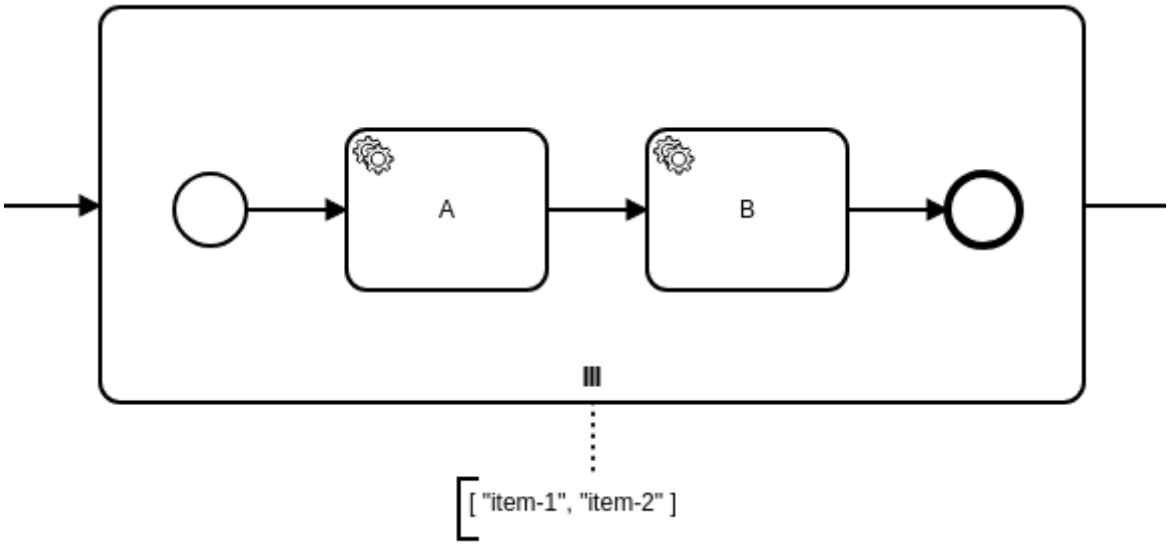
Instancia múltiple

Las siguientes actividades se pueden marcar como de varias instancias:

- Tareas de servicio
- Recibir tareas
- Subprocesos Embebidos

Una actividad de varias instancias se ejecuta varias veces, una vez para cada elemento de una colección determinada (como un bucle *foreach* en un lenguaje de programación).



En el nivel de ejecución, una actividad de instancias múltiples tiene dos partes: un **cuerpo de instancias múltiples** y una actividad interna. El cuerpo de varias instancias es el contenedor de todas las instancias de la actividad interna.

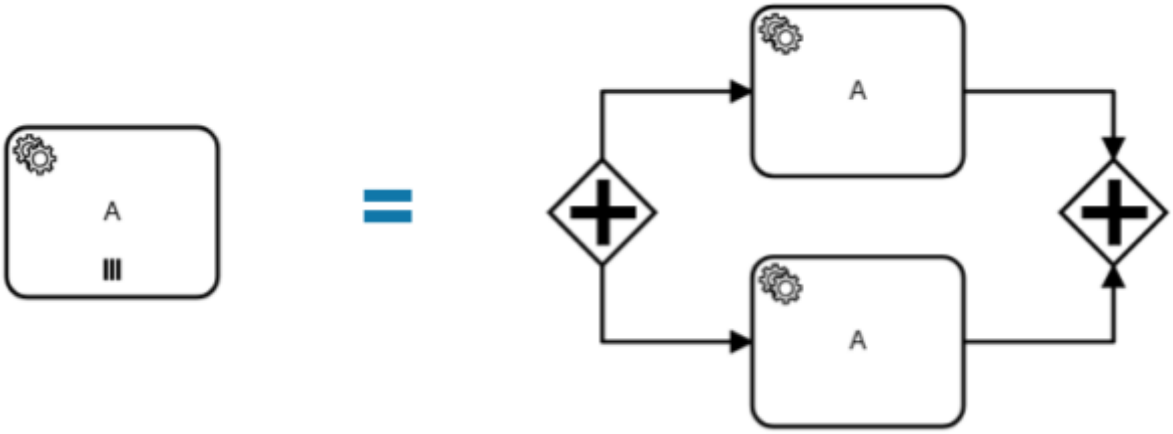
Cuando se ingresa la actividad, se activa el cuerpo de varias instancias y `inputCollection` se crea una instancia para cada elemento del mismo (secuencialmente o en paralelo). Cuando se completan todas las instancias, se completa el cuerpo y se deja la actividad.

Secuencial vs. Paralelo

Una actividad de varias instancias se ejecuta secuencialmente o en paralelo (por defecto). En el BPMN, se muestra una actividad secuencial de instancias múltiples con 3 líneas horizontales en la parte inferior. Un paralelo con 3 líneas verticales.

En el caso de una actividad **secuencial** de instancias múltiples, las instancias se ejecutan una por una. Cuando se completa una instancia, se crea una nueva instancia para el siguiente elemento en `inputCollection`.

En el caso de una actividad **paralela** de instancias múltiples, todas las instancias se crean cuando se activa el cuerpo de instancias múltiples. Las instancias se ejecutan de forma simultánea e independiente entre sí.



Definiendo la colección para iterar

Una actividad de varias instancias **debe** definir la `inputCollection` variable para iterar (por ejemplo `items`). La variable se lee desde la instancia de flujo de trabajo al activar el cuerpo de varias instancias. Debe ser `array` de cualquier tipo (p `["item-1", "item-2"]`. Ej .).

Para acceder al elemento actual `inputCollection` dentro de la instancia, la actividad de múltiples instancias puede definir la `inputElement` variable (por ejemplo `item`). El elemento se almacena como una variable local de la instancia con el nombre dado.

Si `inputCollection` está vacío, el cuerpo de varias instancias se completa inmediatamente y no se crean instancias. Se comporta como si se omitiera la actividad.

Recolectando la salida

La salida de una actividad de multi-instancia (por ejemplo, el resultado de un cálculo) se pueden recoger a partir de los casos mediante la definición de la `outputCollection` **y** la `outputElement` variable.

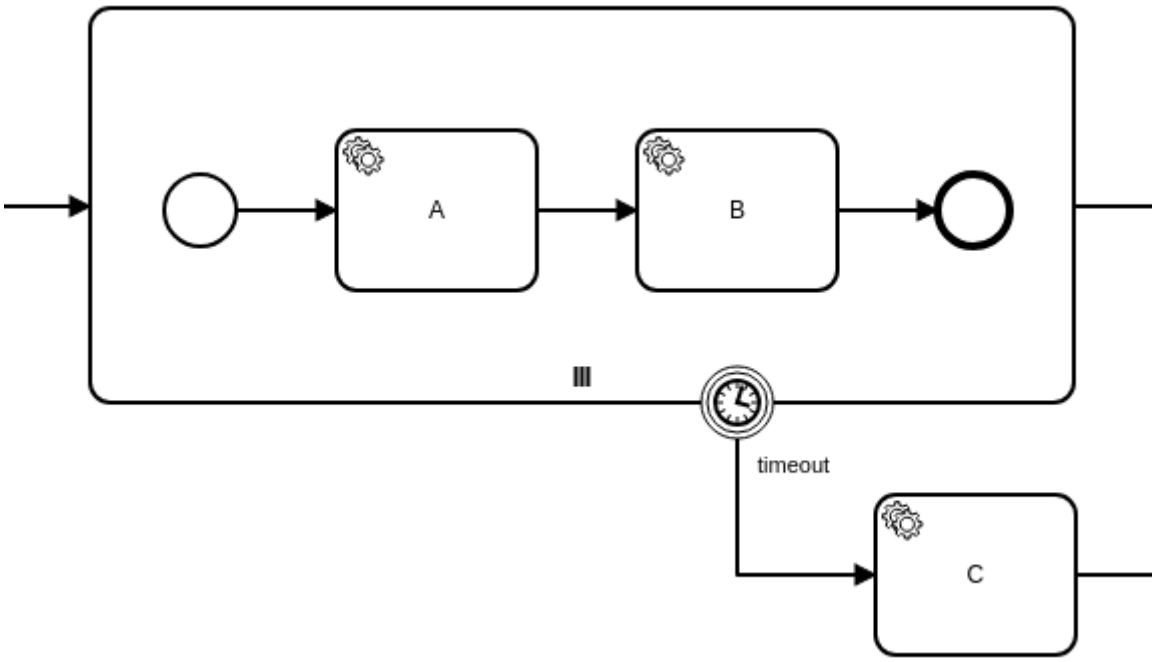
`outputCollection` define el nombre de la variable bajo la cual se almacena la salida recopilada (por ejemplo `results`). Se crea como variable local del cuerpo de varias

instancias y se actualiza cuando se completa una instancia. Cuando se completa el cuerpo de varias instancias, la variable se propaga a su ámbito principal.

`outputElement` define la variable de la que se recopila la salida de la instancia (por ejemplo `result`). Se crea como variable local de la instancia y debe actualizarse con la salida. Cuando se completa la instancia, la variable se inserta en `outputCollection` el mismo índice que el `inputElement` de `inputCollection`. Por lo tanto, el orden de la `outputCollection` se determina y coincide con el `inputCollection`, incluso para actividades paralelas de instancias múltiples. Si la `outputElement` variable no se actualiza, `null` se inserta en su lugar.

Si `inputCollection` está vacío, una matriz vacía se propaga como `outputCollection`.

Eventos límite



Los eventos de límite de interrupción y no interrupción se pueden adjuntar a una actividad de varias instancias.

Cuando se desencadena un evento de límite de interrupción, el cuerpo de varias instancias y **todas las instancias activas** finalizan. La `outputCollection` variable no se propaga al ámbito primario (es decir, sin salida parcial).

Cuando se desencadena un evento de límite sin interrupción, las instancias no se ven afectadas. Las actividades en la ruta de salida no tienen acceso a las variables locales ya que están limitadas a la actividad de instancias múltiples.

Variables especiales de instancias múltiples

Cada instancia tiene una variable local `loopCounter`. Contiene el índice en el `inputCollection` de esta instancia, comenzando con `1`.

Mapeos variables

Las asignaciones de variables de entrada y salida se pueden definir en la actividad de varias instancias. Se aplican **en cada instancia** al activar y completar.

Las asignaciones de entrada pueden acceder a las variables locales de la instancia (por ejemplo `inputElement`, `loopCounter`). Por ejemplo, para extraer partes de la `inputElement` variable y aplicarlas a variables separadas.

Las asignaciones de salida se pueden usar para actualizar la `outputElement` variable. Por ejemplo, para extraer una parte de las variables de trabajo.

Recursos adicionales

▶ Representación XML

▶ Usando el modelador BPMN

▶ Ciclo de vida del flujo de trabajo

Referencias

- Alcances variables
- Mapeos variables