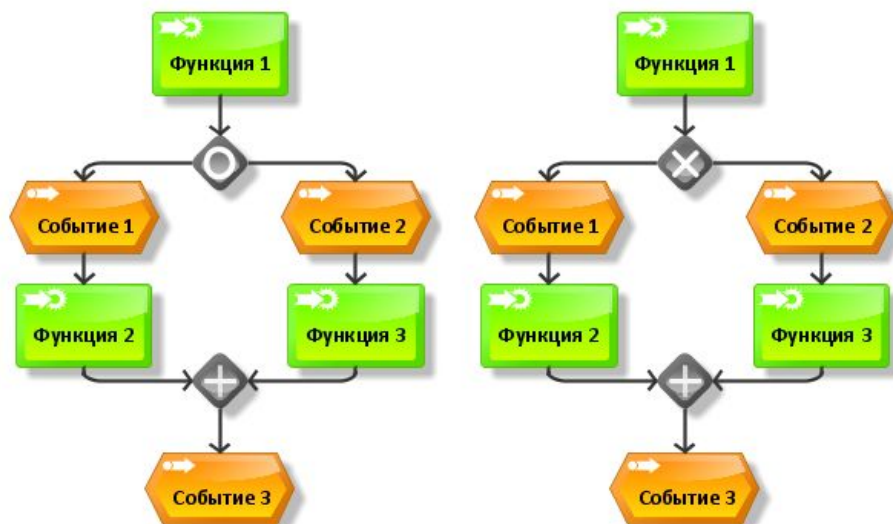
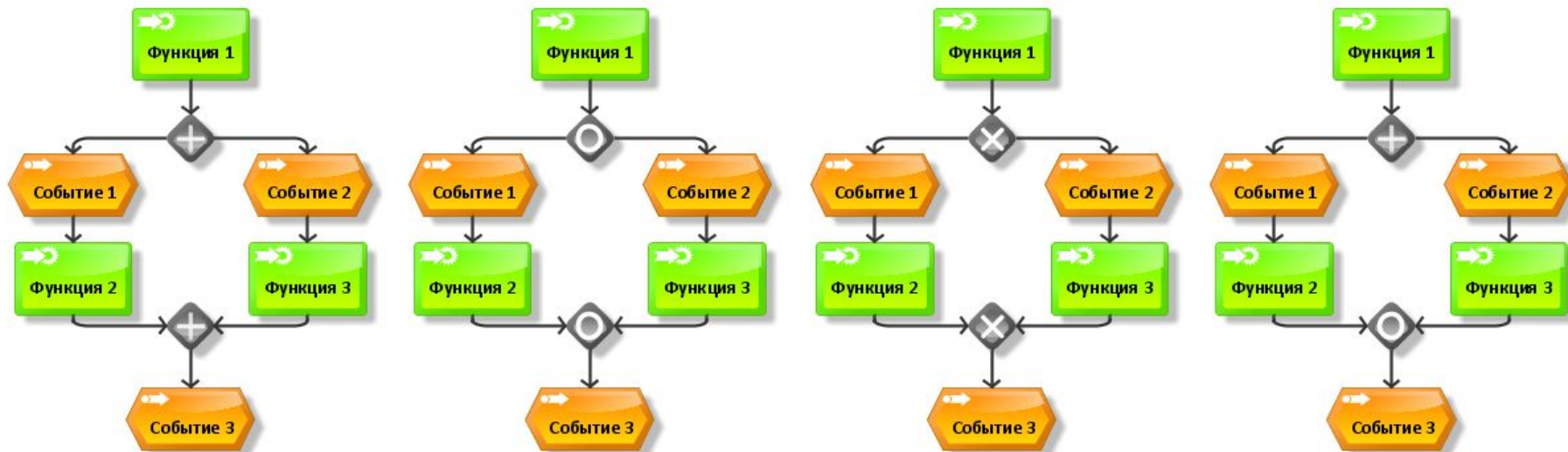
Правила и рекомендации

1. Диаграмма функции ЕРС должна начинаться как минимум одним стартовым событием (стартовое событие может следовать за интерфейсом процесса) и завершаться как минимум одним конечным событием (конечное событие может предшествовать интерфейсу процесса).
2. События и функции по ходу выполнения процесса должны **чередоваться**.
3. Рекомендуемое количество функций на диаграмме - не более **20**. Если количество функций диаграммы значительно превышает 20, то существует вероятность, что неправильно выделены процессы на верхнем уровне и необходимо произвести корректировку модели.
4. События и функции должны содержать строго по **одной входящей и одной исходящей связи** (поток управления), отражающей ход выполнения процесса.
5. На диаграмме **не должны** присутствовать элементы **без единой связи**. Исключение может составлять элемент «цель» всего процесса (диаграммы).

Правила и рекомендации

6. События и логические операторы, окружавшие функцию на вышележащей (родительской) диаграмме, должны быть начальными/результатирующими событиями и операторами на диаграмме декомпозиции функции.
7. Каждый оператор слияния должен обладать минимум двумя входящими связями и только одной исходящей, оператор ветвления - только одной входящей связью и минимум двумя исходящими. Операторы не могут обладать одновременно несколькими входящими и исходящими связями.
8. Логические операторы могут объединять или разветвлять **только функции** или **только события**. Одновременное объединение/ветвление функции и события невозможно.
9. Логический оператор, разветвляющий ветви, и оператор, объединяющий эти ветви, должны совпадать. Допускается также ситуация, когда оператор ветвления «И», оператор объединения – «ИЛИ».

Допустимые и недопустимые ситуации



Алгоритм построения

- **Шаг первый.** Сначала мы определяем, что у нас есть и чего мы хотим – **граничные события**.
- На **втором шаге** «разбавляем» граничные события **действиями и соответствующими им промежуточными событиями**.
- На третьем присоединяем **документы и (или) информацию**, которая необходима для выполнения каждого этапа (входы) и документы, которые являются результатами работы на каждом этапе (выходы). Кроме этого к работам добавляются **связи с исполнителями** с обозначением ролей. Самая распространенная роль – «выполняет», но могут быть и другие: «утверждают результат», «отвечает за техническую часть», «должен быть уведомлен о нестандартном завершении» и т.п.
- На **4-5 шагах** мы оцениваем полноту и качество схемы, анализируем, все ли варианты исполнения процесса учтены в схеме. Если же есть не совсем понятные функции, требующие расшифровки, делаем из них **подпроцессы** и повторяем весь процесс, но уже в отношении этой отдельной работы.

Преимущества и недостатки



- Позволяет отразить все значимые организационные элементы на одной схеме (в отличие от простой блок-схемы).
- Может использоваться на разных уровнях модели – описывать как глобальные процессы, так и делать детальные инструкции за счет того, что каждый функциональный блок может стать подпроцессом.
- Легко делать сложные распараллеливания процесса, так как можно ввести любое количество событий в один ряд.



- Необходимость придумывать события на каждые даже незначительные действия сильно усложняет схему.
- Вероятны организационные разрывы из-за неудобного отслеживания назначений.
- Качественное прописывание входов и выходов приводит к перегрузке схемы прямоугольниками, стрелками, которые начинают пересекаться и тем самым еще сильнее усложняют восприятие схемы.
- При распараллеливании работ очень сложно отразить исполнителей. Если один человек выполняет группу функций, картинка усложняется стрелками. Если присутствуют несколько исполнителей или мы не хотим рисовать длинные стрелки, приходится дублировать «овальчики» с исполнителями.

Средства для моделирования

