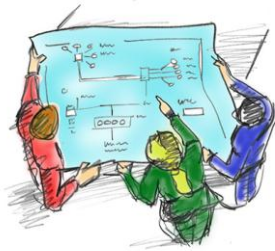




Universidad Católica de Santiago del Estero
Departamento Académico Rafaela

Gestión de Procesos de Negocios



Docentes:

- Ing. Lorena D'Iorio
- Ing. Ariel Rossanigo
- Ing. Román Zenobi

Bibliografía

- ▶ P. Wohed, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, and N. Russell. On the Suitability of BPMN for Business Process Modelling In Proceedings of the 4th International Conference on Business Process Management (BPM 2006), Vienna, September 2006. Springer Verlag.
<http://www.workflowpatterns.com/documentation/documents/BPMN-eval-BPM06.pdf>
- ▶ Villareal, Pablo: Apuntes cátedra *Gestión e Integración de Procesos de Negocio en Empresas y en Ambientes Business-to-Business*, EMISI, UTN, Santa Fe 2011.
- ▶ Links:
www.workflowpatterns.com

Unidad 4: : *PATRONES DE MODELADO DE PROCESOS*

AGENDA:

- Patrones
- Tipos de Patrones
- Herramientas de Modelado con BPMN

PATRONES DE MODELADO DE PROCESOS



Modelo que sirve de muestra para sacar otra cosa igual. (RAE)

Aquel objeto o sustancia que se usará como muestra para medir alguna magnitud o bien para replicarla, en el caso que se busque esta situación.

Un **patrón** es la descripción de un **problema** (recurrente o repetitivo) y su **solución**; el cual se puede aplicar en diferentes contextos, y además proporciona consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos o implementaciones.

Proveen independencia de la tecnología de implementación y de los requerimientos del dominio de aplicación

Ejemplo: Patrones de Diseño Orientado a Objetos [Gamma y otros, 1995]

PATRONES DE MODELADO DE PROCESOS

Patrones de Diseño

Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software.

Resultan ser una solución a un problema de diseño.

Para que una solución sea considerada un patrón debe poseer ciertas características:

- **Efectividad**: resuelve problemas similares en ocasiones anteriores;
- **Reutilizable**: es aplicable a diferentes problemas de diseño en distintas circunstancias.

1979: el arquitecto Christopher Alexander aportó al mundo de la arquitectura el libro ***The Timeless Way of Building***. Proponía aprendizaje y uso de patrones para la construcción de edificios de una mayor calidad.

"Cada patrón describe un problema que ocurre infinidad de veces en nuestro entorno, así como la solución al mismo, de tal modo que podemos utilizar esta solución un millón de veces más adelante sin tener que volver a pensarla otra vez."

1987: Ward Cunningham y Kent Beck usaron varias ideas de Alexander para desarrollar cinco patrones de interacción hombre-ordenador (HCI). Publicaron un artículo titulado **Using Pattern Languages for OO Programs**.

1990: Los patrones ingresan con éxito en el mundo de la informática a partir de la publicación del libro ***Design Patterns*** escrito por el grupo Gang of Four (GoF) compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, 23 patrones de diseño comunes.

Patrones de Modelado de Procesos

Patrones de Flujo de Control

- Capturan la forma en que los tipos/constructores de flujo de control son representados y usados en procesos/workflows.
- Soportan la perspectiva de comportamiento de los procesos.

Patrones de Datos

- Capturan la forma en que los datos son representados y usados en procesos/workflows.
- Soportan la perspectiva de información de los procesos.

Patrones de Recursos

- Capturan la forma en que los recursos son representados y usados en procesos/workflows.
- Soportan la perspectiva organizacional de los procesos.

Patrones de Flujo de Control

Fundamento:

No existe un consenso acerca de los constructores que deberían proveer los lenguajes de modelado o especificación de procesos.

No existen conceptos estándares para el modelado de procesos de negocio.

Existe una gran cantidad de enfoques diferentes para definir el flujo de control en Sistemas de Gestión de Procesos de Negocio.

BPMs Comerciales y Académicos utilizan diferentes lenguajes con diferencias semánticas fundamentales.

Conceptos con nombres similares tienen un significado diferente.

Diferentes lenguajes imponen diferentes restricciones a un concepto.

Algunos conceptos complejos están sólo en algunos lenguajes.

- Lenguajes tienen diferentes niveles de poder de expresividad -

Patrones de Flujo de Control (cont.)

Proveen un enfoque práctico para entender y analizar los diferentes lenguajes con respecto a la definición del flujo de control de procesos/workflows.

Proveen abstracciones de los constructores que puede proveer un lenguaje.

Patrones consisten de:

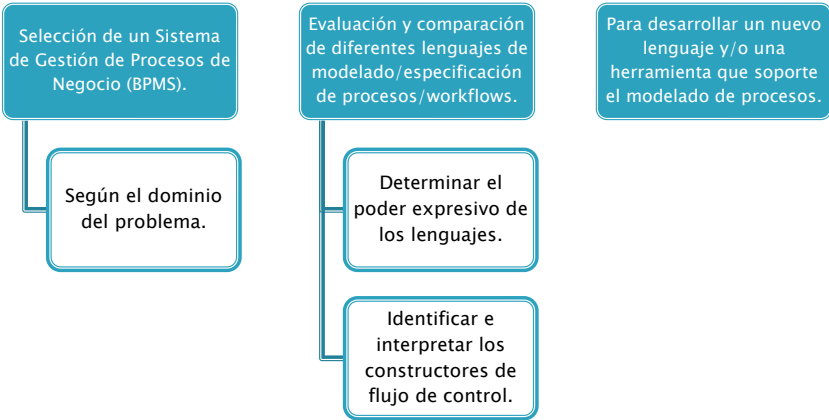
- Una descripción del flujo de control capturado.
- Posible sinónimos.
- Ejemplos de escenarios concretos.
- Problemas de implementación y soluciones (parciales) a los mismos.

Son implementados en un lenguaje específico:

- Tener en cuenta la diferencia entre la semántica del patrón y su implementación en un lenguaje específico.

Patrones de Flujo de Control (cont.)

Uso o aplicación de los patrones:



– No se enfocan o aplican al manejo de excepciones –

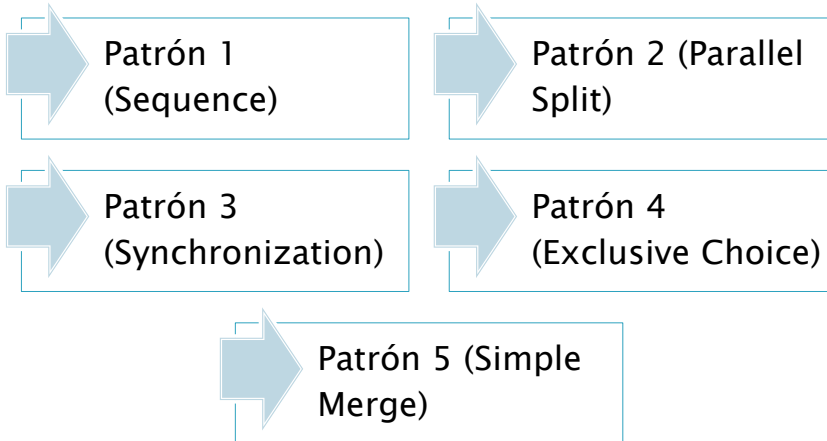
Patrones de Flujo de Control (cont.)

Categorías de Patrones:

- Patrones de Flujo de Control Básicos.
- Patrones de Bifurcación y Sincronización Avanzados.
- Patrones Estructurales.
- Patrones que involucran Múltiples Instancias.
- Patrones basados en Estados.
- Patrones de Cancelación.

Patrones de Flujo de Control (cont.)

Patrones de Flujo de Control Básicos:



Patrones de Flujo de Control (cont.)

Patrón 1 (Sequence)

Descripción:

- Una actividad es habilitada después de la terminación de otra actividad dentro de un mismo caso (instancia) de un proceso.

Sinónimo:

- Ruteo secuencial, ruteo serial.

Patrones de Flujo de Control (cont.)

Patrón 1 (Sequence)



Patrones de Flujo de Control (cont.)

Patrón 2 (Parallel Split)

Descripción:

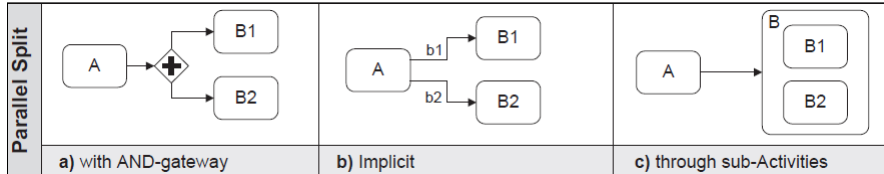
- Un punto en el proceso donde un simple flujo de control se divide en dos o más flujos de control que pueden ser ejecutados en paralelo (actividades pueden ser ejecutadas simultáneamente o en cualquier orden).

Sinónimo:

- AND-split, fork, parallel routing.

Patrones de Flujo de Control (cont.)

Patrón 2 (Parallel Split)



Patrones de Flujo de Control (cont.)

Patrón 3 (Synchronization)

Descripción:

- Un punto en el proceso donde múltiples actividades convergen en un único flujo de control, sincronizando múltiples flujos. Cada flujo de control de entrada a un sincronizador es ejecutado una sola vez, y todos necesitan ser completados para que el sincronizador continúe el flujo de control.

Sinónimo:

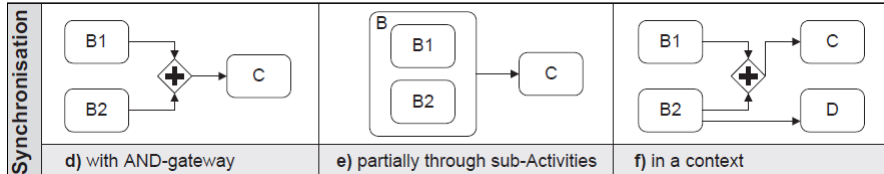
- AND-join, join, sincronizador.

Implementación:

- Lenguajes de modelado de procesos de negocio y BPMs pueden soportar la ejecución de constructores AND-split y AND-join "explícitos" o "implícitos".

Patrones de Flujo de Control (cont.)

Patrón 3 (Synchronization)



Patrones de Flujo de Control (cont.)

Patrón 4: Exclusive Choice

Descripción:

- Un punto en el proceso donde basado en la evaluación de datos del proceso (variables de instancia del proceso), uno de varios flujos alternativos es seleccionado.

Sinónimo:

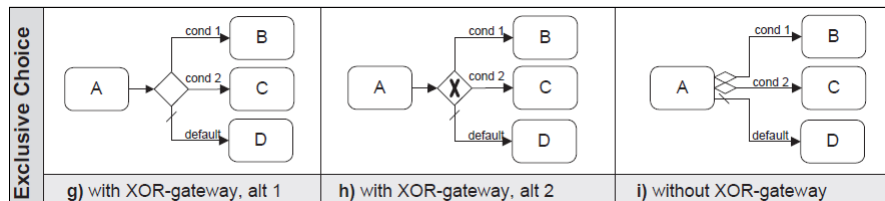
- XOR-split, decision, conditional routing, switch.

Implementación:

- Algunos lenguajes proveen un constructor explícito para este patrón. En otros lenguajes éste es emulado especificando condiciones mutuamente exclusivas sobre varias transiciones salientes de una actividad.

Patrones de Flujo de Control (cont.)

Patrón 4: Exclusive Choice



Patrones de Flujo de Control (cont.)

Patrón 5: Simple Merge

Descripción:

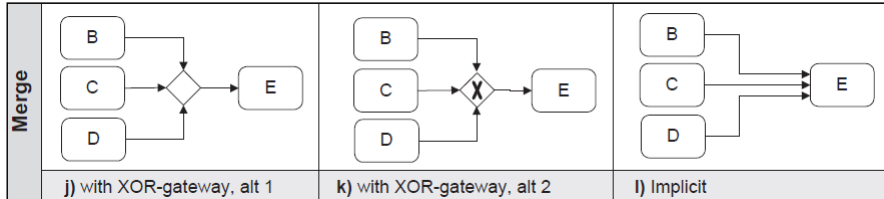
- Un punto en el proceso donde dos o más flujos alternativos se unen sin sincronización (Supuesto: los flujos alternativos no se ejecutan en paralelo).

Sinónimo:

- XOR-split, decision, conditional routing, switch.

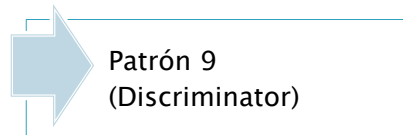
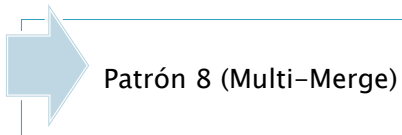
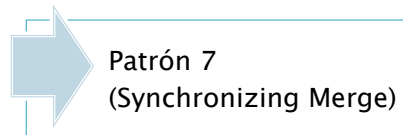
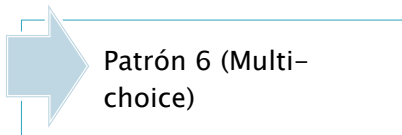
Patrones de Flujo de Control (cont.)

Patrón 5: Simple Merge



Patrones de Flujo de Control (cont.)

Patrones de Bifurcación y Sincronización Avanzados:



Patrones de Flujo de Control (cont.)

Patrón 6: Multi-Choice

Descripción:

- Uno o varios flujos alternativos pueden ser seleccionados en base a la evaluación de datos del proceso (variables de instancia del proceso) por el BPMS.

Sinónimo:

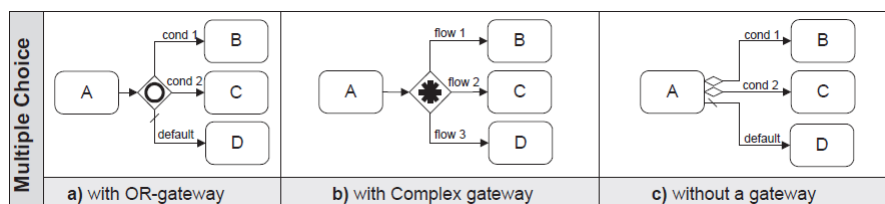
- OR-split, Conditional routing, selection.

Problemas

- Este patrón es implementado en forma directa si el lenguaje soporta condiciones sobre transiciones o un constructor explícito del mismo.
- De lo contrario, se lo debe implementar combinando los patrones Parallel Split y Exclusive Choice.

Patrones de Flujo de Control (cont.)

Patrón 6: Multi-Choice



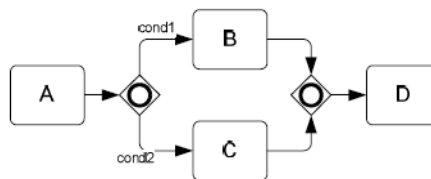
Patrones de Flujo de Control (cont.)

Patrón 7: Synchronizing Merge

Descripción:	Sinónimo:	Problemas	Implementación:
<ul style="list-style-type: none">• Una forma de sincronización en la cual la ejecución procede si y sólo si uno de los flujos o caminos de entrada ha finalizado:• Si se ejecutó más de un camino, estos son sincronizados.• Si se ejecutó un sólo camino, se continúa la ejecución sin sincronización.	<ul style="list-style-type: none">• Synchronizing Join.	<ul style="list-style-type: none">• Decidir cuándo sincronizar y fusionar los flujos.	<ul style="list-style-type: none">• Sólo en tiempo de ejecución puede suceder que se decida cuándo el flujo de salida puede ser disparado.• Otra opción es disparar el flujo de salida usando condiciones.• Si no existe implementación directa: evitar el uso del OR-split e implementarlo combinando AND-splits y XOR-Splits, y utilizando un AND-join y XOR-join.

Patrones de Flujo de Control (cont.)

Patrón 7: Synchronizing Merge



Patrones de Flujo de Control (cont.)

Patrón 8 (Multi-Merge)

Descripción:

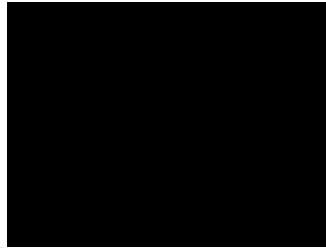
- Un punto en el proceso donde dos o más flujos convergen sin sincronización. Si más de un flujo es activado, posiblemente en forma concurrente, la actividad que sigue al merge es comenzada por cada activación de un flujo entrante.

Implementación:

- Si el lenguaje no soporta múltiples instancias de actividades, y el multi-merge no es parte de un loop, la solución es duplicar las actividades que le siguen por cada flujo.

Problemas

- Algunos BPMs no generan otra instancia de la actividad siguiente al multi-merge si existe alguna que se está ejecutando.



Patrones de Flujo de Control (cont.)

Patrón 9 (Discriminator)

Descripción:

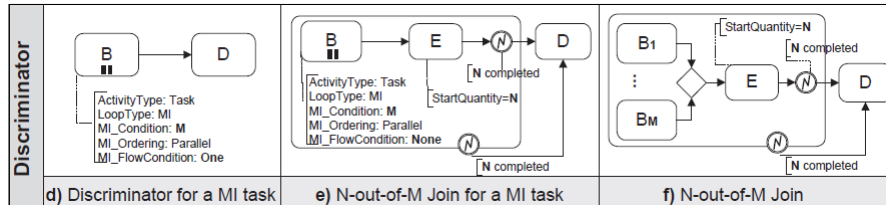
- Una forma de sincronización en la cual, para varios flujos entrantes ejecutándose en paralelo, el primero que finaliza habilita la siguiente actividad. A partir de ese momento espera la terminación de los restantes flujos y luego los “ignora”.

Problemas:

- Generalmente no es soportado completamente por los BPMs y lenguajes.
- Para que se comporte apropiadamente en loops se necesita hacer un seguimiento de los flujos que han finalizados y resetearse a sí mismo.

Patrones de Flujo de Control (cont.)

Patrón 9 (Discriminator)



Patrones de Flujo de Control (cont.)

Patrones Estructurales:

Patrón 10 (Arbitrary Cycles)

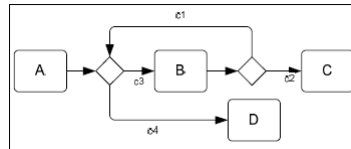
Patrón 11 (Implicit Termination)

Se refieren a restricciones sintácticas que algunos lenguajes imponen, principalmente en la definición de ciclos o bucles.

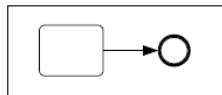
Patrones de Flujo de Control (cont.)

Patrones Estructurales:

Patrón 10 (Arbitrary Cycles)



Patrón 11 (Implicit Termination)



Patrones de Flujo de Control (cont.)

Patrón 10 (Arbitrary Cycles)

Descripción:

- Un punto en el proceso donde una o más actividades pueden ser realizadas varias veces. Puede haber más de un punto de entrada y de salida.

Sinónimos

- Loop, iteration, cycles.

Problema:

- Algunos lenguajes sólo soportan ciclos estructurados:
- (un único punto de entrada y un único punto de salida)

Implementación

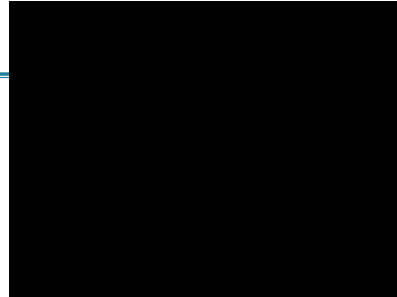
- En lenguajes que soportan sólo ciclos estructurados es posible implementar ciclos arbitrarios usando postcondiciones sobre descomposiciones (subprocesos).

Patrones de Flujo de Control (cont.)

Patrón 11 (Terminación Implícita)

Descripción:

- Un proceso debería ser finalizado cuando no existan más actividades habilitadas para su ejecución.



Problema:

- La mayoría de los lenguajes y BPMs utilizan un nodo final explícito. Ninguna actividad puede ser ejecutada si el nodo es alcanzado.

Patrones de Flujo de Control (cont.)

Patrones que involucran Múltiples Instancias:

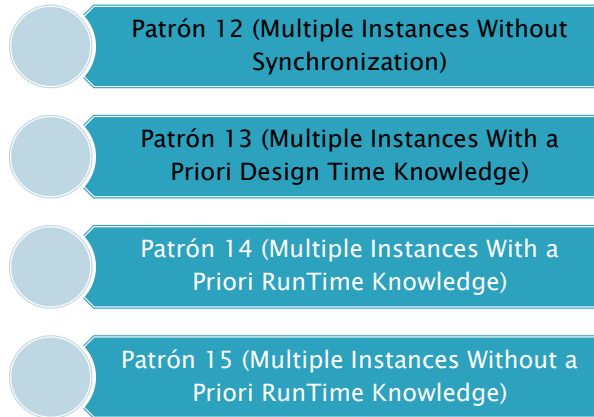
Patrones para representar múltiples instancias de actividades.

Dos requerimientos a considerar:

- Capacidad para iniciar múltiples instancias de una actividad o subproceso (estos patrones deben satisfacer este requerimiento)
- Capacidad para sincronizar estas instancias y continuar luego de que todas hallan finalizado (esto puede ser ignorado en los casos que no se requiera sincronización)

Patrones de Flujo de Control (cont.)

Patrones que involucran Múltiples Instancias:



Patrones de Flujo de Control (cont.)

Patrón 12 (Multiple Instances Without Synchronization)

Descripción:

- Para una instancia de un proceso, una actividad es habilitada varias veces.
- Las instancias de la actividad son independientes unas de otras.
- No se requiere sincronizar las instancias de la actividad.

Problema:

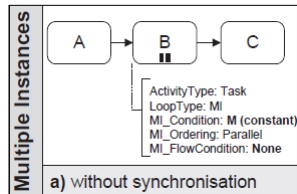
- Algunos BPMs requieren que se defina explícitamente el número de instancias permitidas.

Implementación

- Soportado generalmente por todos los BPMs. El problema es la sincronización, no la creación de instancias.
- Este puede ser implementado usando un loop, contadores y el parallel split.

Patrones de Flujo de Control (cont.)

Patrón 12 (Multiple Instances Without Synchronization)



Patrones de Flujo de Control (cont.)

Patrón 13 (Multiple Instances With a Priori Design Time Knowledge)

Descripción:

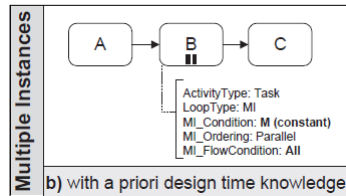
- Para una instancia de un proceso, una actividad es habilitada varias veces. El número de instancias de la actividad es conocido en tiempo de diseño. Una vez que todas las instancias son finalizadas, alguna otra actividad puede comenzar.

Implementación

- Si el BPMS no lo soporta directamente, una solución simple es replicar la actividad tantas veces como se necesita, utilizando el patrón parallel split para replicarlas y el de sincronización para sincronizar a las mismas.

Patrones de Flujo de Control (cont.)

Patrón 13 (Multiple Instances With a Priori Design Time Knowledge)



Patrones de Flujo de Control (cont.)

Patrón 14 (Multiple Instances With a Priori Runtime Knowledge)

Descripción:

- Para una instancia de un proceso, una actividad es habilitada varias veces. El número de instancias de la actividad varía y puede depender de las características de la instancia del proceso o la disponibilidad de recursos, pero es conocido en alguna etapa durante el tiempo de ejecución, previo a que las instancias de la actividad sean creadas. Una vez que todas las instancias son finalizadas otra actividad puede ser comenzada.

Problema:

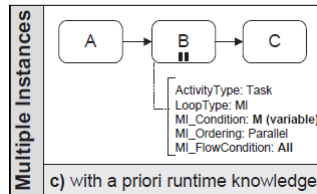
- si el nro. de instancias no es conocida de antemano no es posible replicar la actividad en el modelo de proceso.

Implementación

- Si el BPMS no lo soporta directamente, pero soporta múltiples instancias, es posible representarlo usando colas o técnicas de programación que están más allá de los constructores de flujo de control.

Patrones de Flujo de Control (cont.)

Patrón 14 (Multiple Instances With a Priori Runtime Knowledge)



Patrones de Flujo de Control (cont.)

Patrón 15 (Multiple Instances Without a Priori Runtime Knowledge)

Descripción:

• Para una instancia de un proceso, una actividad es habilitada varias veces. El número de instancias de la actividad no es conocido en ninguna etapa durante el tiempo de ejecución, previo a que las instancias de la actividad sean creadas. Una vez que todas las instancias son finalizadas alguna otra actividad puede ser comenzada. La diferencia con el patrón 14 es que mientras algunas de las instancias se están ejecutando o ya han finalizado, nuevas instancias pueden ser creadas.

Problema:

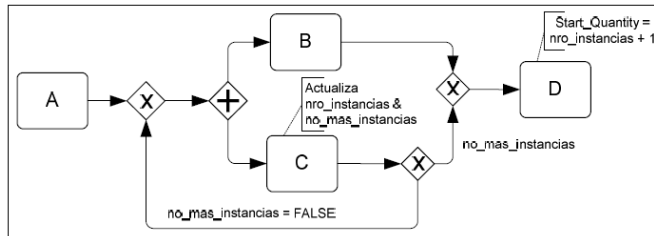
• Algunos lenguajes sólo soportan la creación de múltiples instancias si el nro. de instancias puede ser conocido en alguna etapa del proceso

Implementación

• Si el BPMS no lo soporta directamente, éste patrón es difícil de implementar porque no se conoce el número de actividades que pueden ser instanciadas.

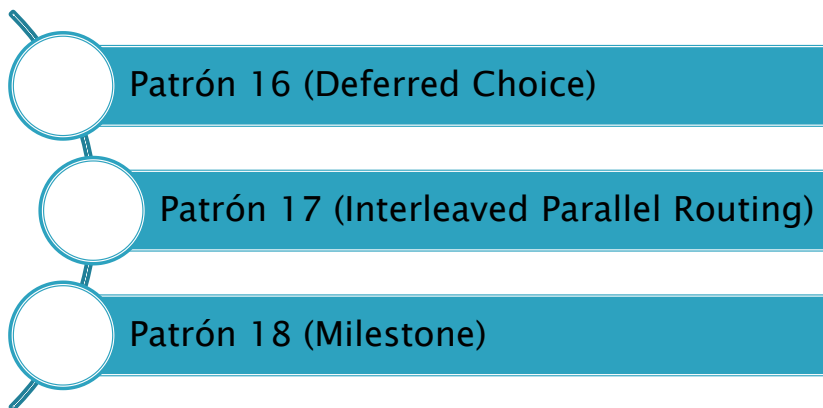
Patrones de Flujo de Control (cont.)

Patrón 15 (Multiple Instances Without a Priori Runtime Knowledge)



Patrones de Flujo de Control (cont.)

Patrones basados en Estados:



Patrones de Flujo de Control (cont.)

Patrones basados en Estados:

- En casos reales, la mayoría de las instancias de un proceso están en un estado en donde esperan por ser ejecutadas, en lugar de estar ejecutándose.
- Constructores básicos para selección de flujos están basados en datos o capturados a través de decisiones de alguna actividad (selección explícita).
- A veces se requiere que la decisión sea demorada tanto como sea posible, es decir, que se toma en tiempo de ejecución (selección implícita).
- En estos casos, BPMs que no representan estados tienen dificultades para dar soporte al requerimiento anterior

Patrones de Flujo de Control (cont.)

Patrón 16 (Deferred Choice)

Descripción:

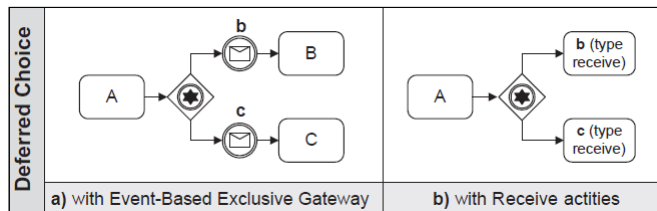
- Un punto de un proceso en donde uno de varios flujos es seleccionado. A diferencia del XOR-split, la decisión no es realizada explícitamente sino que varias alternativas son ofrecidas al BPMS:
 - * Sólo una alternativa es ejecutada. Una vez que el BPMS activó un flujo alternativo, las otras son descartadas.
 - * La selección es demorada hasta que comience el procesamiento de una de los flujos alternativos.

Sinónimos:

- External choice, implicit choice, deferred XOR-split.

Patrones de Flujo de Control (cont.)

Patrón 16 (Deferred Choice)



Patrones de Flujo de Control (cont.)

Patrón 17 (Interleaved Parallel Routing)

Descripción:

- Un conjunto de actividades se ejecutan en un orden arbitrario: cada una de las actividades es ejecutada, pero el orden se decide en tiempo de ejecución. Dos actividades nunca son ejecutadas al mismo tiempo.

Sinónimo:

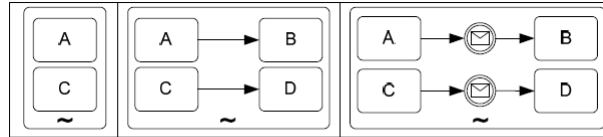
- Unordered sequence.

Implementación

- * Una solución (a veces no muy aceptable) es fijar el orden de ejecución de las actividades:
- * Otra solución es usar una combinación de secuencias posibles y XOR-split (desventaja: orden está explícito).
- * Otra solución es usar una combinación de secuencias posibles y Deferred Choice (problema cuando son demasiadas las tareas en paralelo).

Patrones de Flujo de Control (cont.)

Patrón 17 (Interleaved Parallel Routing)



17.a) Tareas simples

17.b) Para secuencias de actividades

17.c) Para secuencias de actividades y con Eventos de Mensajes

Patrones de Flujo de Control (cont.)

Patrón 18 (Milestone)

Descripción:

- La habilitación de una actividad depende de si la instancia se encuentra en un estado específico, es decir, si un cierto milestone ha sido alcanzado y aún no expiró.
- Puede ser utilizado para consultar el estado de una instancia de proceso.

Sinónimos:

- deadline, state condition.

Problema

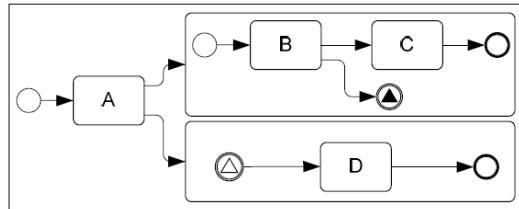
- son pocos los sistemas que lo soportan directamente, aunque es un patrón muy útil en casos reales.

Implementación

- En algunos casos es posible utilizar el patrón deferred choice o introducir una variable.
- Es posible representarlo usando eventos o mensajes de envío y recepción.

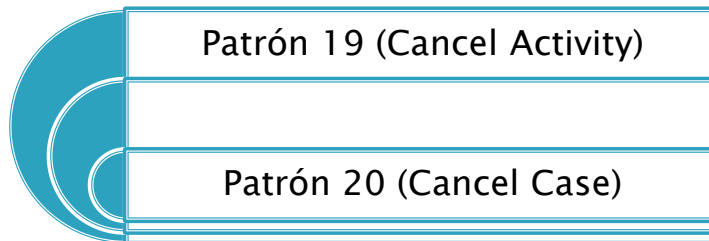
Patrones de Flujo de Control (cont.)

Patrón 18 (Milestone)



Patrones de Flujo de Control (cont.)

Patrones de Cancelación:



Patrones de Flujo de Control (cont.)

Patrón 19: Cancel Activity

Descripción:

- Una actividad habilitada es deshabilitada, es decir, el flujo esperando por la ejecución de la actividad es removido.

Sinónimos:

- withdraw activity.

Problema

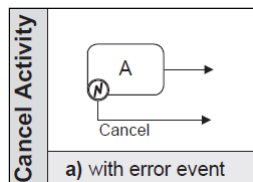
- Son pocos los lenguajes que lo soportan directamente en el lenguaje de modelado (gráficamente).

Implementación

- Muchos BPMs soportan este patrón usando una API → No es posible modelarlo en forma directa con el lenguaje, sino dentro de las actividades es posible invocar una función que soporte la deshabilitación de otra actividad.

Patrones de Flujo de Control (cont.)

Patrón 19: Cancel Activity



Patrones de Flujo de Control (cont.)

Patrón 20: Cancel Case

Descripción:

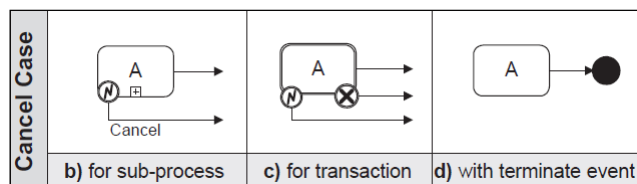
- Una instancia de proceso es cancelada y removida completamente (aún si existen partes del proceso que son instanciadas varias veces).

Sinónimos:

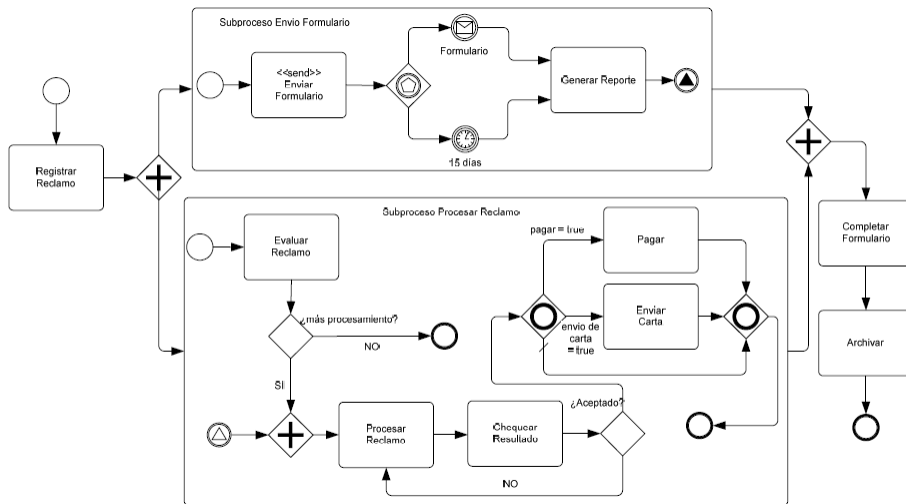
- withdraw activity.

Patrones de Flujo de Control (cont.)

Patrón 20: Cancel Case



Patrones de Flujo de Control (cont.)



Herramientas de Modelado con BPMN

BizAgi: www.bizagi.com

Tibco: http://developer.tibco.com/business_studio/default.jsp

Eclipse BPMN: <http://www.eclipse.org/bpmn/>

Jbpm:

<https://hudson.jboss.org/hudson/job/jBPM/lastSuccessfulBuild/rtrifact/jbpm-distribution/target/jbpm-5.2.0-SNAPSHOT/docsbuild/jbpm-docs/html/ch10.html>

ADONIS: www.adonis-community.com

BONITA: <http://www.bonitasoft.com/>

Visio Stencils