

**BÚSQUEDA SEMÁNTICA EN UN REPOSITORIO DE PROCESOS
DE NEGOCIO**



**DANIEL FELIPE RIVAS BURBANO
DAVID SANTIAGO CORCHUELO CASTRO**

ANEXO No 4.

PATRONES DE FLUJO DE CONTROL

Director: Dr. JUAN CARLOS CORRALES

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELEMÁTICA
POPAYÁN
2010**

ANEXO No 4. PATRONES DE FLUJO DE CONTROL

La lista de los 43 patrones de flujo de control descritos por Wil van der Aalst y otros, en el trabajo de investigación denominado “*Workflow Control-Flow Patterns: A Revised View*”, fueron implementados en la notación BPMO. Cada implementación tuvo en cuenta, principalmente, la funcionalidad de cada patrón referida al trabajo de Wil van der Aalst.

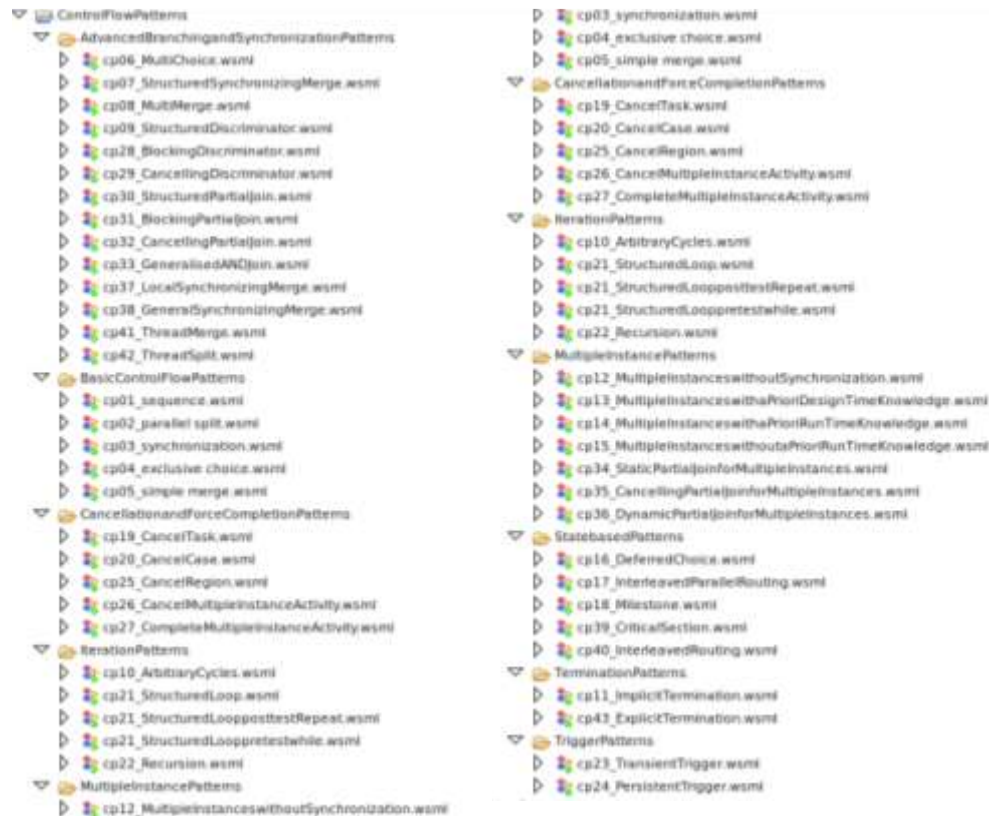


Figura 1. Implementación de los patrones de referencia en el modelador BPMO 1.4.

A continuación se presenta la descripción detallada de los 12 patrones utilizados en este proyecto, considerados básicos según el análisis y clasificación manifestado en la monografía.

Sequence Pattern: Este patrón sirve como el bloque fundamental para los BP Models. Consiste en una secuencia de mínimo 3 actividades de tipo tarea “T” que representa a las tareas de tipo ManualTask, Web ServiceTask, GoalTask de BPMO, donde el flujo de control conecta cada tarea precedente con la siguiente, como se puede apreciar por las flechas continuas de la Figura 28.

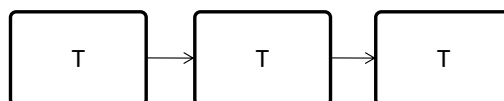


Figura 1. Sequence Pattern con una secuencia de 3 tareas.

Parallel Split Pattern: Este patrón permite dividir el flujo de control en 2 o más flujos salientes que puedan ejecutar otras actividades simultáneamente. En la Figura 17 se presenta una actividad de tipo tarea, que después de ser ejecutada, genera un flujo de control entrante hacia una compuerta Parallel Split, la cual divide el flujo de control en 2 ramas que permiten la posterior ejecución de 2 tareas en forma simultánea.

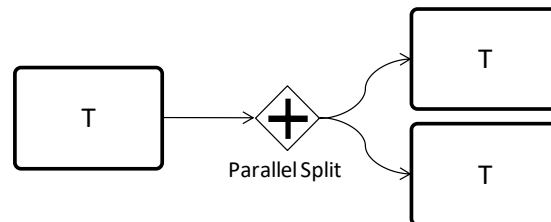


Figura 2. Parallel Split Pattern con 3 tareas.

Synchronization Pattern: Este patrón permite la convergencia del flujo de control de 2 o más ramas en una sola, tal que el flujo de control de la rama saliente ocurra solo cuando el flujo de cada rama entrante converja después de ejecutarse las actividades de cada rama. En la Figura 18 se presentan 2 actividades de tipo tarea que terminan su ejecución en momentos diferentes, generando flujos de control asíncronos, así, la compuerta Synchronization espera a que se ejecuten todas las tareas entrantes para generar el flujo de control de salida.

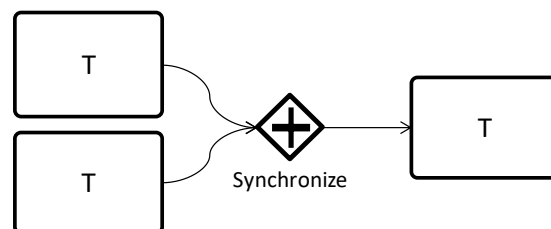


Figura 3. Synchronize Pattern con 3 tareas.

Exclusive Choice Pattern: Este patrón permite la divergencia del flujo de control de una rama en 2 o más ramas, tal que el flujo de control sea direccionado a una actividad específica dependiendo del resultado de una actividad precedente, los valores de la actividad ó algunos datos específicos hacen que haya una decisión para el enrutamiento del flujo de control. En la Figura 19 se presenta una actividad de tipo tarea, que después de ejecutarse genera un flujo de control de salida hacia una compuerta Exclusive Choice para luego enrutar el flujo de control por solo una de las ramas salientes, dependiendo de los resultados de la tarea inicial.

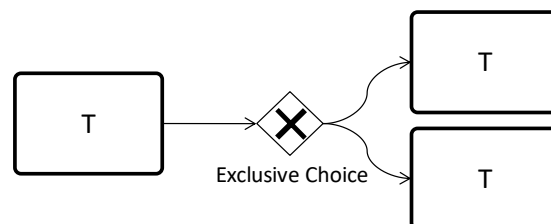


Figura 4. Exclusive Choice Pattern con 3 tareas.

Simple Merge Pattern: Este patrón permite la convergencia del flujo de control de 2 o más ramas en una sola, de manera asíncrona, tal que el flujo de control pase a una sola rama de salida. En la Figura 20 se presenta 2 actividades de tipo tarea, que independiente del momento en que terminen su ejecución, cada uno de sus flujos generados hacia la compuerta Simple Merge, convergen de manera asíncrona, ocasionando una ejecución diferente de la tarea posterior para cada uno de los resultados de las tareas iniciales.

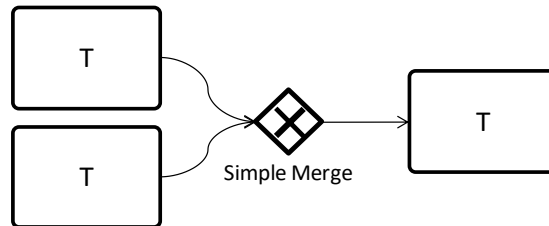


Figura 5. Simple Merge Pattern con 3 tareas.

Multiple Choice Pattern: Este patrón permite la divergencia del flujo de control de una rama en 2 o más ramas, tal que el flujo de control sea direccionado a cada una de las ramas salientes de manera síncrona o asíncrona dependiendo de los resultados de la tarea inicial. En la Figura 21 se presenta una actividad de tipo tarea, que después de ejecutarse genera un flujo de control entrante para la compuerta Multiple Choice, la cual divide de manera síncrona o asíncrona el flujo de control hacia las 2 tareas posteriores.

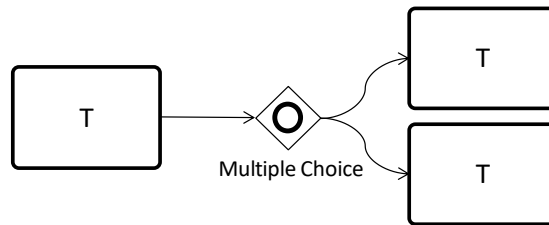


Figura 6. Multi Choice Pattern con 3 tareas.

Multi Merge Pattern: Este patrón permite la convergencia del flujo de control de 2 o más ramas en una sola, de manera síncrona o asíncrona, tal que el flujo de control pase a una sola rama de salida. En la Figura 22 se presenta 2 actividades de tipo tarea, que independiente del momento en que terminen su ejecución, cada uno de sus flujos generados hacia la compuerta Multi Merge, convergen de manera síncrona o asíncrona, ocasionando una ejecución diferente de la tarea posterior para cada uno de los resultados de las tareas iniciales.

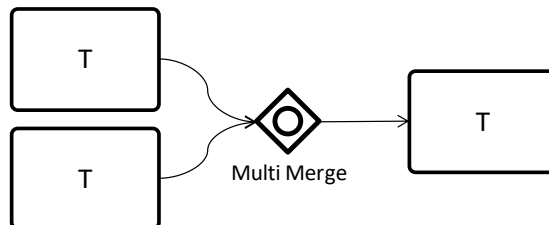


Figura 7. Multiple Merge Pattern con 3 tareas.

Deferred Choice Pattern: Este patrón permite la divergencia del flujo de control de una rama en 2 o más ramas, tal que el flujo de control sea direccionado a una actividad específica dependiendo del resultado de una actividad precedente, los valores de la

actividad ó algunos datos específicos hacen que haya una decisión para el enrutamiento del flujo de control. En la Figura 23 se presenta una actividad de tipo tarea, que después de ejecutarse genera un flujo de control de salida hacia una compuerta Deferred Choice para luego enrutar el flujo de control por solo una de las ramas salientes, dependiendo de los resultados de la tarea inicial.

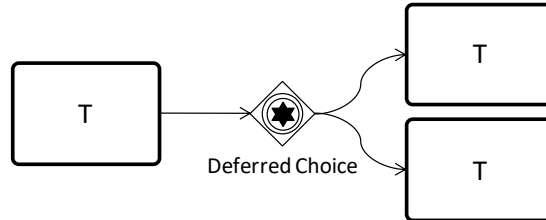


Figura 8. Deferred Choice Pattern con 3 tareas.

Discriminator Pattern: Este patrón permite la convergencia del flujo de control de 2 o más ramas en una sola de manera asíncrona, tal que el flujo de control pase a una sola rama de salida. En la Figura 24 se presenta 2 actividades de tipo tarea, que independiente del momento en que terminen su ejecución, cada uno de sus flujos generados hacia la compuerta Discriminator, converge de manera asíncrona de tal manera que una vez llega a la compuerta Discriminator el flujo de control correspondiente a la tarea que lo generó primero, el flujo correspondiente a la tarea pendiente se termina, ocasionando una ejecución diferente de la tarea posterior para cada uno de los resultados de las tareas iniciales.

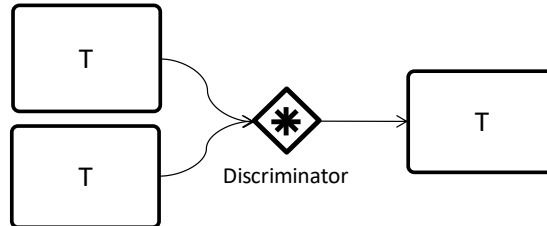


Figura 9. Discriminator Pattern con 3 tareas.

Interleaved Parallel Routing Pattern: Este patrón permite la divergencia del flujo de control de una rama en 2 o más ramas, tal que el flujo de control sea direccionado a una actividad específica dependiendo del resultado de una actividad precedente, los valores de la actividad ó algunos datos específicos hacen que haya una decisión para el enrutamiento del flujo de control. En la Figura 25 se presenta una actividad de tipo tarea, que después de ejecutarse genera un flujo de control de salida hacia una compuerta Interleaved Parallel Routing para luego enrutar el flujo de control por solo una de las ramas salientes, dependiendo de los resultados de la tarea inicial.

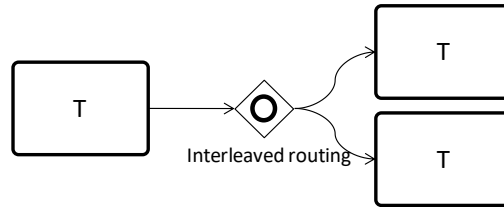


Figura 10. Interleaved Parallel Routing Pattern con 3 tareas.

Multi Merge Synchronization Pattern: Este patrón permite la convergencia del flujo de control de 2 o más ramas en una sola, tal que el flujo de control de la rama saliente ocurra solo cuando el flujo de cada rama entrante converja después de ejecutarse las actividades de cada rama. En la Figura 26 se presentan 2 actividades de tipo tarea que terminan su ejecución en momentos diferentes, generando flujos de control asíncronos, así, la compuerta Multi Merge Synchronization espera a que se ejecuten todas las tareas entrantes para generar el flujo de control de salida.

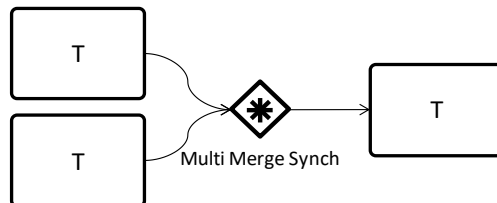


Figura 11. Multi Merge Synchronization Pattern con 3 tareas.

Multiple Instantiation Pattern: Este patrón permite la divergencia del flujo de control de una rama en 2 o más ramas, tal que el flujo de control sea direccionado a una actividad específica dependiendo del resultado de una actividad precedente, los valores de la actividad ó algunos datos específicos hacen que haya una decisión para el enrutamiento del flujo de control. En la Figura 27 se presenta una actividad de tipo tarea, que después de ejecutarse genera un flujo de control de salida hacia una compuerta Multiple Instantiation para luego enrutar el flujo de control por solo una de las ramas salientes, dependiendo de los resultados de la tarea inicial.

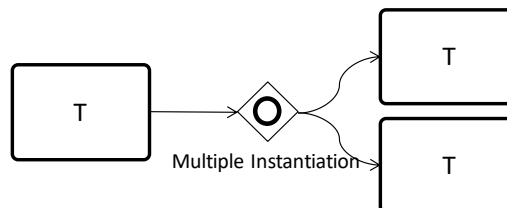


Figura 12. Multiple Instantiation Synchronization Pattern con 3 tareas.