

Tarefa

Tarefas (*Tasks*) são unidades atômicas de trabalho que o processo deve executar.

Atividade de Chamada

Atividades de Chamada (*Call Activity*) são uma referência síncrona a um processo reutilizado no processo atual.

Subprocesso Compactado

Subprocesso Compactado (*Collapsed Subprocess*) são uma representação compactada do subprocesso onde marcador (+) o torna expandido.

Subprocesso de Transação

Transações (*Transactions*) são um conjunto de atividades, logicamente relacionadas que podem seguir um protocolo transacional.

Subprocesso de Evento

Subprocessos de Evento (*Event Subprocess*) são contidos num processo e ativados por um evento de início, sendo executado até o final ou enquanto o seu contexto estiver ativo.

Subprocesso Expandido

Subprocesso Expandido (*Expanded Subprocess*) são um agrupamento ou escopo de atividades embutidas no processo.

Tipos de tarefas

Enviar

Receber

Usuário

Manual

Regra de Negócio

Serviço

Script

Marcadores

Subprocesso

Repetição sequencial

Instâncias múltiplas em paralelo

Instâncias múltiplas sequencial

Atividades em ordem arbitrária (*ad hoc*)

Compensação

Exclusivo com Base em Dados

Desvio Condicional Exclusivo (*Exclusive Gateway*) avaliam cada ramificação em separado e desvia para o primeiro caminho que a condição for verdadeira. É recomendada uma convergência exclusiva explícita (não obrigatória).

Exclusivo com Base em Eventos

Desvio Condicionado por Evento Exclusivo (*Event-based Gateway*) aguardam acontecer o primeiro evento para desviar ao seu caminho. É recomendada uma convergência exclusiva explícita (não obrigatória).

Inclusivo com Base em Dados

Ativação Inclusiva Condicional (*Inclusive Gateway*) Em um ponto de ramificação, após avaliar as condições, um ou mais caminhos são ativados. Em um ponto de convergência de fluxo, espera que todos fluxos de entrada ativos tenham completado para ativar o fluxo de saída.

Paralelo

Ativação Incondicional em Paralelo (*Parallel Gateway*) em um ponto de ramificação, todos os fluxos de saída são ativados simultaneamente. Em um ponto convergência de fluxos, espera que todos os caminhos de entrada completem, antes de disparar o fluxo de saída.

Complexo

Junção Complexa (*Complex Gateway*) Comportamento complexo de ramificação ou convergência que não pode ser capturado por outros tipos de desvio.

Esta guia foi criada com a ferramenta Camunda Modeler

cawemo

BPMN.io

BPMN 2.0 – Business Process Model and Notation								
	Captura						Acionamento	
	Início			Intermediários			Fim	
Eventos	O processo é iniciado por um evento.	O subprocesso de evento é iniciado e provoca a interrupção do processo pai.	O subprocesso de evento é iniciado e não provoca a interrupção do processo pai.	O processo continua somente se a captura do evento ocorrer.	A atividade é cancelada e o fluxo do processo é desviado para a sequência do evento.	A atividade não é cancelada e o fluxo do processo também sai na sequência do evento.	O processo aciona o evento e continua imediatamente.	O processo ou subprocesso aciona o evento e conclui.
Posição								
Simples (<i>None</i>): indicam pontos de início, fim e mudanças de estado.								
Mensagem (<i>Message</i>): recebimento e envio de mensagens.								
Temporal (<i>Timer</i>): ponto, instante, intervalo, e limite de tempo único ou cíclico.								
Escalável (<i>Escalation</i>): ativa a mudança para um nível mais alto de responsabilidade.								
Condicional (<i>Conditional</i>): reação a alterações nas condições de negócio ou regra.								
Conector (<i>Connector</i>): dois eventos associados são uma sequência de fluxo.								
Erro (<i>Error</i>): capturam ou acionam erro técnico ou de negócio pré-definido.								
Cancelamento (<i>Cancel</i>): acionam ou reagem a cancelamento de transação.								
Compensação (<i>Compensation</i>): tratamento ou ativação de ação de compensação.								
Sinal (<i>Signal</i>): emitem sinais entre processos e podem ser emitidos várias vezes.								
Múltiplo (<i>Multiple</i>): capturam um ou vários eventos; acionam todos eventos.								
Múltiplo Paralelo (<i>Parallel Multiple</i>): capturam todos eventos em paralelo.								
Final (<i>Terminate</i>): ativam a terminação imediata de uma instância de processo.								

Fontes de referência:
www.bpmn.de/poster | www.omg.org/bpmn/ | www.omg.org/spec/DMN | learn-dmn-in-15-minutes.com

Fluxos de Sequencia (*Sequence Flows*) definem a sucessão de transições de execução.

Fluxo Padrão (*Default Flow*) será executado quando todas as outras condições não forem satisfeitas. Podem ser usados para representar o caminho feliz (*happy path*) do processo.

Fluxo Condicional (*Conditional Flow*) contém uma condição na qual define quando este fluxo será executado.

Elementos da notação e modelo BPMN 2.0 para documentação

Fluxo de Mensagens (*Message Flow*) simboliza fluxos de informação que transpõem fronteiras internas e externas de uma organização. Podem ser conectados a piscinas, tarefas e eventos de mensagem.

Processo A

Processo B

Piscinas (*Pools*) e Raias (*Lanes*) representam as responsabilidades pelas atividades que podem ser organizações, papéis ou sistemas.

Objeto de Dados (*Data Object Reference*) representam todos os tipos de informações, independentemente de sua natureza física. Eles podem incluir documentos em papel, informações abstratas ou registros eletrônicos de dados.

Repositório de Dados (*Data Store Reference*) representam um armazenamento de dados que existe, independentemente da instância de processo específica e permanecerá disponível por muito tempo após o término da instância do processo.

Texto de Anotação (*Text Annotation*) é um artefato que pode ser associar a qualquer objeto de fluxo em mais de um ponto para contextualizar qualquer anotação de texto.

Agrupamento (*Group*) documentam um grupo de elementos do diagrama sem interferir na execução do processo. Evite confundir com subprocessos expandido e de evento.

Esta guia foi criada com a ferramenta Camunda Modeler

cawemo

BPMN.io

www.projeler.com.br

Avenida Carlos Gomes, 222, 8º andar

Porto Alegre - RS Tel 51 2139 5920

PROJELER

DMN 1.3 – Decision Model and Notation

Requisito de Decisão (DRD Decision Requirements Diagram)

Decisão

Conhecimento de Negócio

Dados de Entrada

Fonte de Conhecimento

Exigência de Informação

Exigência de Conhecimento

Exigência de Autoridade

Decisões (Decisions) determinam as saídas com a aplicação de entradas a lógicas de decisão.

Conhecimentos de Negócio (Business Knowledge) representam funções que encapsulam lógica de tomada de decisão reutilizáveis.

Dados de Entrada (Input Data) são insumos de informação exigidos por uma decisão.

Fontes de Conhecimento (Knowledge Sources) possuem a sabedoria e regulação para a tomada de decisão, tais como, normas, políticas, melhores práticas e conhecimento analítico.

Exigências de Informação (Information Requirements) representam uma obrigatoriedade de informação para as entradas de uma decisão.

Exigências de Conhecimento (Knowledge Requirements) são utilizadas para ligar a decisão a Conhecimento de Negócio quando necessário. Representam um refinamento dos requisitos de lógica de decisão.

Exigências de Autoridade (Authority Requirements) demonstram onde ir para descobrir como tomar uma decisão.

Lógica de Decisão (Decision Logic)

E

CONCLUSÃO


Tabela de Decisão (Decision Table)

Política de Acerto	Entradas (Facts)		Saídas (Results)		Anotações
	Expressão de Entrada 1	Expressão de Entrada 2..n	Nome da Saída 1	Nome da Saída 2..n	
OU (rules)	1	Valor de Entrada 1a	Valor de Entrada 2a	Resultado 1a	Resultado 2a
	2		Valor de Entrada 2b	Resultado 1b	Resultado 2b
	3	Valor de Entrada 1c		Resultado 1c	Resultado 2c

Indiferente

Políticas de Acerto (Hit Policy)

Acerto Simples (Single Hit)	U	Unique	Retorna somente um resultado (padrão quando implícito).
	A	Any	Muitas regras podem coincidir e terá a mesma saída.
	P	Priority	Uma saída é selecionada por uma lista priorizada.
	F	First	A primeira saída é selecionada pela ordem das regras.
Acertos Múltiplos (Multiple Hit)	O	Output	Resultados são ordenados a partir de uma lista de valores.
	R	Rule	Resultados são ordenados na ordem das regras.
	C	Collect	Lista de resultados é retomada em ordem arbitrária.
	C+, C<, C>, C#		Soma (sum), mínimo (min), máximo (max) e contador (count).



<https://mauriciobitencourt.com/tag/guia/>

FEEL – Friendly Enough Expression Language

Instruções Condicionais e de Laço

Exemplo // Resultado	Descrição
if (taxa < 5) then "baixo" else "alto" // true	Se taxa for menor que 5 então "baixo" senão "alto"
some i in [1, 2, 3, 4, 5] satisfies i > 4 // true	Testa se pelo menos um elemento satisfaz a expressão.
for i in [1, 2, 3, 4, 5] return i * i // [1, 4, 9, 16, 25]	Itera cada elemento para retornar o produto.

Funções de String

Exemplo // Resultado	Descrição
string length("Guia para Modelagem") // 19	Tamanho
upper case("Guia BPMN 2.0") // GUIA BPMN 2.0	Caixa alta
lower case("Guia BPMN 2.0") // guia bpmn 2.0	Caixa baixa
substring("Guia BPMN 2.0", 5, 4) // BPMN	Parte da string
replace("Guia BPMN 2.0", "BPMN", "DMN") // Guia DMN 2.0	Substitui parte da string
contains("Guia BPMN 2.0", "BPMN") // true	Verifica se contém a string
string(123) // "123"	Transforma em string

Funções de Números

Exemplo // Resultado	Descrição
abs(-1) // 1	Valor absoluto
even(2) // true	Número par
odd(2) // false	Número ímpar
sqrt(9) // 3.0	Raiz quadrada

Funções de Datas

Exemplo // Resultado	Descrição
day of week(date("2020-12-31")) // "Thursday"	Dia da semana
month of year(date("2020-12-31")) // "December"	Mês do ano
week of year(date("2020-12-31")) // 53	Semana do ano

Funções de Listas

Exemplo // Resultado	Descrição
concatenate([1, 2, 3], [4, 5]) // [1, 2, 3, 4, 5]	Concatenar listas
count([1, 2, 3, 4, 5]) // 5	Contar lista
distinct values([1, 1, 2, 2, 3, 3, 4, 5]) // [1, 2, 3, 4, 5]	Valores distintos
flatten([1, [2, 3], [4, 5]]) // [1, 2, 3, 4, 5]	Aplainar
max([1, 2, 3, 4, 5]) //5	Máximo
mean([1, 2, 3, 4, 5]) //3	Intermediário
min([1, 2, 3, 4, 5]) / 1	Mínimo
reverse([1, 2, 3, 4, 5]) // [5, 4, 3, 2, 1]	Reverso
sort([5, 4, 1, 2, 3]) // [1, 2, 3, 4, 5]	Ordenação
sum([1, 2, 3, 4, 5]) // 15	Soma
index of(["a", "b", "c", "d", "e", "f"], "c") // 3	Posição
append([1, 2, 3, 4, 5], 6) // [1, 2, 3, 4, 5, 6]	Adicionar
list contains([1, 2, 3, 4, 5], 5) // true	Contido
remove(["a", "b", "c", "d", "e", "f"], 2) // ["a", "c", "d", "e", "f"]	Remoção
sublist([1, 2, 3, 4, 5], 2, 3) // [2, 3, 4]	Sublista

Esta guia de referência contém um resumo dos principais elementos das notações e modelos BPMN 2.0 e DMN 1.3 – v2021-09-20

FEEL – Friendly Enough Expression Language

Tipos de Dados

Tipo	Descrição	Exemplo
String	Qualquer sequência de caracteres	"Fulano de Tal", "123-XYZ"
Boolean	Lógica booleana	false, true
Integer	Números inteiros (32 bit)	-2,147,483,648 a 2,147,483,647
Long	Números inteiros (64 bit)	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,808
Double	Números com ponto flutuante	3.14159265358979323846
Date	Unidades temporais	date and time("2019-08-12T22:22:22")

Comparações

Nome	Operador	Exemplo	Descrição
Igual		"iss", "pis", "cofins"	Testa se o valor de entrada é igual a um dos literais da lista.
Menor	<	< 10	Testa se o valor de entrada é menor que o valor indicado.
Menor ou igual	<=	<= 10	Testa se o valor de entrada é menor ou igual ao valor indicado.
Maior	>	> 10	Testa se o valor de entrada é maior que o valor indicado.
Maior ou igual	>=	>= 10	Testa se o valor de entrada é maior ou igual ao valor indicado.

Intervalos

Início	Fim	Exemplo	Descrição
incluir	incluir	[1..10]	Testa se o valor de entrada é maior ou igual ao valor inicial e, se é menor ou igual ao valor final.
excluir	incluir]1..10] ou (1..10]	Testa se o valor de entrada é maior que o valor inicial e se é menor ou igual ao valor final.
incluir	excluir	[1..10[ou [1..10)	Testa se o valor de entrada é maior ou igual que o valor inicial e se é menor ao valor final.
excluir	excluir]1..10[ou (1..10)	Testa se o valor de entrada é maior que o valor inicial e menor que o valor final.
[date and time("2015-03-16T12:00:00") .. date and time("2016-03-16T12:00:00")]			Testa se a data de entrada é maior ou igual a data inicial e menor ou igual a data final.

Disjunções

Expressão	Descrição
2, 5, 7	Testa se o valor de entrada é 2, 5 ou 7.
<2, >=7	Testa se o valor de entrada é menor que 2 ou maior ou igual a 7.
2, [5..7]	Testa se o valor de entrada é 2 ou entre 5 e 7.
> date and time("2016-03-16T12:00:00")	Testa se a data de entrada é maior que a data fixa na regra da tabela de decisão.

Negações

Expressão	Descrição
not(3)	A valor de entrada é qualquer valor numérico diferente de 3
not(>10)	A entrada é qualquer valor numérico não maior que 10
not(10, [20..30])	A entrada não é igual a 10 ou entre 20 ou 30
not("Aceito")	A entrada não corresponde à sequência de caracteres "Aceito"

Roadmap

Educação Remota em Tecnologia e Transformação

<https://www.roadmap.com.br>