

## Действия

- Задача** - единица работы. Если задача помечена символом , то задача является подпроцессом и может быть детализирована.
- Транзакция** - набор логически связанных действий. Для транзакции может быть определен протокол выполнения.
- Событийный подпроцесс** помещается внутри другого процесса. Он начинает выполняться, если инициируется его начальное событие. Событийный подпроцесс может прерывать родительский подпроцесс или выполняться параллельно с ним.
- Вызывающее действие** является точкой входа для глобально определенного подпроцесса, который повторно используется в данном процессе.

### Маркеры действий

Маркер отражает поведение действия во время выполнения:

-  Маркер подпроцесса
-  Маркер цикла
-  Маркер параллельных множественных экземпляров (МЭ)
-  Маркер последовательных множественных экземпляров (МЭ)
-  Маркер ad hoc
-  Маркер компенсации

### Типы задач

Тип определяет природу действия, которое будет выполнено:

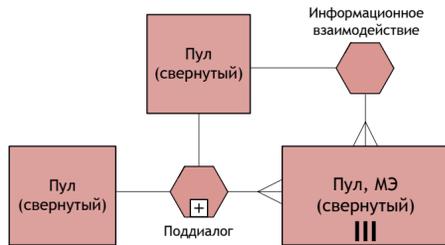
-  Задача отправки сообщения
-  Задача получения сообщения
-  Пользовательская задача
-  Неавтоматизированная задача
-  Задача-бизнес-правило
-  Задача-сервис
-  Задача-сценарий

- Поток управления** - определяет порядок выполнения действий.
- Поток по умолчанию** - определяет ветвь процесса, выполняемую, когда все условия ветвления не выполнены.
- Условный поток** - связан с условием, определяющим будет ли выполнен данный поток.

## Диалоги

-  **Информационное взаимодействие** задает цепочку логически связанных обменов сообщениями. Если информационное взаимодействие помечено символом , то оно может быть детализировано.
-  **Связь** соединяет информационное взаимодействие с одним участником
-  **Разветвляющаяся связь** соединяет информационное взаимодействие с несколькими участниками

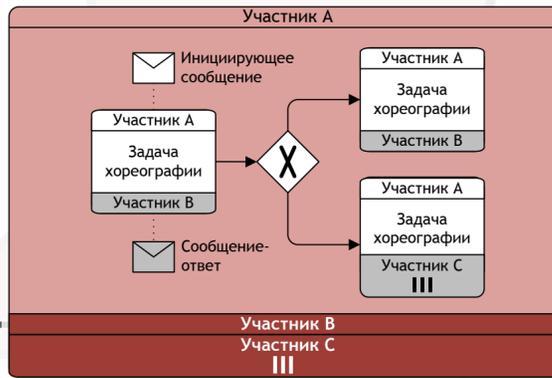
### Схема диалога



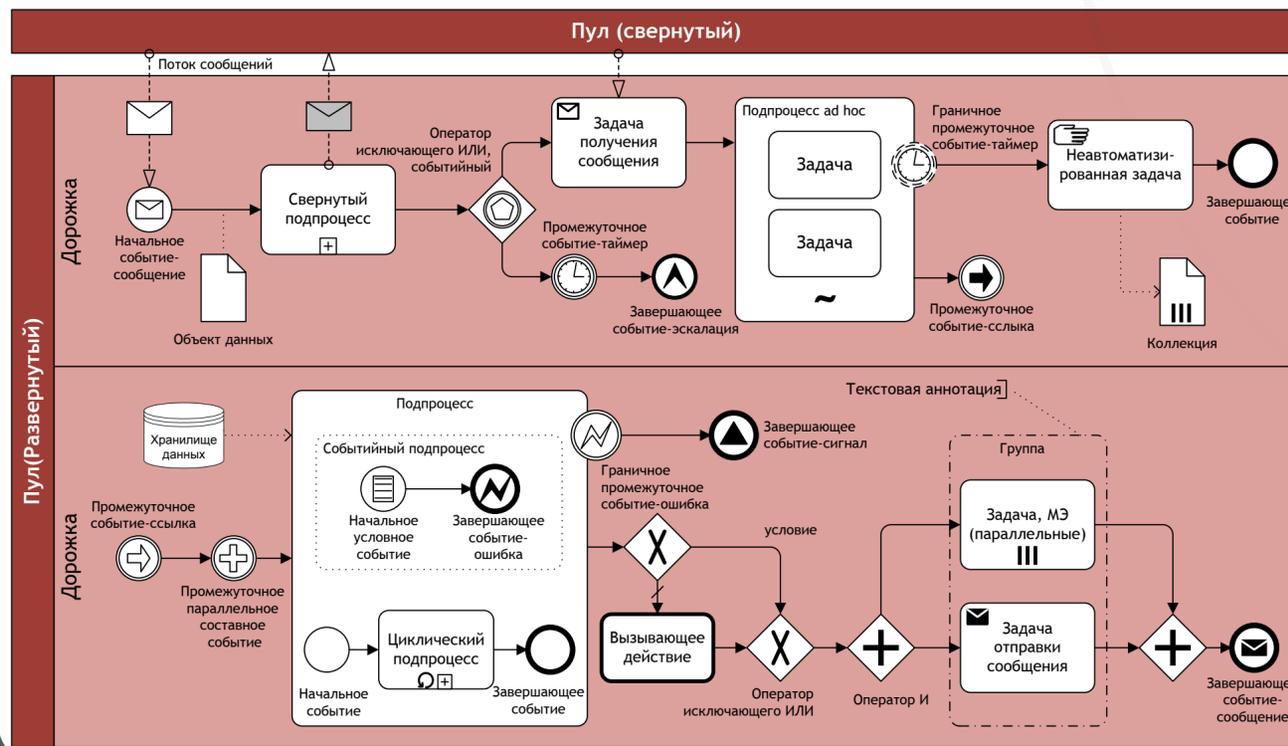
## Хореографии

- Участник А** (Task Choreography) - Задача хореографии (обмен сообщениями) между двумя участниками.
- Участник В** (Marker of multiple participants) - Маркер множественных участников показывает, что несколько участников одного типа участвуют в хореографии.
- Участник С** (Subprocess Choreography) - Подпроцесс хореографии содержит детализацию хореографии, включающую несколько взаимодействий.

### Схема хореографии



## Схема взаимодействия



## События

События	Начальные			Промежуточные			Завершающие	
	Верхнеуровневые	Прерывающие событийный подпроцесс	Не прерывающие событийный подпроцесс	Обрабатывающие	Граничные прерывающие	Граничные не прерывающие	Генерирующие	Генерирующие
<b>Простое:</b> нетипизированное событие, обычно показывающее начало или окончание процесса.								
<b>Сообщение:</b> получение и отправка сообщений.								
<b>Таймер:</b> циклические события, моменты времени, временные периоды и таймауты.								
<b>Эскалация:</b> перенос рассмотрения вопроса на более высокий уровень организационной иерархии								
<b>Условное:</b> реакция на изменение бизнес-условий или интеграция бизнес-правил.								
<b>Ссылка:</b> пара соответствующих ссылок эквивалента потока последовательности.								
<b>Ошибка:</b> генерация и обработка заданного типа ошибок.								
<b>Отмена:</b> обработка отмены транзакции или инициирование отмены.								
<b>Компенсация:</b> обработка или инициирование компенсации.								
<b>Сигнал:</b> передается между процессами и может обрабатываться многими получателями.								
<b>Составное:</b> обработка одного события из множества или генерация всех определенных событий.								
<b>Параллельное составное:</b> обработка всего множества параллельных событий.								
<b>Останов:</b> вызывает немедленное прекращение выполнения процесса.								

## Данные

- Входные данные** - внешний вход для процесса. Действия могут использовать эти данные.
- Выходные данные** - результат выполнения процесса.
- Объект данных** представляет информацию, которая обрабатывается в ходе процесса, например, документ или письмо.
- Коллекция объектов данных** представляет группу объектов, несущих информацию, например, список заказанных товаров.
- Хранилище данных** - объект, который процесс может использовать для записи и извлечения данных, например, база данных. Хранилище данных позволяет хранить данные после окончания жизненного цикла экземпляра процесса.
- Сообщение** позволяет явно показать передачу информации в ходе общения двух участников.

## Логические операторы

- Оператор исключающего ИЛИ, управляемый данными** - При ветвлении направляет поток лишь по одной из исходящих ветвей. При синхронизации потоков оператор ожидает завершения одной входящей ветви и активирует исходящий поток управления.
- Оператор исключающего ИЛИ, событийный** - Предшествует только событиям обработки или заданиям-обработчикам сообщений. Поток управления направляется по той ветви, где событие произошло раньше.
- Оператор И** - При разделении на параллельные потоки все ветви активируются одновременно. При синхронизации параллельных ветвей оператор ждет завершения всех входящих ветвей и затем активирует исходящий поток.
- Оператор ИЛИ** - При ветвлении активируется одна или более ветвей. При слиянии все выполняющиеся входящие ветви должны быть завершены.
- Оператор исключающего ИЛИ, событийный (создает новый экземпляр)** - Наступление каждого из последующих событий создает экземпляр процесса.
- Оператор И, событийный (создает новый экземпляр)** - Наступление всех последующих событий создает экземпляр процесса.
- Сложный оператор** - Моделирует сложные условия ветвления и слияния.

## Роли

- Пулы (участники) и дорожки** отражают распределение обязанностей. Пул или дорожка обозначает организацию, роль или систему. Дорожки позволяют иерархически делить пулы и другие дорожки.
- Поток сообщений** описывает информационный поток между участниками процесса. Поток сообщений может присоединяться к пулам, действиям или событиям-сообщениям.
- Порядок обмена сообщениями** может быть задан при помощи потока сообщений и потока управления.